```
0001  ;951112 1554
0002  ;*************************************************************************
0003  ;* PROGRAM NAME: UZE201  .Z80   DATE: 30.10.1995    LINK INTO: is main    *
0004  ;* LANGUAGE: Z80-Macro Assembler for ZMAC Cromemco Version 03.04 (ZASM)    *
0005  ;* PURPOSE:  Advanced Bare-Bones BIOS for Universal Z80 Engine (CPAC80)    *
0006  ;* INCLUDE FILES: none.                                                   *
0007  ;* OBJECT FILES:  none.                                                   *
0008  ;* INPUT FILES:   none.                                                   *
0009  ;* OUTPUT FILES:  none.                                                   *
0010  ;* ---------------------------------------------------------------------- *
0011  ;* VERSION:  2.01 12-NOV-95 Bugfixes from version 1.02 introduced         *
0012  ;* ---------------------------------------------------------------------- *
0013  ;* VERSION:  2.01  3-NOV-95 ROM-Version (V.2.00 is RAM-version)           *
0014  ;* ---------------------------------------------------------------------- *
0015  ;* VERSION:  2.00 30-OCT-95 Defined A-command to get commands in clear text *
0016  ;*           Routines derived from EXT-BAS/10K-BASIC from SBCS.           *
0017  ;*           This version is for ADA 950225. It sets the PIO's in mode-3  *
0018  ;*           and loops in HP-MAN mode, instead of DumyAp.                 *
0019  ;* ---------------------------------------------------------------------- *
0020  ;* VERSION:  1.02 29-DEC-94 Introduced proper handshake                   *
0021  ;* ---------------------------------------------------------------------- *
0022  ;* VERSION:  1.01 26-DEC-94 Rearranged source and added routines          *
0023  ;* ---------------------------------------------------------------------- *
0024  ;* VERSION:  1.00 22-OCT-94 First release.                                *
0025  ;* ---------------------------------------------------------------------- *
0026  ;* VERSION:  0.03 20-OCT-94 Extensive debugging and M8 added.             *
0027  ;* ---------------------------------------------------------------------- *
0028  ;* VERSION:  0.02 19-OCT-94 Interrupt works!                              *
0029  ;* ---------------------------------------------------------------------- *
0030  ;* VERSION:  0.01  6-SEP-94 Start of labour                               *
0031  ;* ---------------------------------------------------------------------- *
0032  ;* ASSEMBLE AS:  ZMAC UZE.ddd XREF SYMB HEX DATE=mmddyy TIME=hhmmss       *
0033  ;* ---------------------------------------------------------------------- *
0034  ;* COMMENTS: Use ZMAC.COM on Osborne or ZMAC.CPM and ZMAC.COM (via GENCOM  *
0035  ;*           and 22NICE (New Insystem CP/M Emulator) by Sydex) for DOS.   *
0036  ;* ====================================================================== *
0037  ;* COMMAND - A command is invoked by an interrupt on first character. It  *
0038  ;*           executes and returns to the interrupt service routine. The   *
0039  ;*           prime registers are used (AF', BC', DE', HL').               *
0040  ;*           A command cannot be accessed from within.                    *
0041  ;* PUBLIC-PRCEDURE - Can be called by any application. The procedure re-  *
0042  ;*           sumes the current task via ILoopV vector. No register is     *
0043  ;*           altered. They can be considered an internal commands.        *
0044  ;*           If executed by E-command, instruction que stack is poped.    *
0045  ;* PUBLIC-SUBROUTINE - Can be called from any application. The subroutine *
0046  ;*           returns to the calling routine. No register is altered.      *
0047  ;*           They can be called directly with M8-command.                 *
0048  ;* RESTART - Fast and efficient subroutines. Specific registers must be set *
0049  ;*           on entry and/or are set on exit. Returns to caller.          *
0050  ;* PAGE-SUB - As subroutine, but is located in page zero and address is not *
0051  ;*           filled in table, where PubVec points to.                     *
0052  ;* SUBROUTINE - Can be called from any application. Specific registers must *
0053  ;*           be set on entry and/or are set on exit. Except for the       *
0054  ;*           required registers, the state of the others (AF, BC, DE, HL)  *
0055  ;*           are undefined. Returns to caller.                            *
0056  ;* ---------------------------------------------------------------------- *
0057  ;* EXTENSION: The A-command expects at least 1 character [A;Z] as an ope- *
0058  ;*  ration name. The operation name can have up to 63 characters. The keys *
0059  ;*  are converted to upper case. Any character outside [A;Z] is assumed to *
0060  ;*  signal end of ascii-entry. The command found is executed in exactly the *
0061  ;*  same manner as when started with the "E"-command. The execution address *
0062  ;*  can be found in (param1), (param2) holds a pointer in SysBuf where the *
0063  ;*  first unaccepted character code is stored. This can be used if para-  *
0064  ;*  meters are expected by the named routine.                             *
0065  ;* ---------------------------------------------------------------------- *
0066  ;* LISTING NOTE: The micro-letter cannot be represented by the Cromemco   *
0067  ;*   assembler. Therefore, micro seconds are represented as Ms.           *
0068  ;* ====================================================================== *
0069  ;* AUTHOR: Hans-Ruedi H. Wernli, Pletschgasse, CH-3952 Susten, Switzerland. *
0070  ;* CO-AUTHOR:    Robert Glaisen, Haus Colinal, CH-3952 Susten, Switzerland. *
0071  ;*************************************************************************
0072  ;
0073  ;
```

```
                0074  ; DEFAULT SETTING ON COLD-BOOT
                0075  ;------------------------------------------------------------------------
                0076  ;
                0077  ; CPU:         Interrupt Mode-2 selected (I-reg, Device)
                0078  ; DAISY: Daisy-chain interrupt precedence: CTC-SIO-PIO
                0079  ;
                0080  ;
                0081  ; USED DEVICES FOR BIOS-OPERATION
                0082  ;------------------------------------------------------------------------
                0083  ;
                0084  ; SIO-B and CTC-channel 3 for communication with outside world
                0085  ;
                0086  ;
                0087  ; MEMORY MAP 0000 TO FFFF
                0088  ;========================================================================
                0089  ;
                0090  ; FIRMWARE - EPROM 0000 TO 7FFF
                0091  ;------------------------------------------------------------------------
                0092  ;
                0093  ; 0000: RST0 Cold-Boot address
                0094  ; 0008: RST1 INput STATus  restart
                0095  ; 0010: RST2 OuTput STATus restart
                0096  ; 0018: RST3 CTS sTATus    restart
                0097  ; 0020: RST4 CONsoleInput  restart
                0098  ; 0028: RST5 CONsoleOUTput restart
                0099  ; 0030: RST6 unassigned (can be used for Interrupt Mode-0)
                0100  ; 0038: RST7 Interrupt Mode-1 restart. Loads addr from RAM at SysWrd+2
                0101  ; 0040: CmdLst Command List for T, E, D, U, H, M, A and Else
                0102  ; 0050: IM2Lst Interrupt Mode-2 table
                0103  ; 0064: PubVec constant points to PubPro (100 decimal)
                0104  ; 0066: Non-maskable interrupt vector. Loads addr from RAM at SysWrd+4
                0105  ; 006D:  3 spare bytes
                0106  ; 0070: Dummy Application
                0107  ; 0074: 12 spare bytes
                0108  ; 0080: CoolSt Cool-start address. System setup, destructive boot.
