```
0001                    ************************************************************************
0002                    *                                                                      *
0003                    * Title:                      H J S 2 2                                *
0004                    *                                                                      *
0005                    * Description:                                                          *
0006                    *    This is the HJS22 monitor program. It only runs on the            *
0007                    *    homebrew TTL CPU build by Henk Stegeman in 1976.                  *
0008                    *                                                                      *
0009                    *    This source can be assembled by a modified M6809 assembler.       *
0010                    *      C:> as9 hjs22.as -l > hjs22.lst                                 *
0011                    *                                                                      *
0012                    * History:                                                              *
0013                    *    14/04/2001 - Version 1.4 Set breakpoint logic added.              *
0014                    *                                                                      *
0015                    *    15/04/2001 - Version 1.4a Bug fixed in unable to set BP logic.    *
0016                    *                                                                      *
0017                    *    ??/04/2001 - Version 1.5 User status added & more rel addressing  *
0018                    *                                                                      *
0019                    *    22/01/2007 - Version 1.5a More comment added.                     *
0020                    *                                                                      *
0021                    * Notes:                                                               *
0022                    *    - The auto-restart sw must be set to ON for breakpoint restart.   *
0023                    *    - Object is not relocatable and runs in interrupt level 3.        *
0024                    *    - See also boot22.asm which loads program from EPROM to RAM       *
0025                    *      at location $0200.                                              *
0026                    *    - boot22 starts at $8000.                                         *
0027                    *    - The Tx/Rx speed is 1200 bps.                                    *
0028                    *                                                                      *
0029                    ************************************************************************
0030 0200                           org   $0200                   start address
0031                    *
0032                    *** HJS22 equates.
0033                    *
0034 0000                  R0       equ   0                       general registers.
0035 0001                  R1       equ   1
0036 0002                  R2       equ   2
0037 0003                  R3       equ   3
0038 0004                  R4       equ   4
0039 0005                  R5       equ   5
0040 0006                  R6       equ   6
```

```
0041 0007                  R7       equ   7
0042 0008                  R8       equ   8
0043 0009                  R9       equ   9
0044
0045 000B                  A0       equ   11                    address registers.
0046 000D                  A1       equ   13
0047 000F                  A2       equ   15
0048
0049 001B                  escape   equ   $1B                   Escape.
0050 000D                  cr       equ   $0D                   Carriage Return.
0051 000A                  lf       equ   $0A                   Line Feed.
0052 0000                  null     equ   $00                   Null.
0053 008F                  brkpoint equ   $8F                   Breakpoint. (HLT with restart)
0054
0055                    *
0056                    *** HJS22 monitor program starts here...
0057                    *
0058 0200                  monitor  equ   *
0059 0200 EE 0C                     jmp   *,eyecatch            branch over
0060 0202 05 B6                     fdb   txmit
0061 0204 05 E4                     fdb   recv
0062 0206 32 32 2F 30 34 2F         fcc   '22/04/01'            eye-catcher.
         30 31
0063 020E                  eyecatch equ   *
0064                    *   Clean monitor start / restart if Acc=$00.
0065 020E E0 05                     jiz   *,monstart            start monitor (POR) ?
0066                    *   Breakpoint detected if Acc=$01.
0067 0210 24 01                     cmp   #$01                  entry caused by breakpoint ?
0068 0212 C2 04 EA                  jie   brkdet                jump if yes.
0069
0070                    *   Monitor start / restart.
0071 0215                  monstart equ   *
0072 0215 27 03                     lda   #$03                  switch to interrupt level 3
0073 0217 48 06 D6                  sta   intlvl3               *
0074 021A A7                        rtc                         *
0075 021B 86 06 D6                  swp   intlvl3               *
0076 021E CC 06 24                  jts   crlf                  start with a new line.
0077 0221 CC 06 24                  jts   crlf                  *
0078 0224 8B 0B 06 DA               lrd   %A0,#msg01            Welcome msg.
0079 0228 CC 06 36                  jts   txmsg                 xmit it
```

```
0080                    *   Any breakpoint left active in user program ?
0081 022B 47 06 D7          removbrk lda   saveopc                restore any pre-restart
0082 022E 24 8F                      cmp   #brkpoint              user program opcode ?
0083 0230 E2 12                      jie   *,norest               no.
0084 0232 58 06 D9                   sta   [savebpa+1]            restore it.
0085 0235 27 8F                      lda   #brkpoint              *
0086 0237 48 06 D7                   sta   saveopc                *
0087 023A CC 06 24                   jts   crlf                   xmit crlf
0088 023D 8B 0B 09 3D                lrd   %A0,#msg27             Breakpoint removed
0089 0241 CC 06 36                   jts   txmsg                  xmit it.
0090
0091 0244 27 8F             norest   lda   #brkpoint              reset opcode save area.
0092 0246 48 06 D7                   sta   saveopc                *
0093
0094 0249 CC 06 24          nextcmd1 jts   crlf                   xmit crlf
0095 024C                   nextcmd  equ   *
0096 024C A1                         cla                          clear address
0097 024D 08 0F                      sta   %A2                    build area.
0098 024F 08 0E                      sta   %A2-1                  *
0099 0251 27 3E                      lda   #'>'                   xmit prompt.
0100 0253 CC 05 B6                   jts   txmit                  *
0101 0256 CC 05 E4                   jts   recv                   read monitor command.
0102 0259 CC 05 B6                   jts   txmit                  echo it.
0103
0104 025C 24 0D                      cmp   #cr                    simple cr ?
0105 025E C2 02 49                   jie   nextcmd1
0106 0261 24 3F                      cmp   #'?'
0107 0263 C2 04 4F                   jie   help                   help info.
0108 0266 25 DF                      and   #$DF                   make it upper case.
0109 0268 24 4C                      cmp   #'L'
0110 026A C2 02 A8                   jie   load                   load S19 file.
0111 026D 24 52                      cmp   #'R'
0112 026F C2 03 31                   jie   run                    run user pgm.
0113 0272 24 47                      cmp   #'G'
0114 0274 C2 03 6E                   jie   go                     continue user pgm.
0115 0277 24 44                      cmp   #'D'
0116 0279 C2 03 7B                   jie   display                display storage.
0117 027C 24 4D                      cmp   #'M'
0118 027E C2 03 E9                   jie   modify                 modify storage.
0119 0281 24 43                      cmp   #'C'
```

```
0120 0283 C2 04 66                  jie   clear                 clear storage.
0121 0286 24 42                     cmp   #'B'
0122 0288 C2 04 9D                  jie   break                 set breakpoint.
0123 028B 24 53                     cmp   #'S'
0124 028D C2 05 1B                  jie   ustatus               display user status.
0125 0290 24 50                     cmp   #'P'
0126 0292 C2 05 A4                  jie   setupc                set user program pc.
0127 0295 24 58                     cmp   #'X'
0128 0297 C2 02 2B                  jie   removbrk              remove breakpoint.
0129                        *   Invalid command.
0130 029A CC 06 24                  jts   crlf                  xmit invalid cmd msg.
0131 029D 8B 0B 06 FE               lrd   %A0,#msg02            *
0132 02A1 CC 06 36                  jts   txmsg                 *
0133
0134 02A4 EE A6                     jmp   *,nextcmd             lets try it again.
0135
0136                        *
0137                        *** Load a S19 file into storage.
0138                        *
0139 02A6 00 00                     fdb   0000                  return address
0140 02A8                      load     equ   *
0141 02A8 CC 06 24                  jts   crlf                  xmit invalid cmd msg.
0142 02AB 8B 0B 07 0F               lrd   %A0,#msg03            xmit start loading...
0143 02AF CC 06 36                  jts   txmsg                 *
0144
0145                        *   Load user program.
0146 02B2 CC 05 E4             load01   jts   recv              get first char of record.
0147 02B5 24 53                     cmp   #'S'                  found ?
0148 02B7 E3 F9                     jne   *,load01              branch if not.
0149 02B9 CC 05 E4                  jts   recv                  get S record type.
0150 02BC 24 31                     cmp   #'1'                  16 bits addr data record ?
0151 02BE E2 06                     jie   *,loaddata            branch if yes.
0152 02C0 24 39                     cmp   #'9'                  end record ?
0153 02C2 E2 57                     jie   *,loadend             branch if yes.
0154 02C4 EE EC                     jmp   *,load01              try again.
0155
0156                        *   Read 16 bits load address.
0157 02C6 CC 06 52             loaddata jts   hexinp            get total byte count.
0158 02C9 08 03                     sta   %R3                   set checksum.
0159 02CB 22 03                     sub   #$03                  cal number of data bytes.
```

```
0160 02CD 08 02                     sta   %R2                   save it.
0161
0162 02CF CC 06 52          load04   jts   hexinp                get byte load address.
0163 02D2 08 0E                     sta   %A2-1                 save higher address byte.
0164 02D4 09 03                     adm   %R3                   add to checksum.
0165 02D6 CC 06 52                  jts   hexinp                get lowest byte load address.
0166 02D9 08 0F                     sta   %A2                   save lower address byte.
0167 02DB 09 03                     adm   %R3                   add to checksum.
0168
0169                     *   Start loading data.
0170 02DD CC 06 52          load05   jts   hexinp                get data.
0171 02E0 18 0F                     sta   [%A2]                 store data.
0172 02E2 14 0F                     cmp   [%A2]                 store successfull ?
0173 02E4 E3 10                     jne   *,load06              if not: load error.
0174 02E6 09 03                     adm   %R3                   add to checksum.
0175 02E8 0C 0F                     ind   %A2                   incr store pointer.
0176 02EA 89 02 F0                  djnz  %R2,load05
0177
0178                     *   Checksum oke ? Issue message if not.
0179 02ED CC 06 52                  jts   hexinp                get check sum byte.
0180 02F0 00 03                     add   %R3                   add cal check sum.
0181 02F2 20 01                     add   #$01                  checksum oke ?
0182 02F4 E0 BC                     jiz   *,load01              if yes: get next record.
0183
0184                     *   Issue load failure message.
0185 02F6 8B 0B 07 43       load06   lrd   %A0,#msg05           Load program failed at xxxx
0186 02FA CC 06 36                  jts   txmsg                 tx it.
0187 02FD 07 0E                     lda   %A2-1                 get high byte failing addr.
0188 02FF CC 06 7D                  jts   hex2out               tx it.
0189 0302 07 0F                     lda   %A2                   get low byte failing addr.
0190 0304 CC 06 7D                  jts   hex2out               tx it.
0191 0307 CC 06 24                  jts   crlf
0192 030A 8B 0B 07 5E               lrd   %A0,#msg06            Flushing data input until CNTL/Q
0193 030E CC 06 36                  jts   txmsg                 tx it.
0194 0311 CC 05 E4          load07   jts   recv                  read input.
0195 0314 24 11                     cmp   #$11                  CNTL/Q ?
0196 0316 E3 F9                     jne   *,load07              loop if not.
0197 0318 CE 02 4C                  jmp   nextcmd               try it again.
0198
0199                     *   Load successfully ended, issue message.
```

```
0200 031B CC 05 E4          loadend  jts   recv                 read input
0201 031E 24 0D                      cmp   #cr                  till end
0202 0320 E3 F9                      jne   *,loadend            end of S9 record.
0203 0322 27 0D                      lda   #cr                  cr
0204 0324 CC 05 B6                   jts   txmit                xmit it
0205 0327 8B 0B 07 29                lrd   %A0,#msg04           Program successfully loaded.
0206 032B CC 06 36                   jts   txmsg                tx it.
0207 032E CE 02 4C                   jmp   nextcmd              play it again, Sam.
0208
0209                      *
0210                      *** Run user program.
0211                      *
0212 0331                    run     equ   *
0213 0331 CC 06 A8                   jts   hexaddr              get address
0214
0215 0334 CC 06 24                   jts   crlf                 xmit start user msg.
0216 0337 8B 0B 07 7F                lrd   %A0,#msg10           *
0217 033B CC 06 36                   jts   txmsg                *
0218 033E CC 06 24                   jts   crlf
0219 0341 A1                         cla
0220 0342 48 06 D5                   sta   intlvl0              switch to user
0221 0345 A7                         rtc                        *
0222 0346 86 06 D5                   swp   intlvl0              interrupt level 0.
0223 0349 DC 00 3F                   jts   [$003F]              Go for it !
0224
0225                      *   End user program has ended.
0226 034C                    endupgm  equ   *
0227 034C 48 01 00                   sta   $0100                save user pgm acc/rc.
0228 034F 27 03                      lda   #$03                 save user status and
0229 0351 48 01 01                   sta   $0101                switch to interrupt level 3
0230 0354 A7                         rtc                        *
0231 0355 86 01 01                   swp   $0101                *
0232
0233 0358 CC 06 24                   jts   crlf                 new line
0234 035B 8B 0B 07 98                lrd   %A0,#msg11           User program ended.
0235 035F CC 06 36                   jts   txmsg                *
0236 0362 47 01 00           endu01  lda   $0100                get user pgm rc.
0237 0365 CC 06 7D                   jts   hex2out              xmit it.
0238 0368 CC 06 24                   jts   crlf                 *
0239 036B CE 02 4C                   jmp   nextcmd              and now something completely...
```

```
0240
0241                    *
0242                    *** Continue user program after breakpoint.
0243                    *
0244 036E                  go       equ   *
0245 036E CC 06 24                  jts   crlf                  new clean line.
0246 0371 A7                        rtc                         restore user status reg.
0247 0372 86 01 01                  swp   $0101                 *
0248 0375 47 01 00                  lda   $0100                 restore user Acc
0249 0378 DE 01 03                  jmp   [$0103]               continue user pgm
0250
0251                    *
0252                    *** Display memory contents.