                0109  ;       WarmSt, SioIsr, Commands, Procedures, Subroutines, SVInit-Table,
                0110  ;       Version
                0111  ; 7FFF: Last firmware address
                0112  ;
                0113  ;
                0114  ; SOFTWARE - S-RAM 8000 TO FFFF
                0115  ;------------------------------------------------------------------------
                0116  ;
                0117  ; 8000: Soft application and data storage area 32'512 bytes minus stack
                0118  ; FEFF: Stack bottom, grows backward to 8000
                0119  ; FF00: SysWrd System Words,   42 unsigned word variables
                0120  ; FF54: SysByt System Bytes,   14 unsigned byte variables
                0121  ; FF62: AllInt All interrupt,  10 interrupt service jumps
                0122  ; FF80: SysBuf System Default Buffer 64 bytes
                0123  ; FFC0: IQRS   Instruction Queue Return Stack (32 commands)
                0124  ; FFFF: RET-instruction for command A. May be overwritten with care.
                0125  ;
                0126  ;
                0127  ; CONSTANTS DEFINITIONS
                0128  ;========================================================================
                0129  ;
                0130  ; FLOOBY DUST
                0131  ;------------------------------------------------------------------------
                0132  ;
(000A)          0133  lf      equ   10              ;line feed
(000D)          0134  cr      equ   13              ;carriage return
(0020)          0135  blank   equ   32              ;space or blank character
                0136  ;
                0137  ;
```

```
                0138  ; DEVICES
                0139  ;------------------------------------------------------------------------------
                0140  ;
      (0010)    0141  ctc0     equ    16              ;CTC channel 0 address 10h
      (0011)    0142  ctc1     equ    17              ;CTC channel 1 address 11h
      (0012)    0143  ctc2     equ    18              ;CTC channel 2 address 12h
      (0013)    0144  ctc3     equ    19              ;CTC channel 3 address 13h, used
      (0018)    0145  sioad    equ    24              ;SIO-A data    address 18h
      (0019)    0146  sioac    equ    25              ;SIO-A command address 19h
      (001A)    0147  siobd    equ    26              ;SIO-B data    address 1Ah, used
      (001B)    0148  siobc    equ    27              ;SIO-B command address 1Bh, used
      (001C)    0149  pioad    equ    28              ;PIO-A data    address 1Ch
      (001D)    0150  pioac    equ    29              ;PIO-A command address 1Dh
      (001E)    0151  piobd    equ    30              ;PIO-B data    address 1Eh
      (001F)    0152  piobc    equ    31              ;PIO-B command address 1Fh
      (00F0)    0153  wdtsr    equ   240              ;WDT st-by reg address F0h
      (00F1)    0154  wdtcr    equ   241              ;WDT command   address F1h
      (00F4)    0155  dcipr    equ   244              ;Daisy-chain Int precedence reg F4h
                0156  ;
                0157  ;
                0158  ; VECTOR SYMBOL NAMES DECLARATION FOR RESTARTS
                0159  ;------------------------------------------------------------------------------
                0160  ;
      (0000)    0161  ColdSt   equ    0               ;vector for RST00
      (0008)    0162  InStat   equ    8               ;vector for RST08
      (0010)    0163  OtStat   equ   16               ;ector for RST10
      (0018)    0164  CTStat   equ   24               ;vector for RST18
      (0020)    0165  ConIn    equ   32               ;vector for RST20
      (0028)    0166  ConOut   equ   40               ;vector for RST28
                0167  ;        equ   48               ;vector for RST30 IM0
                0168  ;        equ   56               ;vector for RST38 IM1
                0169  ;
                0170  ;
                0171  ; VARIABLE-AREA DEFINITION
                0172  ;------------------------------------------------------------------------------
                0173  ;
      (FF00)    0174  SysWrd   equ   65280            ;first system word variable 42
      (FF54)    0175  SysByt   equ   65364            ;first system byte variable 14
      (FF62)    0176  AllIsr   equ   65378            ;start of jump table with 10 entries
      (FF80)    0177  SysBuf   equ   65408            ;start system buffer 128 bytes
                0178  ;
                0179  ;
                0180  ;                                                    END OF PRE-DEFINITIONS
                0181  ;==============================================================================
                0182  ; C O D E  S T A R T
                0183  ;==============================================================================
                0184  ;
                0185  ; ORIGINE DECLARATION - ROMABLE CODE
                0186  ;------------------------------------------------------------------------------
                0187  ;
      (0000)    0188          org   0                 ;start at 0
                0189  ;
                0190  ;
                0191  ; START, RESTART AND INTERRUPT VECTORS
                0192  ;==============================================================================
                0193  ;
                0194  ; COLDST - Cold Start Boot-Up                                          RST-00
                0195  ;          On Entry:
                0196  ;          On Exit: All registers and Flags used
                0197  ;------------------------------------------------------------------------------
                0198  ;
0000  F3        0199          di                      ;disable interrupt
0001  C38000    R 0200        jp    coolst            ;00h cool-boot start address
0004  FFFFFFFF   0201          db    255,255,255,255   ;free
                0202  ;
                0203  ;
                0204  ;
```

```
                0205  ; INSTAT - Input Status for receiver                                RST-08
                0206  ;          On Entry: NC do Not Carry on, i.e. return status only
                0207  ;                    CY wait until character pending
                0208  ;          On Exit:   Z no character pending in rx-buffer
                0209  ;                    NZ character ready to read from rx-buffer
                0210  ;          Bytes:     8
                0211  ;          Execution times: NC  7.50 Ms
                0212  ;                           CY  8.75 Ms + (n-1) * 10.25 Ms
                0213  ;-------------------------------------------------------------------------
                0214  ;
0008  DB1B      0215          in    a,(siobc)         ;read RR0
000A  CB47      0216          bit   0,a               ;NZ if character ready
000C  D0        0217          ret   nc                ;NC=return with flag
000D  C0        0218          ret   nz                ;NZ=character pending
000E  18F8      0219          jr    instat            ;repeat restart
                0220  ;
                0221  ;
                0222  ; OTSTAT - Output Status for transmitter                            RST-10
                0223  ;          On Entry: NC do Not Carry on, i.e. return status only
                0224  ;                    CY wait until tx-buffer empty
                0225  ;          On Exit:   Z tx-buffer void
                0226  ;                    NZ tx-character still pending
                0227  ;          Bytes:     8
                0228  ;          Execution times: NC  7.50 Ms
                0229  ;                           CY  8.75 Ms + (n-1) * 10.25 Ms
                0230  ;-------------------------------------------------------------------------
                0231  ;
0010  DB1B      0232          in    a,(siobc)         ;read RR0
0012  CB57      0233          bit   2,a               ; Z if tx-buffer void
0014  D0        0234          ret   nc                ;NC=return with flag
                0235  ;       ret   z                 ; Z=buffer void
0015  C0        0236          ret   nz                ;bugfix 951112
0016  18F8      0237          jr    otstat            ;repeat restart
                0238  ;
                0239  ;
                0240  ; CTSTAT - Cleared To Send Signal check                             RST-18
                0241  ;          On Entry: NC do Not Carry on, i.e. return status only
                0242  ;                    CY wait until CTS low
                0243  ;          On Exit:   Z CTS high
                0244  ;                    NZ CTS low
                0245  ;          Bytes:     8
                0246  ;          Execution times: NC  7.50 Ms
                0247  ;                           CY  8.75 Ms + (n-1) * 10.25 Ms
                0248  ;-------------------------------------------------------------------------
                0249  ;
0018  DB1B      0250          in    a,(siobc)         ;read RR0
001A  CB6F      0251          bit   5,a               ;check CTS line
001C  D0        0252          ret   nc                ;NC=return with flag
001D  C0        0253          ret   nz                ;NZ=CTS low
                0254  ;       jr    ctstat            ;repeat restart
001E  1854      0255          jr    rstsio            ;bugfix - reset SIO-B
                0256  ;
                0257  ;
                0258  ; CONIN  - Console Input, wait for and get character               RST-20
                0259  ;          On Entry:
                0260  ;          On Exit:  A=character read
                0261  ;          Bytes:     5
                0262  ;          Execution times: 9.50 Ms + InStat
                0263  ;-------------------------------------------------------------------------
                0264  ;
0020  37        0265          scf                     ;wait for character flag
0021  CF        0266          rst   instat            ;wait for character
0022  DB1A      0267          in    a,(siobd)         ;get character
0024  C9        0268          ret
0025  FFFFFF    0269          db    255,255,255       ;spare bytes
                0270  ;
                0271  ;
```

```
                0272  ; CONOUT - Console Output, send a char if CTS is low and buffer void   RST-28
                0273  ;          On Entry: A=character to send
                0274  ;          On Exit:  A=character sent
                0275  ;          Bytes:     8
                0276  ;          Execution times: 17.25 Ms + CTStat + OtStat
                0277  ;-------------------------------------------------------------------------
                0278  ;
0028  37        0279          scf                     ;wait for CTS low flag
0029  F5        0280          push  af                ;save character to be sent
002A  DF        0281          rst   ctstat            ;wait until CTS is low
002B  D7        0282          rst   otstat            ;wait until tx-buffer is empty
002C  F1        0283          pop   af                ;restore character to be sent
002D  D31A      0284          out   (siobd),a         ;send character
002F  C9        0285          ret
                0286  ;
                0287  ;
                0288  ; INTMD0 - Interrupt Mode 0 restart.                                  RST-30
                0289  ;          The memory location contains the address, where the appropriate
                0290  ;          Interrupt service routine can be found. Initially set to IM0ret
                0291  ;          BB-BIOS uses Interrupt-Mode 2. Change (SysWrd+76)=IMode0
                0292  ;          for application dependant service routine.
                0293  ;-------------------------------------------------------------------------
                0294  ;
0030  2A4CFF    0295          ld    hl,(SysWrd + 76)  ;execution address vector in addr
0033  E9        0296          jp    (hl)              ;IMode0 = im0ret
0034  FB        0297  im0ret: ei                      ;default Int-Mode-0 service routine
0035  ED4D      0298          reti                    ;enable interrupt and return
0037  FF        0299          db    255               ;spare byte
                0300  ;
                0301  ;
                0302  ; INTMD1 - Interrupt Mode 1 restart.                                  RST-38
                0303  ;          The memory location contains the address, where the appropriate
                0304  ;          Interrupt service routine can be found. Initially set to IM1ret
                0305  ;          BB-BIOS uses Interrupt-Mode 2. Change (SysWrd+2)=IMode1
                0306  ;          for application dependant service routine.
                0307  ;-------------------------------------------------------------------------
                0308  ;
0038  2A02FF    0309          ld    hl,(SysWrd + 2)   ;execution address vector in addr
003B  E9        0310          jp    (hl)              ;IMode1 = im1ret
003C  FB        0311  im1ret: ei                      ;default Int-Mode-1 service routine
003D  ED4D      0312          reti                    ;enable interrupt and return
003F  FF        0313          db    255               ;spare byte
                0314  ;
                0315  ;
                0316  ; CMDLST - Command list. Command interpreter loads address by           0040
                0317  ;          calculating offset from command number, then jumps to
                0318  ;          appropriate routine. Do NOT alter sequence without adjusting
                0319  ;          SIOSR SIO-Service Routine.     8 commands, 16 bytes
                0320  ;-------------------------------------------------------------------------
                0321  ;
0040  4F01      0322  CmdLst: dw    TASK              ;T-command TASK
0042  6201      0323          dw    EXEC              ;E-command EXEC
0044  8101      0324          dw    DNLD              ;D-command DNLD
0046  C501      0325          dw    UPLD              ;U-command UPLD
0048  0A02      0326          dw    HELO              ;H-command HELO
004A  2702      0327          dw    MONI              ;M-command MONI
004C  5702      0328          dw    AUXY              ;A-command AUXY
004E  4101      0329          dw    RetIsr            ;else-command
                0330  ;
```

```
                0331  ;
                0332  ; IM2TBL Table for Interrupt Mode 2 execution addresses                  0050
                0333  ;        ALLINT is intitialized for JP WARMST for all interrupts
                0334  ;        except the SIO-B vector jumps to SIOISR
                0335  ;-------------------------------------------------------------------------
                0336  ;
0050  62FF      0337  im2tbl: dw    allisr           ;SIO-A, to be programmed          80
0052  65FF      0338          dw    allisr +  3      ;SIO-B sevice routine              82
0054  68FF      0339          dw    allisr +  6      ;PIO-A, to be programmed           84
0056  6BFF      0340          dw    allisr +  9      ;PIO-B, to be programmed           86
0058  6EFF      0341          dw    allisr + 12      ;CTC-0, to be programmed           88
005A  71FF      0342          dw    allisr + 15      ;CTC-1, to be programmed           90
005C  74FF      0343          dw    allisr + 18      ;CTC-2, to be programmed           92
005E  77FF      0344          dw    allisr + 21      ;CTC-3, to be programmed           94
0060  7AFF      0345          dw    allisr + 24      ;spare                             96
0062  7DFF      0346          dw    allisr + 27      ;spare                             98
                0347  ;
                0348  ;
                0349  ; PUBVEC Public Vector.                                              0064h
                0350  ;        This location points to start of a table that holds a list of
                0351  ;        entry addresses for subroutines accessible for any application.
                0352  ;        Firmware description must identify procedures in list.
                0353  ;-------------------------------------------------------------------------
                0354  ;
0064  8B06      0355  pubvec: dw    pubpro           ;ROM-addr of procedures list      100
                0356  ;
                0357  ;
                0358  ; NMIVEC - Non-maskable interrupt vector                          NMI-66h
                0359  ;          An NMI performas just a warm-start. Stack and SIO-B are reset,
                0360  ;          but memory and system variables are left untouched. NMI ought
                0361  ;          to be used as power fail recovery or as Reset with a push button.
                0362  ;          To INACTIVATE NMI, load 61h (97d) into FF04h (65284d) and use
                0363  ;          default interrupt service routine of Int-Mode-1, or any other
                0364  ;          address that points to an appropriate service routine.
                0365  ;-------------------------------------------------------------------------
                0366  ;
0066  2A04FF    0367  nmivec: ld    hl,(SysWrd + 4)  ;NMIVec = warmst
0069  E5        0368          push  hl               ;initially jump to warmst
006A  ED45      0369          retn                   ;continue at warmst
006C  FFFFFFFF  0370          db    255,255,255,255  ;4 spare bytes
                0371  ;
                0372  ;
                0373  ; DUMYAP - Dummy Application                        APPLICATION-LOOP  0070
                0374  ;          On Entry:
                0375  ;          On Exit:  (HL)=ILoopV
                0376  ;          Bytes:     4  t-cycles: 20  execution time 5 Ms
                0377  ;                                      (200'000 loops per second)
                0378  ;-------------------------------------------------------------------------
                0379  ;
0070  2A06FF    0380  dumyap: ld    hl,(SysWrd + 6)  ;ILoopV
0073  E9        0381          jp    (HL)             ;ILoopV initially=dumyap
                0382  ;
                0383  ;
                0384  ; RSTSIO - Resets SIO-B. This is a patch from CTSTAT and a bugfix.      0074
                0385  ;-------------------------------------------------------------------------
                0386  ;
0074  3E15      0387  rstsio: ld    a,00010101b      ;select WR5
0076  D31B      0388          out   (siobc),a
0078  3E6A      0389          ld    a,01101010b      ;CTS=0, RTS=1
007A  D31B      0390          out   (siobc),a
007C  189A      0391          jr    ctstat           ;end of patch
                0392  ;
                0393  ;
                0394  ; SPARE bytes                                                        007E
                0395  ;-------------------------------------------------------------------------
007E  FFFF      0396          db    255,255
                0397  ;
                0398  ;
```

```
                0399  ;==============================================================================
                0400  ; M A I N   P R O G R A M  :  S E T U P
                0401  ;==============================================================================
                0402  ;
                0403  ; COOLST - Cool Start.                                                    0080
                0404  ;          Tests memory and initialises all system variables.