0253                    *
0254 037B                  display  equ   *
0255 037B CC 06 A8                  jts   hexaddr               get address
0256 037E 07 0F                     lda   %A2                   make
0257 0380 25 F0                     and   #$F0                  it
0258 0382 08 0F                     sta   %A2                   xxx0
0259 0384 CC 06 24                  jts   crlf
0260 0387 8B 0B 09 66               lrd   %A0,#msg30            header 0 1 2 ... E F
0261 038B CC 06 36                  jts   txmsg                 *
0262                    *   %A2 contains entered hex address.
0263 038E 8A 03 08                  ldr   %R3,#8                set outer loop counter.
0264 0391 07 0E            disp01   lda   %A2-1                 xmit memory
0265 0393 08 0C                     sta   %A1-1                 save it for ASCII print
0266 0395 CC 06 7D                  jts   hex2out               address
0267 0398 07 0F                     lda   %A2                   *
0268 039A 08 0D                     sta   %A1                   *
0269 039C CC 06 7D                  jts   hex2out               *
0270 039F 27 2D                     lda   #'-'                  xxxx -
0271 03A1 CC 06 17                  jts   space
0272 03A4 CC 05 B6                  jts   txmit
0273                    *   Print memory contents in hex.
0274 03A7 8A 02 10                  ldr   %R2,#16               set inner loop counter.
0275 03AA CC 06 17         disp02   jts   space
0276 03AD 17 0F                     lda   [%A2]                 get memory contents
0277 03AF 0C 0F                     ind   %A2                   incr memory pointer.
0278 03B1 CC 06 7D                  jts   hex2out               xmit it
0279 03B4 89 02 F3                  djnz  %R2,disp02
```

```
0280                    *   Print memory contents in ASCII.
0281 03B7 8A 02 10                 ldr   %R2,#16                 set inner loop counter.
0282 03BA CC 06 17                 jts   space                   seperate hex/ascii fields
0283 03BD CC 06 17                 jts   space                   *
0284 03C0 27 2A                    lda   #'*'                    *
0285 03C2 CC 05 B6                 jts   txmit
0286 03C5 17 0D            disp03   lda   [%A1]                   get memory contents
0287 03C7 24 20                    cmp   #$20                    printable ?
0288 03C9 E5 06                    jil   *,disp04                *
0289 03CB 24 7A                    cmp   #$7A                    *
0290 03CD E4 02                    jig   *,disp04                *
0291 03CF EE 02                    jmp   *,disp05                YES
0292 03D1 27 2E            disp04   lda   #'.'                    NO !
0293 03D3 CC 05 B6         disp05   jts   txmit                   xmit it.
0294 03D6 0C 0D                    ind   %A1                     incr memory pointer.
0295 03D8 89 02 EA                 djnz  %R2,disp03
0296
0297 03DB 27 2A                    lda   #'*'                    close it off.
0298 03DD CC 05 B6                 jts   txmit
0299 03E0 CC 06 24                 jts   crlf                    xmit start user msg.
0300 03E3 89 03 AB                 djnz  %R3,disp01              another line ?
0301
0302 03E6 CE 02 4C                 jmp   nextcmd                 what's next ?
0303
0304                    *
0305                    *** Modify memory contents.
0306                    *
0307 03E9                  modify   equ   *
0308 03E9 CC 06 A8                 jts   hexaddr                 get address
0309                    *   %A2 contains entered hex address.
0310 03EC CC 06 24         mod01    jts   crlf
0311 03EF 07 0E                    lda   %A2-1                   xmit memory
0312 03F1 CC 06 7D                 jts   hex2out                 address
0313 03F4 07 0F                    lda   %A2                     *
0314 03F6 CC 06 7D                 jts   hex2out                 *
0315 03F9 CC 06 17                 jts   space                   xmit space
0316 03FC 17 0F                    lda   [%A2]                   get memory contents
0317 03FE CC 06 7D                 jts   hex2out                 xmit it
0318 0401 CC 06 17                 jts   space                   xmit space
0319
```

```
0320 0404 CC 05 E4          mod02    jts   recv                get input.
0321 0407 24 0D                      cmp   #cr                 CR ?
0322 0409 E2 33                      jie   *,mod07             next location
0323 040B 24 1B             mod03    cmp   #escape             Escape ?
0324 040D C2 02 49                   jie   nextcmd1            that's all for today.
0325 0410 24 71                      cmp   #'q'                quit ?
0326 0412 C2 02 49                   jie   nextcmd1            that's all for today.
0327 0415 CC 05 B6                   jts   txmit               echo input.
0328
0329 0418 24 3A                      cmp   #$3A                0-9 or A-F ?
0330 041A E5 02                      jil   *,mod04             jump if 0-9
0331 041C 20 09                      add   #$09                correction
0332 041E 25 0F             mod04    and   #$0F                remove high order bits.
0333 0420 38 07                      sta   *,mod06+1           save it.
0334 0422 07 02             mod05    lda   %R2                 get modify value.
0335 0424 A2                         ral                       move them to high order.
0336 0425 A2                         ral
0337 0426 A2                         ral
0338 0427 A2                         ral
0339 0428 20 00             mod06    add   #$00                merge it with input.
0340 042A 08 02                      sta   %R2                 save it again.
0341 042C CC 05 E4                   jts   recv                get input.
0342 042F 24 0D                      cmp   #cr
0343 0431 E3 D8                      jne   *,mod03             next location
0344 0433 CC 05 B6                   jts   txmit               echo it.
0345                      *   Store modified value.
0346 0436 07 02                      lda   %R2                 get modify value and
0347 0438 18 0F                      sta   [%A2]               store it
0348 043A 14 0F                      cmp   [%A2]               verify store.
0349 043C E3 04                      jne   *,mod08             not ok ?
0350 043E 0C 0F             mod07    ind   %A2                 bump address pointer.
0351 0440 EE AA                      jmp   *,mod01             loop
0352                      *   Modify not possible.
0353 0442 CC 06 24          mod08    jts   crlf                crlf.
0354 0445 8B 0B 07 B5                lrd   %A0,#msg15          Verify after modify failed.
0355 0449 CC 06 36                   jts   txmsg               tx it.
0356 044C CE 02 4C                   jmp   nextcmd             play it again Sam.
0357
0358                      *
0359                      *** Transmit help information.
```

```
0360                        *
0361 044F                      help     equ   *
0362 044F CC 06 24                      jts   crlf                  new line.
0363 0452 8B 0B 06 E0                   lrd   %A0,#msg01+6          Version...
0364 0456 CC 06 36                      jts   txmsg                 *
0365 0459 CC 06 24                      jts   crlf                  new line.
0366 045C 8B 0B 07 E6                   lrd   %A0,#msg20            Valid commands...
0367 0460 CC 06 36                      jts   txmsg                 *
0368 0463 CE 02 4C                      jmp   nextcmd               play it again Sam.
0369
0370                        *
0371                        *** Clear memory.
0372                        *
0373 0466                      clear    equ   *
0374 0466 CC 06 24                      jts   crlf                  new line.
0375 0469 8B 0B 07 D0                   lrd   %A0,#msg17            Are you sure ?
0376 046D CC 06 36                      jts   txmsg                 *
0377 0470 CC 05 E4                      jts   recv                  get input
0378 0473 CC 05 B6                      jts   txmit                 echo input.
0379 0476 25 DF                         and   #$DF                  make upper case.
0380 0478 24 59                         cmp   #'Y'                  confirmed ?
0381 047A C3 02 49                      jne   nextcmd1              NO.
0382                        *   Clear user registers.
0383 047D 8A 01 10                      ldr   %R1,#16               16 registers.
0384 0480 A1                            cla
0385 0481 48 00 00             clr00    sta   $0000                 intlvl0 registers.
0386 0484 4B 04 83                      inc   clr00+2               bump pointer.
0387 0487 89 01 F7                      djnz  %R1,clr00             loop.
0388                        *   Clear user status register.
0389 048A 48 01 01                      sta   $0101                 clear ustatus.
0390                        *   Clear memory.
0391 048D 58 04 9A             clr01    sta   [clrpntr+1]           clear it.
0392 0490 3C 08                         ind   *,clrpntr+1           bump pointer.
0393 0492 3E 08                         dcd   *,clrcnt+1
0394 0494 E7 F7                         jnc   *,clr01
0395 0496 CE 80 00                      jmp   $8000                 restart monitor.
0396
0397 0499 04 9D                clrpntr  fdb   *+4                   start location.
0398 049B 80 00                clrcnt   fdb   $8000                number of bytes.
0399
```

```
0400                    *
0401                    *** Set breakpoint.
0402                    *
0403 049D                  break    equ   *
0404 049D CC 06 A8                  jts   hexaddr                 get breakpoint address
0405                    *   Check if a breakpoint is allready set.
0406 04A0 47 06 D7                  lda   saveopc                 allready set ?
0407 04A3 24 8F                     cmp   #brkpoint               *
0408 04A5 E2 19                     jie   *,bpfree                *
0409                    *   Issue error message, BP allready set.
0410 04A7 CC 06 24                  jts   crlf                    new line.
0411 04AA 8B 0B 09 20               lrd   %A0,#msg26              BP allready set at xxxx.
0412 04AE CC 06 36                  jts   txmsg                   tx it.
0413 04B1 47 06 D8                  lda   savebpa                 get high byte bp addr.
0414 04B4 CC 06 7D                  jts   hex2out                 tx it.
0415 04B7 47 06 D9                  lda   savebpa+1               get low byte bp addr.
0416 04BA CC 06 7D                  jts   hex2out                 tx it.
0417 04BD CE 02 49                  jmp   nextcmd1                play it again Sam.
0418
0419 04C0 17 0F            bpfree   lda   [%A2]                   get current opcode.
0420 04C2 48 06 D7                  sta   saveopc                 save it.
0421 04C5 27 8F                     lda   #brkpoint               set breakpoint.
0422 04C7 18 0F                     sta   [%A2]                   *
0423 04C9 14 0F                     cmp   [%A2]                   verify ok ?
0424 04CB E3 0D                     jne   *,notoke                no.
0425                    *   Breakpoint set, save breakpoint address.
0426 04CD 07 0F                     lda   %A2                     get breakpoint
0427 04CF 48 06 D9                  sta   savebpa+1               address
0428 04D2 07 0E                     lda   %A2-1                   and
0429 04D4 48 06 D8                  sta   savebpa                 save it.
0430 04D7 CE 02 49                  jmp   nextcmd1
0431                    *   Issue error message.
0432 04DA 48 06 D7         notoke   sta   saveopc                 set bp free again.
0433 04DD CC 06 24                  jts   crlf                    crlf.
0434 04E0 8B 0B 09 06               lrd   %A0,#msg25              Unable to set breakpoint.
0435 04E4 CC 06 36                  jts   txmsg                   tx it.
0436 04E7 CE 02 4C                  jmp   nextcmd                 play it again Sam.
0437
0438 04EA                  brkdet   equ   *
0439                    *   Switch to monitor interrupt level (3).
```

```
0440 04EA 27 03                     lda   #$03                    switch to
0441 04EC 48 06 D6                  sta   intlvl3                 interrupt level 3
0442 04EF A7                        rtc                           *
0443 04F0 86 06 D6                  swp   intlvl3                 *
0444                      *   Saved PC points 2 byte beyond breakpoint. Subtract 2 correction.
0445 04F3 A6                        stc                           set carry.
0446 04F4 47 01 03                  lda   $0103                   get low byte
0447 04F7 23 02                     sbc   #$02                    minus two.
0448 04F9 48 01 03                  sta   $0103                   save it.
0449 04FC E6 03                     jic   *,brk01                 borrow ?
0450 04FE 4D 01 02                  dcr   $0102                   yes, high byte minus one.
0451                      *   Restore original opcode in user pgm.
0452 0501 47 06 D7          brk01   lda   saveopc                 restore orignal opcode
0453 0504 24 8F                     cmp   #brkpoint               is this correct ?
0454 0506 C2 02 49                  jie   nextcmd1                jump if not.
0455 0509 58 01 03                  sta   [$0103]                 restore user pgm opcode.
0456 050C 27 8F                     lda   #brkpoint               set breakpoint cleared.
0457 050E 48 06 D7                  sta   saveopc                 *
0458
0459 0511 CC 06 24                  jts   crlf                    new line.
0460 0514 8B 0B 09 52               lrd   %A0,#msg29              Breakpoint entered.
0461 0518 CC 06 36                  jts   txmsg                   *
0462
0463 051B                   ustatus  equ   *
0464 051B CC 06 24                  jts   crlf                    new line.
0465 051E 8B 0B 09 66               lrd   %A0,#msg30              Regs =
0466 0522 CC 06 36                  jts   txmsg                   *
0467 0525 8B 0B 09 9C               lrd   %A0,#msg30a             Regs =
0468 0529 CC 06 36                  jts   txmsg                   *
0469 052C 47 01 01                  lda   $0101                   get user int level
0470 052F A2                        ral                           move
0471 0530 A2                        ral                           intlvl
0472 0531 A2                        ral                           to
0473 0532 A2                        ral                           high order.
0474 0533 38 03                     sta   *,brk05+3               set register range addr.
0475 0535 8B 0F 00 00       brk05   lrd   %A2,#$0000              set to intlvl 0 regs.
0476 0539 8A 02 10                  ldr   %R2,#16                 set loop counter.
0477 053C 17 0F             brk06   lda   [%A2]                   get register contents
0478 053E 0C 0F                     ind   %A2                     incr register pointer.
0479 0540 CC 06 7D                  jts   hex2out                 xmit it
```

```
0480 0543 CC 06 17                  jts   space
0481 0546 89 02 F3                  djnz  %R2,brk06             end of hex part of line ?
0482
0483 0549 CC 06 24                  jts   crlf                  new line.
0484 054C 8B 0B 09 A5               lrd   %A0,#msg30b           PC =
0485 0550 CC 06 36                  jts   txmsg                 *
0486 0553 47 01 02                  lda   $0102                 PC high
0487 0556 CC 06 7D                  jts   hex2out               xmit it.