                0405  ;          If (SysWrd+0) = 32767 then memory test was successfull.
                0406  ;          Memory test executes in 770ms (3'080'225 t-cycles) [0080]
                0407  ;------------------------------------------------------------------------------
                0408  ;
0080  F3        0409  coolst: di                    ;cool boot entry, disable interrupt
0081  21FF7F    0410        ld    hl,32767        ;HL points to RAM-start - 1
0084  16FF      0411        ld    d,255           ;all bits set, data to be loaded
0086  23        0412  cools0: inc   hl              ;point to next address
0087  7C        0413        ld    a,h
0088  B5        0414        or    l
0089  2807      0415        jr    z,cools1        ;if HL=65536 then HL=0
008B  72        0416        ld    (hl),d          ;set all bits in memory
008C  7E        0417        ld    a,(hl)          ;read after write
008D  3C        0418        inc   a               ;255 + 1 = 0
008E  2009      0419        jr    nz,cools3       ;fail
0090  18F4      0420        jr    cools0          ;set all bits
0092  14        0421  cools1: inc   d               ;255 + 1 = 0, data to clear memory
0093  2B        0422  cools2: dec   hl              ;HL starts at 65535
0094  72        0423        ld    (hl),d          ;write zero
0095  7E        0424        ld    a,(hl)          ;read after write
0096  BA        0425        cp    d               ;check zero = same
0097  28FA      0426        jr    z,cools2        ;if cleared, test next until fail
0099  EB        0427  cools3: ex    de,hl           ;DE=first unmatching address
009A  2100FF    0428        ld    hl,SysWrd       ;HL points to MCFAdr
009D  73        0429        ld    (hl),e          ; usually, this would be
009E  23        0430        inc   hl              ; 32767, last ROM address fails to
009F  72        0431        ld    (hl),d          ; be reset to zero
00A0  23        0432        inc   hl              ;HL points to SysWrd+2
00A1  118D05    0433        ld    de,svinit + 2   ;start of firm system variables
00A4  EB        0434        ex    de,hl           ;HL=srce, DE=dest
00A5  01FE00    0435        ld    bc,254          ;FF02 to FFFF = 254 bytes
00A8  EDB0      0436        ldir                  ;initialize whole area
00AA  CD7203    0437        call  iqpush          ;push ILoopV onto IQRS
                0438  ;
                0439  ;
                0440  ; WARM START. System setup. Cold boot continues via cool boot here.
                0441  ;              Memory and variables are left unaltered. Warm boot could be
                0442  ;              used as system recovery from NMI.
                0443  ;              IX points to SysWrd, IY points to SysByt
                0444  ;------------------------------------------------------------------------------
                0445  ;
00AD  F3        0446  warmst: di                    ;warm boot entry, disable interrupt
00AE  AF        0447        xor   a               ;table of jump addresses on page-0
00AF  ED47      0448        ld    i,a
00B1  ED5E      0449        im    2               ;interrupt mode 2
00B3  017000    0450        ld    bc,dumyap       ;dummy application absolute address
00B6  2100FF    0451        ld    hl,SysWrd       ;stack top, just below SysWrd
00B9  F9        0452        ld    sp,hl           ;stackpointer at FF00 (FEFE & FEFF)
00BA  C5        0453        push  bc              ;stack underflow starts dummy applic.
00BB  21C0FF    0454        ld    hl,SysBuf + 64  ;reset IQRS
00BE  224AFF    0455        ld    (SysWrd + 74),hl
00C1  CD7203    0456        call  iqpush          ;initialize IQRS with DumyAp
00C4  0604      0457        ld    b,4             ;---- 4 CTC to initialize
00C6  0E10      0458        ld    c,16            ;ports 16, 17, 18, 19
00C8  1602      0459        ld    d,2             ;set to 9600 Baud
00CA  1E45      0460        ld    e,01000101b
00CC  ED59      0461  ctcset: out   (c),e           ;channel control word
00CE  ED51      0462        out   (c),d           ;initialize counter
00D0  0C        0463        inc   c               ;next CTC-port address
00D1  10F9      0464        djnz  ctcset          ;do 4 ports
00D3  21EE00    0465        ld    hl,siotbl       ;---- initialize SIO-A
00D6  0E19      0466        ld    c,sioac         ;port address for SIO-A command
00D8  0609      0467        ld    b,9             ;9 bytes for SIO-A
00DA  EDB3      0468        otir                  ;initialize SIO-A
00DC  0E1B      0469        ld    c,siobc         ;---- initialize SIO-B
00DE  060B      0470        ld    b,11            ;11 bytes for SIO-B
00E0  EDB3      0471        otir                  ;initialize and start SIO-B
```

```
00E2  DD2100FF    0472          ld    ix,SysWrd         ;IX=pointer to 1st system word
00E6  FD2154FF    0473          ld    iy,SysByt         ;IY=pointer to 1st system byte
00EA  FB          0474          ei                      ;ready, enable interrup
00EB  C37000    R 0475          jp    dumyap            ;infinite loop
                  0476  ;
                  0477  ;
                  0478  ; SIOTBL - Used to initialize SIO-A to 9600,N,8,1. Rx & Tx disabled   9 Byte
                  0479  ;          Bytes: 20          SIO-B to 9600,N,8,1. Rx & Tx enabled   11 byte
                  0480  ;-------------------------------------------------------------------------------
                  0481  ;
00EE  18          0482  siotbl: db    00011000b         ;WR0 00-011-000       ---- SIO-A ----
00EF  14          0483          db    00010100b         ;WR0 00-010-100
00F0  44          0484          db    01000100b         ;WR4 01-00-01-00
00F1  13          0485          db    00010011b         ;WR0 00-010-011
00F2  C0          0486          db    11000000b         ;WR3 11-0-0-0-0-0-0
00F3  15          0487          db    00010101b         ;WR0 00-010-101
00F4  60          0488          db    01100000b         ;WR5 x-11-0-0-0-0-0
00F5  11          0489          db    00010001b         ;WR0 00-010-001
00F6  00          0490          db    00000000b         ;WR1 000-00-0-0-0
00F7  18          0491          db    00011000b         ;WR0 00-011-000       ---- SIO-B ----
00F8  14          0492          db    00010100b         ;WR0 00-010-100
00F9  44          0493          db    01000100b         ;WR4 01-00-01-00
00FA  12          0494          db    00010010b         ;WR0 00-010-010
00FB  52          0495          db     82               ;WR2 01010010
00FC  11          0496          db    00010001b         ;WR0 00-010-001
00FD  08          0497          db    00001000b         ;WR1 000-01-0-0-0
00FE  15          0498          db    00010101b         ;WR0 00-010-101
00FF  6A          0499          db    01101010b         ;WR5 x-11-0-1-0-1-0
0100  13          0500          db    00010011b         ;WR0 00-010-011
0101  C1          0501          db    11000001b         ;WR3 11-0-0-0-0-0-1
                  0502  ;
                  0503  ;
                  0504  ; SIOBIE - SIO-B Interrupt Enable after an interrupt occured
                  0505  ;-------------------------------------------------------------------------------
                  0506  ;
0102  10          0507  siobie: db    00010000b         ;WR0 00-010-000 reset ext int
0103  20          0508          db    00100000b         ;WR0 00-100-000 enable int on 1st chr
                  0509  ;
                  0510  ;
                  0511  ; I N T E R R U P T   S E R V I C E   R O U T I N E
                  0512  ;===============================================================================
                  0513  ;
                  0514  ; SIO-B Interrupt service routine.
                  0515  ;       Save environment, i.e. registers AF, BC, DE and HL. IX, IY not used
                  0516  ;       Read character from SIO-B and convert it to upper case, save it
                  0517  ;       Call routine according to command received
                  0518  ;       Clear SIO-B errors and enable SIO-B interrupt
                  0519  ;       Restore environmnt and re-enter loop, either at DumyAp, Task or Exec
                  0520  ;-------------------------------------------------------------------------------
                  0521  ;
0104  D9          0522  sioisr: exx                     ;swap BC, DE, HL with prime regs
0105  08          0523          ex    af,af'            ;swap AF with prime register
0106  E7          0524          rst   ConIn             ;(A) = char
0107  CBAF        0525          res   5,a               ;assume letter, convert to uppercase
0109  110000      0526          ld    de,0              ;command list offset
010C  FE54        0527          cp    'T'               ;TASK command
010E  281F        0528          jr    z,cmdvec          ;T - DE=0
0110  1C          0529          inc   e                 ;
0111  FE45        0530          cp    'E'               ;EXEC command
0113  281A        0531          jr    z,cmdvec          ;E - DE=1
0115  1C          0532          inc   e
0116  FE44        0533          cp    'D'               ;DNLD command (receive)
0118  2815        0534          jr    z,cmdvec          ;D - DE=2
011A  1C          0535          inc   e
011B  FE55        0536          cp    'U'               ;UPLD command (send)
011D  2810        0537          jr    z,cmdvec          ;U - DE=3
011F  1C          0538          inc   e
0120  FE48        0539          cp    'H'               ;HELO command
0122  280B        0540          jr    z,cmdvec          ;H - DE=4
0124  1C          0541          inc   e
0125  FE4D        0542          cp    'M'               ;MONI command
0127  2806        0543          jr    z,cmdvec          ;Z - DE=5
0129  1C          0544          inc   e
```

```
012A  FE41        0545          cp    'A'               ;AUXY command
012C  2801        0546          jr    z,cmdvec          ;A - DE=6
012E  1C          0547          inc   e                 ;ELSE DE=7
012F  CD0A03      0548  cmdvec: call  swpcmd            ;swap commands, update current & last
0132  7B          0549          ld    a,e               ;get command number
0133  3259FF      0550          ld    (SysByt + 5),a    ;(SysByt+3)=CmdNbr
0136  83          0551          add   a,e               ;double command number
0137  5F          0552          ld    e,a               ;DE=offset
0138  214000      0553          ld    hl,CmdLst         ;point to start of list
013B  19          0554          add   hl,de             ;point to command jump op-code
013C  5E          0555          ld    e,(hl)
013D  23          0556          inc   hl
013E  56          0557          ld    d,(hl)            ;DE=vector
013F  EB          0558          ex    de,hl             ;HL=vector
0140  E9          0559          jp    (hl)              ;continue at appropriate command
                  0560  ;
                  0561  ;
                  0562  ; RETISR - Return from interrupt service routine. Every command returns
                  0563  ;          here. This concludes the service routine.
                  0564  ;------------------------------------------------------------------------
                  0565  ;
0141  210201      0566  retisr: ld    hl,siobie         ;Enable interrupt on 1st char rcvd
0144  0E1B        0567          ld    c,siobc           ;port address of SIO-B
0146  0602        0568          ld    b,2               ;2 bytes
0148  EDB3        0569          otir                    ;make SIO-B ready
014A  D9          0570          exx                     ;restore BC, DE, HL registers
014B  08          0571          ex    af,af'            ;restore AF registers
014C  FB          0572          ei                      ;enable interrupt
014D  ED4D        0573          reti                    ;return from interrupt
                  0574  ;
                  0575  ;
                  0576  ; C O M M A N D   E X E C U T I O N
                  0577  ;========================================================================
                  0578  ;
                  0579  ; TASK   Replaces stack top with return address by the                COMMAND
                  0580  ;        new tasks execution address. When the interrupt service
                  0581  ;        routine terminates, it returns to the new task instead of the old
                  0582  ;        one, that had been interrupted, as is the case with EXECUTE.
                  0583  ;        On Entry:
                  0584  ;        On Exit:  CTaddr=<addr>
                  0585  ;                  CTbyte=unknown
                  0586  ;                  CTpntr=<addr>
                  0587  ;                  CTcntr=0
                  0588  ;                  Param1=<addr>
                  0589  ;                  Param2=unknown
                  0590  ;                  Param3=Param1=<addr>
                  0591  ;                  Param4=0
                  0592  ;                  (HL)=Task addr
                  0593  ; Syntax: T<addr>,
                  0594  ;------------------------------------------------------------------------
                  0595  ;
014F  3E01        0596  TASK:   ld    a,1               ;flag Param1
0151  CD0303      0597          call  getpar            ;(Param1)=Task addr
0154  CD5E05      0598          call  setpar            ;set Param1 thru Param4
0157  CD6D05      0599          call  setvar            ;copy Param1-4 to CT vars-block
015A  E1          0600          pop   hl                ;drop old return-to-loop address
015B  2A08FF      0601          ld    hl,(SysWrd + 8)   ;CTaddr, task address
015E  E5          0602          push  hl                ;new return address for new task
015F  C34101    R 0603          jp    retisr            ;RETI starts task
                  0604  ;
                  0605  ;
```

```
                    0606  ; EXEC   loads execution address into ILoopV.                        COMMAND
                    0607  ;        When the interrupt service routine returns to looping application
                    0608  ;        (dummy), task is executed almost immediately. However, if an
                    0609  ;        application is running already, the new task can be executed only
                    0610  ;        after the current task concludes. The old task address is pushed
                    0611  ;        on the Instruction Queue Return Stack. The new requested task should
                    0612  ;        pop the old task just before it terminates (iqpop).
                    0613  ;        If there is not enough space on IQRS, the earliest entry is dumped
                    0614  ;        to make room to push the most recent one.
                    0615  ;        On Entry:
                    0616  ;        On Exit:  CEaddr=<addr>
                    0617  ;                  CEbyte=unknown
                    0618  ;                  CEpntr=<addr>
                    0619  ;                  CEcntr=0
                    0620  ;                  Param1=<addr>
                    0621  ;                  Param2=unknown
                    0622  ;                  Param3=Param1=<addr>
                    0623  ;                  Param4=0
                    0624  ;                  ILoopV=Param1=<addr>
                    0625  ;                  (HL)=<addr>
                    0626  ; Syntax: E<addr>,
                    0627  ;-------------------------------------------------------------------------------
                    0628  ;
0162  3E01        0629  EXEC:   ld    a,1               ;flag Param1
0164  CD0303      0630          call  getpar            ;(Param1)=Execution addr
0167  CD5E05      0631          call  setpar            ;set Param1 thru Param4
016A  CD6D05      0632          call  setvar            ;copy Param1-4 to CE vars-block
016D  CDB003      0633          call  iqtest            ;check enough stack
0170  2003        0634          jr    nz,exec1          ;NZ=enough
0172  CDBF03      0635          call  iqdump            ;dump earliest entry
0175  CD7203      0636  exec1:  call  iqpush            ;save old task
0178  2A10FF      0637          ld    hl,(SysWrd + 16)  ;CEaddr
017B  2206FF      0638          ld    (SysWrd + 6),hl   ;ILoopV, ILoopV=CEaddr
017E  C34101    R 0639          jp    retisr
                    0640  ;
                    0641  ;
                    0642  ; DNLD   Accepts downloaded bytes.                                   COMMAND
                    0643  ;        Stores <byte> bytes in buffer at address <addr>.