0488 0559 47 01 03                  lda   $0103                 PC low
0489 055C CC 06 7D                  jts   hex2out               xmit it.
0490
0491 055F CC 06 36                  jts   txmsg                 ACC =
0492 0562 47 01 00                  lda   $0100
0493 0565 CC 06 7D                  jts   hex2out               xmit it.
0494
0495 0568 CC 06 36                  jts   txmsg                 Status =
0496 056B 47 01 01                  lda   $0101
0497 056E 8A 02 04                  ldr   %R2,#4                set loop counter.
0498 0571 A2                brk02   ral                         get bit.
0499 0572 38 09                     sta   *,brk04+1             save it for later.
0500 0574 27 30                     lda   #'0'                  '0'
0501 0576 E9 01                     jns   *,brk03               or
0502 0578 AC                        ina                         '1'
0503 0579 CC 05 B6          brk03   jts   txmit                 out ?
0504 057C 27 00             brk04   lda   #$00                  restore status.
0505 057E 89 02 F0                  djnz  %R2,brk02             loop.
0506
0507 0581 CC 06 36                  jts   txmsg                 Intlvl =
0508 0584 47 01 01                  lda   $0101
0509 0587 CC 06 93                  jts   hexLout               xmit it.
0510
0511 058A CC 06 36                  jts   txmsg                 BP =
0512 058D 47 06 D7                  lda   saveopc               breakpoint set ?
0513 0590 24 8F                     cmp   #brkpoint             *
0514 0592 C2 02 49                  jie   nextcmd1              jump if not.
0515                      * BP is set, print it.
0516 0595 47 06 D8                  lda   savebpa               get high byte bp addr.
0517 0598 CC 06 7D                  jts   hex2out               tx it.
0518 059B 47 06 D9                  lda   savebpa+1             get low byte bp addr.
0519 059E CC 06 7D                  jts   hex2out               tx it.
```

```
0520
0521 05A1 CE 02 49                  jmp   nextcmd1               next...
0522
0523                        *
0524                        *** Set user program PC.
0525                        *
0526 05A4                      setupc   equ   *
0527 05A4 CC 06 A8                      jts   hexaddr              get user pgm PC address
0528 05A7 07 0F                         lda   %A2                  * low
0529 05A9 48 01 03                      sta   $0103                *
0530 05AC 07 0E                         lda   %A2-1                * high
0531 05AE 48 01 02                      sta   $0102                *
0532 05B1 CE 02 49                      jmp   nextcmd1             next...
0533
0534                        *
0535                        *** Transmit one byte (in accumulator) subroutine.
0536                        *
0537 05B4 00 00                         fdb   0000                 return address
0538 05B6                      txmit    equ   *
0539 05B6 AB                            rti                        issue start puls
0540
0541 05B7 8A 01 09                      ldr   %R1,#9               set loop counter.
0542
0543 05BA 8A 00 09                      ldr   %R0,#09              wait one startbit time.
0544 05BD 0D 00                delay01  dcr   %R0                  *
0545 05BF C7 05 BD                      jnc   delay01              *
0546
0547 05C2 A5                   loop01   rrs                        shift out
0548 05C3 C8 05 C9                      jis   set
0549 05C6 AB                   reset    rti                        Xmit "0"
0550 05C7 EE 03                         jmp   *,cont01
0551 05C9 AA                   set      sti                        Xmit "1"
0552 05CA EE 00                         jmp   *,cont01
0553
0554 05CC 8A 00 08             cont01   ldr   %R0,#08              wait one bit time.
0555 05CF 0D 00                delay02  dcr   %R0                  *
0556 05D1 C7 05 CF                      jnc   delay02              *
0557
0558 05D4 89 01 EB                      djnz  %R1,loop01           Decr loop counter
0559
```

```
0560 05D7 AA                        sti                             Issue stop puls
0561
0562 05D8 8A 00 0E                  ldr   %R0,#08+6                 Wait one startbit time.
0563 05DB 0D 00            delay03  dcr   %R0
0564 05DD C7 05 DB                  jnc   delay03
0565
0566 05E0 ED D4                     jfs   *,txmit                   Return to caller.
0567
0568
0569                    *
0570                    *** Receive one byte (in accumulator) subroutine.
0571                    *
0572 05E2 00 00                     fdb   0000                      Return address
0573 05E4                  recv     equ   *
0574 05E4 CF 05 E9         loop03   jif   wait02                    wait for start bit.
0575 05E7 EE FB                     jmp   *,loop03                  *
0576 05E9 CF 05 E9         wait02   jif   wait02                    *
0577
0578                    * We have a possible start bit.
0579 05EC 8A 00 02         cont03   ldr   %R0,#02                   Wait half bit time.
0580 05EF 0D 00            delay05  dcr   %R0                       *
0581 05F1 C7 05 EF                  jnc   delay05                   *
0582
0583                    * Do we still have a start bit or was is some noise ?
0584 05F4 CF 05 E4                  jif   recv                      it was noise.
0585
0586 05F7 8A 01 09                  ldr   %R1,#1+8                  Set loop counter.
0587
0588 05FA                  nextbit  equ   *
0589 05FA 8A 00 07                  ldr   %R0,#07                   Wait one bit time.
0590 05FD 0D 00            delay06  dcr   %R0                       *
0591 05FF C7 05 FD                  jnc   delay06                   *
0592
0593 0602 EF 03                     jif   *,onebit                  Sample received bit.
0594 0604 A9                        rts                             Set "0"
0595 0605 EE 03                     jmp   *,shiftbit
0596 0607 A8               onebit   sts                             Set "1"
0597 0608 EE 00                     jmp   *,shiftbit
0598 060A A5               shiftbit rrs
0599
```

```
0600 060B 89 01 EC                djnz  %R1,nextbit           Decr loop counter
0601
0602 060E CF 06 13          loop05  jif   exit02              wait for stop bit.
0603 0611 EE FB                     jmp   *,loop05            *
0604
0605 0613 ED CF             exit02  jfs   *,recv              Return to caller.
0606
0607
0608                      *
0609                      *** Transmit a space character.
0610                      *
0611 0615 00 00                     fdb   0000                Return address
0612 0617                   space   equ   *
0613 0617 38 06                     sta   *,save01+1          save acc.
0614 0619 27 20                     lda   #' '                space
0615 061B CC 05 B6                  jts   txmit               send it.
0616 061E 27 00             save01  lda   #$00                Restore acc.
0617 0620 ED F5                     jfs   *,space
0618
0619
0620                      *
0621                      *** Transmit a CR and a LF.
0622                      *
0623 0622 00 00                     fdb   0000                Return address
0624 0624                   crlf    equ   *
0625 0624 38 0B                     sta   *,save02+1          save acc.
0626 0626 27 0D                     lda   #cr                 cr.
0627 0628 CC 05 B6                  jts   txmit               send it.
0628 062B 27 0A                     lda   #lf                 lf.
0629 062D CC 05 B6                  jts   txmit               send it.
0630 0630 27 00             save02  lda   #$00                Restore acc.