                    0644  ;        On Entry:
                    0645  ;        On Exit:  CDaddr=<addr>
                    0646  ;                  CDbyte=<byte>
                    0647  ;                  CDpntr=<addr>+<byte> first free byte in buff
                    0648  ;                  CDcntr=0 if download successfull, else remainig bytes
                    0649  ;                  HexChr=last byte received in hex
                    0650  ;                  Param1=<addr>
                    0651  ;                  Param2=<byte>
                    0652  ;                  Param3=<addr>
                    0653  ;                  Param4=0
                    0654  ;                  Regsters used: AF, BC, DE, HL
                    0655  ; Syntax: D<addr>,<byte>,
                    0656  ;-------------------------------------------------------------------------------
                    0657  ;
0181  3E01        0658  DNLD:   ld    a,1               ;flag Param1
0183  CD0303      0659          call  getpar            ;Param1
0186  3C          0660          inc   a                 ;falg Param2
0187  CD0303      0661          call  getpar            ;Param2
018A  CD5E05      0662          call  setpar            ;set Param1 thru Param4
018D  CD6D05      0663          call  setvar            ;copy Param1-4 to CD vars-block
0190  ED4B1AFF    0664          ld    bc,(SysWrd + 26)  ;CDbyte
0194  ED5B1CFF    0665          ld    de,(SysWrd + 28)  ;CDpntr
0198  3A58FF      0666          ld    a,(SysByt + 4)    ;mode
019B  CB4F        0667          bit   1,a               ;Set: ascii, reset: binary
019D  2813        0668          jr    z,dnld2           ;binary, jump
019F  E7          0669  dnld1:  rst   conin             ;fetch hex character ----------HEX---
01A0  6F          0670          ld    l,a               ;store high nibble character
01A1  E7          0671          rst   conin             ;fetch hex character
01A2  67          0672          ld    h,a               ;store low nibble character
01A3  2248FF      0673          ld    (SysWrd + 72),hl  ;HexChr
01A6  CDFC03      0674          call  ascbin            ;convert
01A9  12          0675          ld    (de),a            ;store
01AA  13          0676          inc   de                ;buffer + 1
01AB  0B          0677          dec   bc
01AC  78          0678          ld    a,b
01AD  B1          0679          or    c                 ;check all bytes converted and sent
01AE  20EF        0680          jr    nz,dnld1          ;loop
01B0  1808        0681          jr    dnld3             ;hex done
```

```
01B2  E7          0682  dnld2:  rst   conin            ;A=character  ----------BINARY-------
01B3  12          0683          ld    (de),a          ;store character received
01B4  13          0684          inc   de              ;point to next destination
01B5  0B          0685          dec   bc
01B6  78          0686          ld    a,b
01B7  B1          0687          or    c               ;check all expected received
01B8  20F8        0688          jr    nz,dnld2        ;loop
01BA  ED531CFF    0689  dnld3:  ld    (SysWrd + 28),de  ;CDpntr, next free byte
01BE  ED431EFF    0690          ld    (SysWrd + 30),bc  ;CDcntr, should be 0
01C2  C34101      0691          jp    retisr
                  0692  ;
                  0693  ;
                  0694  ; UPLD   Sends requested bytes.                                    COMMAND
                  0695  ;        Sends <byte> bytes from buffer at address <addr>.
                  0696  ;        If Bit-0 of MonMod set, uploads <byte>*2 characters
                  0697  ;        On Entry: (SysByt+ 4)=mode
                  0698  ;        On Exit:  (SysByt+ 3)=2 (Param0 flags Param2)
                  0699  ;                  CUaddr=<addr>
                  0700  ;                  CUbyte=<byte>
                  0701  ;                  CUpntr=<addr>+<byte>, first unsent byte
                  0702  ;                  CUcntr=0 if upload successfull, else remaining bytes
                  0703  ;                  Param1=<addr>
                  0704  ;                  Param2=<byte>
                  0705  ;                  Param3=<addr>+<byte>, first unsent byte
                  0706  ;                  Param4=0
                  0707  ; Syntax: U<addr>,<byte>,
                  0708  ;-------------------------------------------------------------------------
                  0709  ;
01C5  3E01        0710  UPLD:   ld    a,1             ;flag Param1
01C7  CD0303      0711          call  getpar          ;Param1
01CA  3C          0712          inc   a               ;falg Param2
01CB  CD0303      0713          call  getpar          ;Param2
01CE  CD5E05      0714          call  setpar          ;set Param1 thru Param4
01D1  CD6D05      0715          call  setvar          ;copy Param1-4 to CU vars-block
01D4  ED4B22FF    0716          ld    bc,(SysWrd + 34)  ;CUbyte
01D8  ED5B24FF    0717          ld    de,(SysWrd + 36)  ;CUpntr
01DC  3A58FF      0718          ld    a,(SysByt + 4)    ;mode
01DF  CB47        0719          bit   0,a             ;Set: ascii, reset: binary
01E1  2813        0720          jr    z,upld2         ;binary, jump
01E3  1A          0721  upld1:  ld    a,(de)          ;get byte            --------HEX-----
01E4  13          0722          inc   de
01E5  CDD603      0723          call  binasc          ;2 bytes into HexChr
01E8  2A48FF      0724          ld    hl,(SysWrd + 72)  ;HexChr
01EB  7D          0725          ld    a,l             ;get high ascii-nibble
01EC  EF          0726          rst   conout          ;send high nibble
01ED  7C          0727          ld    a,h             ;get low ascii-nibble
01EE  EF          0728          rst   conout          ;send low nibble
01EF  0B          0729          dec   bc
01F0  78          0730          ld    a,b
01F1  B1          0731          or    c               ;check all bytes converted and sent
01F2  20EF        0732          jr    nz,upld1        ;loop
01F4  1809        0733          jr    upld3           ;hex done
01F6  1A          0734  upld2:  ld    a,(de)          ;get character --------BINARY--------
01F7  13          0735          inc   de
01F8  EF          0736          rst   conout          ;send character
01F9  23          0737          inc   hl              ;point to next source
01FA  0B          0738          dec   bc
01FB  78          0739          ld    a,b
01FC  B1          0740          or    c               ;check all bytes sent
01FD  20F7        0741          jr    nz,upld2        ;loop
01FF  ED5324FF    0742  upld3:  ld    (SysWrd + 36),de  ;CUpntr, 1st unsent byte
0203  ED4326FF    0743          ld    (SysWrd + 38),bc  ;CUcntr, should be 0
0207  C34101      0744          jp    retisr
                  0745  ;
                  0746  ;
```

```
                    0747  ; HELO   Sends a predefined number of bytes                          COMMAND
                    0748  ;        Sends <byte> bytes from buffer at address <addr>.