0631 0632 ED F0                     jfs   *,crlf
0632
0633
0634                      *
0635                      *** Transmit a message (till 0x00) pointed by %A0.
0636                      *
0637 0634 00 00                     fdb   0000                return address
0638 0636                   txmsg   equ   *
0639 0636 38 15                     sta   *,save03+1          save acc.
```

```
0640 0638 EE 05                    jmp   *,cont06
0641
0642 063A CC 05 B6          loop06 jts   txmit                 transmit it.
0643 063D 0C 0B                    ind   %A0                   increment pointer.
0644 063F 17 0B             cont06 lda   [%A0]                 get character.
0645 0641 24 23                    cmp   #'#'                  no crlf ?
0646 0643 E2 05                    jie   *,cont07
0647 0645 E1 F3                    jnz   *,loop06              end of message ?
0648 0647 CC 06 24                 jts   crlf                  add cr/lf
0649 064A 0C 0B             cont07 ind   %A0                   bump over # (if applicable)
0650 064C 27 00             save03 lda   #$00                  restore acc.
0651 064E ED E6                    jfs   *,txmsg
0652
0653
0654                       *
0655                       *** Receive two hex characters in accumulator.
0656                       *
0657 0650 00 00                    fdb   0000                  return address
0658 0652                  hexinp  equ   *
0659 0652 CC 05 E4                 jts   recv                  read input (high order)
0660 0655 24 0D                    cmp   #cr                   cr ?
0661 0657 E2 20                    jie   *,hex03               jump if yes.
0662 0659 24 71                    cmp   #'q'                  quit ?
0663 065B C2 02 49                 jie   nextcmd1              jump if yes. (it's dirty :-)
0664 065E 24 3A                    cmp   #$3A                  0-9 or A-F ?
0665 0660 E5 02                    jil   *,hex01               jump if 0-9
0666 0662 20 09                    add   #$09                  correction
0667 0664 25 0F             hex01  and   #$0F                  remove high order bits.
0668 0666 A2                       ral                         move them to high order.
0669 0667 A2                       ral
0670 0668 A2                       ral
0671 0669 A2                       ral
0672 066A 38 0C                    sta   *,save04+1            save it.
0673 066C CC 05 E4                 jts   recv                  read input (low order)
0674 066F 24 3A                    cmp   #$3A                  0-9 or A-F ?
0675 0671 E5 02                    jil   *,hex02               jump if 0-9
0676 0673 20 09                    add   #$09                  correction
0677 0675 25 0F             hex02  and   #$0F                  remove high order bits.
0678 0677 20 00             save04 add   #$00                  make it complete.
0679 0679 ED D7             hex03  jfs   *,hexinp              return to caller.
```

```
0680
0681
0682                    *
0683                    *** Transmit accumulator (ligh, low, both) in hex-dec.
0684                    *
0685 067B 00 00                fdb   0000                    return address
0686 067D                  hex2out  equ   *
0687 067D EC 06                 jts   *,hexHout              transmit high part acc.
0688 067F EC 12                 jts   *,hex1out              transmit low part acc.
0689 0681 ED FA                 jfs   *,hex2out
0690
0691 0683 00 00                 fdb   0000                   return address
0692 0685                  hexHout  equ   *
0693 0685 38 07                 sta   *,save13+1             save acc.
0694 0687 A4                    rar                          shift 4 bits high -> low
0695 0688 A4                    rar
0696 0689 A4                    rar
0697 068A A4                    rar
0698 068B EC 06                 jts   *,hex1out              transmit it as hex.
0699 068D 27 00            save13   lda   #$00                restore acc.
0700 068F ED F4                 jfs   *,hexHout
0701
0702 0691 00 00                 fdb   0000                   return address
0703 0693                  hex1out  equ   *
0704 0693                  hexLout  equ   *
0705 0693 38 0E                 sta   *,save14+1             save acc.
0706 0695 25 0F                 and   #$0F                   remove any junk bits
0707 0697 20 30                 add   #$30
0708 0699 24 3A                 cmp   #$3A                   0-9 or A-F ?
0709 069B E5 02                 jil   *,cont15
0710 069D 20 07                 add   #$07                   make it A-F.
0711 069F CC 05 B6         cont15   jts   txmit                xmit it.
0712 06A2 27 00            save14   lda   #$00                restore acc.
0713 06A4 ED ED                 jfs   *,hex1out
0714
0715
0716                    *
0717                    *** Read hex-decimal address.
0718                    *
0719 06A6 00 00                 fdb   0000                   return address
```

```
0720 06A8                      hexaddr  equ   *
0721 06A8 CC 05 E4                      jts   recv                    read input
0722 06AB 24 0D                         cmp   #$0D                    cr ?
0723 06AD E2 24                         jie   *,hex10                 jump if yes.
0724 06AF CC 05 B6                      jts   txmit
0725 06B2 24 3A                         cmp   #$3A                    0-9 or A-F ?
0726 06B4 E5 02                         jil   *,hex11                 jump if 0-9
0727 06B6 20 09                         add   #$09                    correction
0728 06B8 25 0F                hex11    and   #$0F                    remove high order bits.
0729 06BA 38 12                         sta   *,save12+1              save it.
0730                         *   Shift address 4 bits to the left.
0731 06BC 8A 02 04                      ldr   %R2,#4                  set loop counter.
0732 06BF A9                   loop10   rts                           reset shift bit.
0733 06C0 07 0F                         lda   %A2                     get lower byte
0734 06C2 A3                            rls                           move higher bit
0735 06C3 08 0F                         sta   %A2                     in
0736 06C5 07 0E                         lda   %A2-1                   lower order
0737 06C7 A3                            rls                           of
0738 06C8 08 0E                         sta   %A2-1                   higher order byte.
0739 06CA 89 02 F2                      djnz  %R2,loop10              loop
0740
0741 06CD 27 00                save12   lda   #00                     reload input char.
0742 06CF 09 0F                         adm   %A2                     insert it in address.
0743 06D1 EE D5                         jmp   *,hexaddr
0744
0745 06D3 ED D3                hex10    jfs   *,hexaddr               return to caller.
0746
0747 06D5                      intlvl0  rmb   1                       used to switch to intlvl 0
0748 06D6                      intlvl3  rmb   1                       used to switch to intlvl 3
0749 06D7 8F                   saveopc  fcb   brkpoint                save opc area.
0750 06D8                      savebpa  rmb   2                       save breakpoint addr.
0751
0752
0753 06DA 48 4A 53 32 32 20    msg01    fcc   'HJS22 Monitor Version 1.5 running.#'
          4D 6F 6E 69 74 6F
          72 20 56 65 72 73
          69 6F 6E 20 31 2E
          35 20 72 75 6E 6E
          69 6E 67 2E 23
0754 06FD 00                            fcb   null
```

```
0755 06FE 49 6E 76 61 6C 69 msg02    fcc   'Invalid command.'
          64 20 63 6F 6D 6D
          61 6E 64 2E
0756 070E 00                          fcb   null
0757 070F 53 74 61 72 74 20 msg03    fcc   'Start loading S19 file...'
          6C 6F 61 64 69 6E
          67 20 53 31 39 20
          66 69 6C 65 2E 2E
          2E
0758 0728 00                          fcb   null
0759 0729 46 69 6C 65 20 73 msg04    fcc   'File successfully loaded.'
          75 63 63 65 73 73
          66 75 6C 6C 79 20
          6C 6F 61 64 65 64
          2E
0760 0742 00                          fcb   null
0761 0743 4C 6F 61 64 20 65 msg05    fcc   'Load error near address #.'
          72 72 6F 72 20 6E
          65 61 72 20 61 64
          64 72 65 73 73 20
          23 2E
0762 075D 00                          fcb   null
0763 075E 46 6C 75 73 68 69 msg06    fcc   'Flushing input data till CNTL/Q.'
          6E 67 20 69 6E 70
          75 74 20 64 61 74
          61 20 74 69 6C 6C
          20 43 4E 54 4C 2F
          51 2E
0764 077E 00                          fcb   null
0765 077F 53 74 61 72 74 69 msg10    fcc   'Starting user program...'
          6E 67 20 75 73 65
          72 20 70 72 6F 67
          72 61 6D 2E 2E 2E
0766 0797 00                          fcb   null
0767 0798 55 73 65 72 20 70 msg11    fcc   'User program ended with RC=#'
          72 6F 67 72 61 6D
          20 65 6E 64 65 64
          20 77 69 74 68 20
          52 43 3D 23
0768 07B4 00                          fcb   null
```

```
0769 07B5 55 6E 61 62 6C 65 msg15    fcc   'Unable to modify storage !'
          20 74 6F 20 6D 6F
          64 69 66 79 20 73
          74 6F 72 61 67 65
          20 21
0770 07CF 00                        fcb   null
0771 07D0 43 6C 65 61 72 20 msg17    fcc   'Clear memory [y,N]? #'
          6D 65 6D 6F 72 79
          20 5B 79 2C 4E 5D
          3F 20 23
0772 07E5 00                        fcb   null
0773 07E6 56 61 6C 69 64 20 msg20    fcc   'Valid commands:'
          63 6F 6D 6D 61 6E
          64 73 3A
0774 07F5 0D 0A                     fcb   cr,lf
0775 07F7 20 20 44 20 78 78         fcc   '  D xxxx - display memory.'
          78 78 20 2D 20 64
          69 73 70 6C 61 79
          20 6D 65 6D 6F 72
          79 2E
0776 0811 0D 0A                     fcb   cr,lf
0777 0813 20 20 4D 20 78 78         fcc   '  M xxxx - modify memory.'
          78 78 20 2D 20 6D
          6F 64 69 66 79 20
          6D 65 6D 6F 72 79
          2E
0778 082C 0D 0A                     fcb   cr,lf
0779 082E 20 20 42 20 78 78         fcc   '  B xxxx - set breakpoint.'
          78 78 20 2D 20 73
          65 74 20 62 72 65
          61 6B 70 6F 69 6E
          74 2E
0780 0848 0D 0A                     fcb   cr,lf
0781 084A 20 20 50 20 78 78         fcc   '  P xxxx - set user pgm PC.'
          78 78 20 2D 20 73
          65 74 20 75 73 65
          72 20 70 67 6D 20
          50 43 2E
0782 0865 0D 0A                     fcb   cr,lf
0783 0867 20 20 52 20 78 78         fcc   '  R xxxx - run user program.'
```

```
          78 78 20 2D 20 72
          75 6E 20 75 73 65
          72 20 70 72 6F 67
          72 61 6D 2E
0784 0883 0D 0A                     fcb   cr,lf
0785 0885 20 20 47 20 2D 20         fcc   '  G - continue user program.'
          63 6F 6E 74 69 6E
          75 65 20 75 73 65
          72 20 70 72 6F 67
          72 61 6D 2E
0786 08A1 0D 0A                     fcb   cr,lf
0787 08A3 20 20 4C 20 2D 20         fcc   '  L - load user program.'
          6C 6F 61 64 20 75
          73 65 72 20 70 72
          6F 67 72 61 6D 2E
0788 08BB 0D 0A                     fcb   cr,lf
0789 08BD 20 20 53 20 2D 20         fcc   '  S - user program status.'
          75 73 65 72 20 70
          72 6F 67 72 61 6D
          20 73 74 61 74 75
          73 2E
0790 08D7 0D 0A                     fcb   cr,lf
0791 08D9 20 20 58 20 2D 20         fcc   '  X - clear breakpoint.'
          63 6C 65 61 72 20
          62 72 65 61 6B 70
          6F 69 6E 74 2E
0792 08F0 0D 0A                     fcb   cr,lf
0793 08F2 20 20 43 20 2D 20         fcc   '  C - clear memory.'
          63 6C 65 61 72 20
          6D 65 6D 6F 72 79
          2E
0794 0905 00                        fcb   null
0795 0906 55 6E 61 62 6C 65 msg25   fcc   'Unable to set breakpoint.'
          20 74 6F 20 73 65
          74 20 62 72 65 61
          6B 70 6F 69 6E 74
          2E
0796 091F 00                        fcb   null
0797 0920 42 72 65 61 6B 70 msg26   fcc   'Breakpoint allready set at #'
          6F 69 6E 74 20 61
```

```
          6C 6C 72 65 61 64
          79 20 73 65 74 20
          61 74 20 23
0798 093C 00                          fcb   null
0799 093D 42 72 65 61 6B 70 msg27     fcc   'Breakpoint cleared.#'
          6F 69 6E 74 20 63
          6C 65 61 72 65 64
          2E 23
0800 0951 00                          fcb   null
0801 0952 42 72 65 61 6B 70 msg29     fcc   'Breakpoint entered.'