                    0749  ;        On Entry: CHaddr=pointer to first byte to send. Default Prompt
                    0750  ;                  CHbyte=number of bytes to send        Default 1
                    0751  ;        On Exit:  CHaddr=unchanged
                    0752  ;                  CHbyte=unchanged
                    0753  ;                  CHpntr=CHaddr+CHbyte
                    0754  ;                  CHcntr=0 if upload succsessfull, else remaining bytes
                    0755  ; Syntax: H
                    0756  ;
                    0757  ;        To change CHaddr and CHbyte, send the command sequence:
                    0758  ;                  M<4-7>,<chaddr>,<chbyte>,<n>,<n>,E<setvar>,H
                    0759  ;         For EPROM-Version, send the command sequence:
                    0760  ;                  E<whowho>,H
                    0761  ;-------------------------------------------------------------------------
                    0762  ;
020A  ED4B2AFF      0763  HELO:   ld    bc,(SysWrd + 42)  ;CHbyte
020E  78            0764          ld    a,b
020F  B1            0765          or    c                 ;check for zero
0210  280B          0766          jr    z,helo1           ;no bytes to send
0212  2A28FF        0767          ld    hl,(SysWrd + 40)  ;CHaddr
0215  7E            0768  helo0:  ld    a,(hl)            ;get character
0216  EF            0769          rst   conout            ;send character
0217  23            0770          inc   hl                ;point to next source
0218  0B            0771          dec   bc
0219  78            0772          ld    a,b
021A  B1            0773          or    c                 ;check all bytes sent
021B  20F8          0774          jr    nz,helo0          ;loop
021D  222CFF        0775  helo1:  ld    (SysWrd + 44),hl  ;CHpntr, update it
0220  ED432EFF      0776          ld    (SysWrd + 46),bc  ;CHcntr, should be 0
0224  C34101        0777          jp    retisr
                    0778  ;
                    0779  ;
                    0780  ; MONI   Sets monitor flag                                          COMMAND
                    0781  ;        Defined Modes:   0 binary mode (default)           00000000
                    0782  ;                         1 hex-ascii upload (send)         00000001
                    0783  ;                         2 hex ascii download (receive)    00000010
                    0784  ;                         3 hex-ascii up- & download        00000011
                    0785  ;                       4- 7 as 0-3 but 4 parameters follow 000001xx
                    0786  ;                       8-15 calls a subroutine at p1       00001xxx
                    0787  ;        On Entry:
                    0788  ;        On Exit:  MonMod=<mode>
                    0789  ;                  case Bit-2 set only:
                    0790  ;                        CMaddr=<p1>
                    0791  ;                        CMbyte=<p2>
                    0792  ;                        CMpntr=<p3>
                    0793  ;                        CMcntr=<p4>
                    0794  ;                        Param1=<p1>
                    0795  ;                        Param2=<p2>
                    0796  ;                        Param3=<p3>
                    0797  ;                        Param4=<p4>
                    0798  ;                  case Bit-3 set only:
                    0799  ;                        Param1=<p1>
                    0800  ; Syntax: M<mode=[0, 3]>,
                    0801  ; Syntax: M<mode=[4, 7]>,<p1>,<p2>,<p3>,<p4>,  (decimal in any case)
                    0802  ; Syntax: M<mode=[8,15]>,<p1>,
                    0803  ;-------------------------------------------------------------------------
                    0804  ;
0227  3E01          0805  MONI:   ld    a,1               ;flag Param1
0229  CD0303        0806          call  getpar            ;Param1
022C  3A40FF        0807          ld    a,(SysWrd + 64)   ;Param1=mode
022F  3258FF        0808          ld    (SysByt + 4),a    ;MonMod
0232  CB5F          0809          bit   3,a               ;check for sub call
0234  2014          0810          jr    nz,moni3          ;if set, subroutine call
0236  CB57          0811          bit   2,a               ;check for more parameters
0238  280D          0812          jr    z,moni2           ;if reset, none follow
023A  3E01          0813          ld    a,1               ;parameter flag
023C  0604          0814          ld    b,4               ;4 parameters
023E  CD0303        0815  moni1:  call  getpar            ;get a parameter
0241  3C            0816          inc   a                 ;next parameter
0242  10FA          0817          djnz  moni1             ;get 4 parameters
0244  CD6D05        0818          call  setvar            ;copy Param1-4 to CM vars-block
0247  C34101        0819  moni2:  jp    retisr            ;return from patch
```

```
024A  3E01        0820  moni3:  ld    a,1              ;flag Param1
024C  CD0303      0821          call  getpar           ;get subroutine address
024F  214101      0822          ld    hl,retisr        ;return address for subroutine
0252  E5          0823          push  hl               ;subroutine pops this one
0253  2A40FF      0824          ld    hl,(SysWrd + 64) ;get address
0256  E9          0825          jp    (hl)             ;exec sub, return to retisr
                  0826  ;
                  0827  ;
                  0828  ; AUXY   Auxilliary Task                                                COMMAND
                  0829  ;        Execution is address (SysWrd+56)=CAaddr, initially set to 65535,
                  0830  ;        last RAM-address. Here, a RET-instruction can be found.
                  0831  ;        This command must end with a RET-instruction!
                  0832  ;        On Entry:
                  0833  ;        On Exit:
                  0834  ; Syntax: A  (parameters or modes not defined)
                  0835  ;-------------------------------------------------------------------------
                  0836  ;
0257  214101      0837  AUXY:   ld    hl,retisr        ;return address
025A  E5          0838          push  hl               ;Auxilliary task ends with a RET
025B  2A38FF      0839          ld    hl,(SysWrd + 56) ;CAaddr
025E  E5          0840          push  hl               ;Aux-Task address onto stack
025F  C9          0841          ret                    ;Aux Task RET will return to retisr
                  0842  ;
                  0843  ;
                  0844  ; P R O C E D U R E S
                  0845  ;=========================================================================
                  0846  ;
                  0847  ; WHOWHO - Who's who. Prepares H command to                PUBLIC-PROCEDURE
                  0848  ;          Sends newest version string.
                  0849  ;          Version string is to be accessed indirect and through a
                  0850  ;          linked list. No register altered.
                  0851  ; Usage: E<whowho>,H
                  0852  ;-------------------------------------------------------------------------
                  0853  ;
0260  E5          0854  whowho: push  hl               ;save registers used
0261  D5          0855          push  de
0262  C5          0856          push  bc
0263  F5          0857          push  af
0264  2A6400      0858          ld    hl,(pubvec)      ;HL=PubPro
0267  23          0859          inc   hl
0268  23          0860          inc   hl               ;skip addr of UseEnd
0269  5E          0861          ld    e,(hl)
026A  23          0862          inc   hl
026B  56          0863          ld    d,(hl)           ;DE=UzeVer
026C  EB          0864          ex    de,hl            ;HL=UzeVer
026D  44          0865  whowh0: ld    b,h
026E  4D          0866          ld    c,l              ;BC=UzeVer+2
026F  5E          0867          ld    e,(hl)
0270  23          0868          inc   hl
0271  56          0869          ld    d,(hl)           ;DE=(UzeVer)
0272  21FFFF      0870          ld    hl,65535         ;value of unprogrammed word
0275  AF          0871          xor   a                ;clear carry for substraction
0276  ED52        0872          sbc   hl,de            ;must be zero if unprogrammed
0278  EB          0873          ex    de,hl            ;HL=(UzeVer)
0279  20F2        0874          jr    nz,whowh0        ;(UzeVer) programmed, search next
027B  60          0875          ld    h,b
027C  69          0876          ld    l,c              ;HL=UzeVer
027D  23          0877          inc   hl               ;skip unprogrammed word
027E  23          0878          inc   hl               ;point to first character
027F  2228FF      0879          ld    (SysWrd+40),hl   ;CHaddr
0282  54          0880          ld    d,h
0283  5D          0881          ld    e,l              ;DE=buffer start
0284  01FF00      0882          ld    bc,255           ;BC=counter, not more than 255
0287  79          0883          ld    a,c              ;A=255, end marker
0288  EDB1        0884          cpir                   ;search for 255
028A  AF          0885          xor   a                ;clear carry
028B  ED52        0886          sbc   hl,de            ;HL=bytes
028D  2B          0887          dec   hl               ;one less
028E  222AFF      0888          ld    (SysWrd+42),hl   ;CHbyte