          6F 69 6E 74 20 65
          6E 74 65 72 65 64
          2E
0802 0965 00                          fcb   null
0803 0966 20 20 20 20 20 20 msg30     fcc   '       0  1  2  3  4  5  6  7'
          20 30 20 20 31 20
          20 32 20 20 33 20
          20 34 20 20 35 20
          20 36 20 20 37
0804 0983 20 20 38 20 20 39           fcc   '  8  9  A  B  C  D  E  F'
          20 20 41 20 20 42
          20 20 43 20 20 44
          20 20 45 20 20 46
0805 099B 00                          fcb   null
0806 099C 52 65 67 73 20 2D msg30a    fcc   'Regs - #'
          20 23
0807 09A4 00                          fcb   null
0808 09A5 50 43 2D 23 20 20 msg30b    fcc   'PC-#  Acc-#  Status-#  Intlvl-#  BP-#'
          41 63 63 2D 23 20
          20 53 74 61 74 75
          73 2D 23 20 20 49
          6E 74 6C 76 6C 2D
          23 20 20 42 50 2D
          23
0809 09CA 00                          fcb   null
0810
0811                                end
0812
```

```
A0          000B  *0045 0078 0088 0131 0142 0185 0192 0205 0216 0234
                        0260 0354 0363 0366 0375 0411 0434 0460 0465 0467 0484
                        0643 0644 0649
A1          000D  *0046 0265 0268 0286 0294
A2          000F  *0047 0097 0098 0163 0166 0171 0172 0175 0187 0189
                        0256 0258 0264 0267 0276 0277 0311 0313 0316 0347 0348
                        0350 0419 0422 0423 0426 0428 0475 0477 0478 0528 0530
                        0733 0735 0736 0738 0742
R0          0000  *0034 0543 0544 0554 0555 0562 0563 0579 0580 0589
                        0590
R1          0001  *0035 0383 0387 0541 0558 0586 0600
R2          0002  *0036 0160 0176 0274 0279 0281 0295 0334 0340 0346
                        0476 0481 0497 0505 0731 0739
R3          0003  *0037 0158 0164 0167 0174 0180 0263 0300
R4          0004  *0038
R5          0005  *0039
R6          0006  *0040
R7          0007  *0041
R8          0008  *0042
R9          0009  *0043
bpfree      04C0  *0419 0408
break       049D  *0403 0122
brk01       0501  *0452 0449
brk02       0571  *0498 0505
brk03       0579  *0503 0501
brk04       057C  *0504 0499
brk05       0535  *0475 0474
brk06       053C  *0477 0481
brkdet      04EA  *0438 0068
brkpoint    008F  *0053 0082 0085 0091 0407 0421 0453 0456 0513 0749
clear       0466  *0373 0120
clr00       0481  *0385 0386 0387
clr01       048D  *0391 0394
clrcnt      049B  *0398 0393
clrpntr     0499  *0397 0391 0392
cont01      05CC  *0554 0550 0552
cont03      05EC  *0579
cont06      063F  *0644 0640
cont07      064A  *0649 0646
```

```
cont15     069F  *0711 0709
cr         000D  *0050 0104 0201 0203 0321 0342 0626 0660 0774 0776
                       0778 0780 0782 0784 0786 0788 0790 0792
crlf       0624  *0624 0076 0077 0087 0094 0130 0141 0191 0215 0218
                       0233 0238 0245 0259 0299 0310 0353 0362 0365 0374 0410
                       0433 0459 0464 0483 0631 0648
delay01    05BD  *0544 0545
delay02    05CF  *0555 0556
delay03    05DB  *0563 0564
delay05    05EF  *0580 0581
delay06    05FD  *0590 0591
disp01     0391  *0264 0300
disp02     03AA  *0275 0279
disp03     03C5  *0286 0295
disp04     03D1  *0292 0288 0290
disp05     03D3  *0293 0291
display    037B  *0254 0116
endu01     0362  *0236
endupgm    034C  *0226
escape     001B  *0049 0323
exit02     0613  *0605 0602
eyecatch   020E  *0063 0059
go         036E  *0244 0114
help       044F  *0361 0107
hex01      0664  *0667 0665
hex02      0675  *0677 0675
hex03      0679  *0679 0661
hex10      06D3  *0745 0723
hex11      06B8  *0728 0726
hex1out    0693  *0703 0688 0698 0713
hex2out    067D  *0686 0188 0190 0237 0266 0269 0278 0312 0314 0317
                       0414 0416 0479 0487 0489 0493 0517 0519 0689
hexHout    0685  *0692 0687 0700
hexLout    0693  *0704 0509
hexaddr    06A8  *0720 0213 0255 0308 0404 0527 0743 0745
hexinp     0652  *0658 0157 0162 0165 0170 0179 0679
intlvl0    06D5  *0747 0220 0222
intlvl3    06D6  *0748 0073 0075 0441 0443
lf         000A  *0051 0628 0774 0776 0778 0780 0782 0784 0786 0788
                       0790 0792
```

```
load       02A8  *0140 0110
load01     02B2  *0146 0148 0154 0182
load04     02CF  *0162
load05     02DD  *0170 0176
load06     02F6  *0185 0173
load07     0311  *0194 0196
loaddata   02C6  *0157 0151
loadend    031B  *0200 0153 0202
loop01     05C2  *0547 0558
loop03     05E4  *0574 0575
loop05     060E  *0602 0603
loop06     063A  *0642 0647
loop10     06BF  *0732 0739
mod01      03EC  *0310 0351
mod02      0404  *0320
mod03      040B  *0323 0343
mod04      041E  *0332 0330
mod05      0422  *0334
mod06      0428  *0339 0333
mod07      043E  *0350 0322
mod08      0442  *0353 0349
modify     03E9  *0307 0118
monitor    0200  *0058
monstart   0215  *0071 0065
msg01      06DA  *0753 0078 0363
msg02      06FE  *0755 0131
msg03      070F  *0757 0142
msg04      0729  *0759 0205
msg05      0743  *0761 0185
msg06      075E  *0763 0192
msg10      077F  *0765 0216
msg11      0798  *0767 0234
msg15      07B5  *0769 0354
msg17      07D0  *0771 0375
msg20      07E6  *0773 0366
msg25      0906  *0795 0434
msg26      0920  *0797 0411
msg27      093D  *0799 0088
msg29      0952  *0801 0460
msg30      0966  *0803 0260 0465
```

```
msg30a     099C  *0806 0467
msg30b     09A5  *0808 0484
nextbit    05FA  *0588 0600
nextcmd    024C  *0095 0134 0197 0207 0239 0302 0356 0368 0436
nextcmd1   0249  *0094 0105 0324 0326 0381 0417 0430 0454 0514 0521
                       0532 0663
norest     0244  *0091 0083
notoke     04DA  *0432 0424
null       0000  *0052 0754 0756 0758 0760 0762 0764 0766 0768 0770
                       0772 0794 0796 0798 0800 0802 0805 0807 0809
onebit     0607  *0596 0593
recv       05E4  *0573 0061 0101 0146 0149 0194 0200 0320 0341 0377
                       0584 0605 0659 0673 0721
removbrk   022B  *0081 0128
reset      05C6  *0549
run        0331  *0212 0112
save01     061E  *0616 0613
save02     0630  *0630 0625
save03     064C  *0650 0639
save04     0677  *0678 0672
save12     06CD  *0741 0729
save13     068D  *0699 0693
save14     06A2  *0712 0705
savebpa    06D8  *0750 0084 0413 0415 0427 0429 0516 0518
saveopc    06D7  *0749 0081 0086 0092 0406 0420 0432 0452 0457 0512
set        05C9  *0551 0548
setupc     05A4  *0526 0126
shiftbit   060A  *0598 0595 0597
space      0617  *0612 0271 0275 0282 0283 0315 0318 0480 0617
txmit      05B6  *0538 0060 0100 0102 0204 0272 0285 0293 0298 0327
                       0344 0378 0503 0566 0615 0627 0629 0642 0711 0724
txmsg      0636  *0638 0079 0089 0132 0143 0186 0193 0206 0217 0235
                       0261 0355 0364 0367 0376 0412 0435 0461 0466 0468 0485
                       0491 0495 0507 0511 0651
ustatus    051B  *0463 0124
wait02     05E9  *0576 0574 0576
```