0291  3A59FF      0889          ld    a,(SysByt + 5)   ;CmdNbr
0294  3D          0890          dec   a                ;if EXEC, A=1, now 0
0295  CC8A03      0891          call  z,iqpop          ;restore old task if EXECuted
0298  2A06FF      0892          ld    hl,(SysWrd + 6)  ;ILoopV old task
```

```
029B  F1          0893          pop   af              ;restore registers used
029C  C1          0894          pop   bc
029D  D1          0895          pop   de
029E  E3          0896          ex    (sp),hl         ;restore HL, push (ILoopV)
029F  C9          0897          ret
                  0898  ;
                  0899  ;
                  0900  ; DUMPIT - Dump many bytes in lines of 16                  PUBLIC-PROCEDURE
                  0901  ;          Prints a cr/lf-sequence at start
                  0902  ;          On Entry: CMaddr=<addr>
                  0903  ;                    CMbyte=<byte>
                  0904  ;          On Exit:  CMaddr=<addr>
                  0905  ;                    CMByte=<byte>
                  0906  ;                    CMpntr=<addr>+<byte>
                  0907  ;                    CMcntr=<n>
                  0908  ;                    Param3=<addr>+<byte>
                  0909  ;                    SysBuf has last 18 characters
                  0910  ;
                  0911  ; aaaa : bb bb bb bb bb bb bb bb bb  bb bb bb bb bb bb bb | cccccccc<cr><lf>
                  0912  ;
                  0913  ; Usage:  M<4-7>,<addr>,<byte>,<n>,<n>,E<dumpit>,
                  0914  ;-------------------------------------------------------------------------------
                  0915  ;
02A0  E5          0916  dumpit: push  hl              ;save registers
02A1  C5          0917          push  bc
02A2  F5          0918          push  af
02A3  CDC004      0919          call  crlfs           ;issue cr/lf-sequence
02A6  2A32FF      0920          ld    hl,(SysWrd + 50)  ;CMbyte
02A9  44          0921          ld    b,h
02AA  4D          0922          ld    c,l
02AB  CDE504      0923          call  div16           ;BC=BC/16
02AE  78          0924          ld    a,b
02AF  B1          0925          or    c               ;test 0
02B0  2001        0926          jr    nz,dumpi0
02B2  03          0927          inc   bc              ;at least one line
02B3  2A30FF      0928  dumpi0: ld    hl,(SysWrd + 48)  ;CMaddr
02B6  2244FF      0929          ld    (SysWrd + 68),hl  ;Param3 used by LinDmp as C?pntr
02B9  CD3504      0930  dumpi1: call  lindmp          ;dump 16 bytes to SysBuf
02BC  50          0931          ld    d,b             ;save high byte of line counter
02BD  063A        0932          ld    b,58            ;58 bytes to send
02BF  2180FF      0933          ld    hl,SysBuf
02C2  7E          0934  dumpi2: ld    a,(hl)          ;fetch character
02C3  EF          0935          rst   conout          ;send character
02C4  23          0936          inc   hl              ;advance buffer pointer
02C5  10FB        0937          djnz  dumpi2          ;do one line
02C7  CD9204      0938          call  dmpasc          ;prepare ascii characters
02CA  2180FF      0939          ld    hl,SysBuf
02CD  0612        0940          ld    b,18            ;18 bytes in buffer
02CF  7E          0941  dumpi3: ld    a,(hl)          ;fetch character
02D0  EF          0942          rst   conout          ;send character
02D1  23          0943          inc   hl              ;advance buffer pointer
02D2  10FB        0944          djnz  dumpi3          ;do one line
02D4  42          0945          ld    b,d             ;restore high byte of line counter
02D5  0B          0946          dec   bc              ;one line sent
02D6  78          0947          ld    a,b
02D7  B1          0948          or    c
02D8  20DF        0949          jr    nz,dumpi1       ;repeat for all
02DA  3A59FF      0950          ld    a,(SysByt + 5)  ;CmdNbr
02DD  3D          0951          dec   a               ;if EXEC, A=1, now 0
02DE  CC8A03      0952          call  z,iqpop         ;restore old task
02E1  2A44FF      0953          ld    hl,(SysWrd + 68)  ;Param3
02E4  2234FF      0954          ld    (SysWrd + 52),hl  ;update CMpntr
02E7  2A06FF      0955          ld    hl,(SysWrd + 6)   ;ILoopV old task
02EA  F1          0956          pop   af              ;restore registers
02EB  C1          0957          pop   bc
02EC  E3          0958          ex    (sp),hl         ;restore HL, push (ILoopV)
02ED  C9          0959          ret
                  0960  ;
                  0961  ;
```

```
                    0962  ; CRLFP  - Sends carriage return-linefeed sequence          PUBLIC-PROCEDURE
                    0963  ;          On Entry:
                    0964  ;          On Exit:
                    0965  ; Usage: E<crlfp>,
                    0966  ;-------------------------------------------------------------------------------
                    0967  ;
02EE  E5            0968  crlfp:  push  hl               ;save registers
02EF  F5            0969          push  af
02F0  3E0D          0970          ld    a,cr              ;carriage return
02F2  EF            0971          rst   conout            ;send CR
02F3  3E0A          0972          ld    a,lf              ;line feed
02F5  EF            0973          rst   conout            ;send LF
02F6  3A59FF        0974          ld    a,(SysByt + 5)    ;CmdNbr
02F9  3D            0975          dec   a                 ;if EXEC, A=1, now 0
02FA  CC8A03        0976          call  z,iqpop           ;restore old task
02FD  2A06FF        0977          ld    hl,(SysWrd + 6)   ;ILoopV old task
0300  F1            0978          pop   af                ;restore register
0301  E3            0979          ex    (sp),hl           ;restore HL, push ILoopV
0302  C9            0980          ret
                    0981  ;
                    0982  ;
                    0983  ; S U B R O U T I N E S
                    0984  ;===============================================================================
                    0985  ;
                    0986  ; GETPAR - Get one parameter                                        SUBROUTINE
                    0987  ;          This subroutine is a preamble for GetAsc which it calls
                    0988  ;          On Entry: (A)=Parameter Number, 1 or 2
                    0989  ;          On Exit:  case A=1: Param0=1
                    0990  ;                              Param1=parameter received
                    0991  ;                    case A=2: Param0=2
                    0992  ;                              Param2=parameter received
                    0993  ;                    case A=N: Param0=N
                    0994  ;                              ParamN=parameter received
                    0995  ;-------------------------------------------------------------------------------
                    0996  ;
0303  3257FF        0997  getpar: ld    (SysByt + 3),a    ;Param0
0306  CD4303        0998          call  getasc            ;ParamN=value of parameter received
0309  C9            0999          ret
                    1000  ;
                    1001  ;
                    1002  ; SWPCMD Swap Command, update current and last.                  SUBROUTINE
                    1003  ;        On Entry: (A)=command character
                    1004  ;        On Exit:  (A)=command character
                    1005  ;                  OldCmd=CurCmd
                    1006  ;                  CurCmd = (A)
                    1007  ;-------------------------------------------------------------------------------
                    1008  ;
030A  F5            1009  swpcmd: push  af               ;save command character
030B  3A55FF        1010          ld    a,(SysByt+1)      ;get CurCmd
030E  3256FF        1011          ld    (SysByt+2),a      ;OldCmd=CurCmd
0311  F1            1012          pop   af                ;restore command character
0312  3255FF        1013          ld    (SysByt+1),a      ;CurCmd=A
0315  C9            1014          ret
                    1015  ;
                    1016  ;
                    1017  ; DIVHBA Divide HL by BC into A                                SUBROUTINE
                    1018  ;        Used to convert a word-integer into ascii-decimal
                    1019  ;        On Entry:  HL = dividend
                    1020  ;                   BC = divisor
                    1021  ;                  (DE)= buffer to store result
                    1022  ;        On Exit:   HL = HL - A * BC
                    1023  ;                   DE = DE+1
                    1024  ;-------------------------------------------------------------------------------
                    1025  ;
0316  AF            1026  DivHBA: xor   a                ;clear counter
0317  A7            1027  divhb0: and   a                ;clear carry flag
0318  3C            1028          inc   a                 ;presume one substract
0319  ED42          1029          sbc   hl,bc             ;substract once
031B  30FA          1030          jr    nc,divhb0         ;if no carry, one more time
031D  3D            1031          dec   a                 ;compensate one too much
031E  09            1032          add   hl,bc             ;compensate
031F  C630          1033          add   a,48              ;convert to ascii
0321  12            1034          ld    (de),a            ;store ascii-decimal figure
0322  13            1035          inc   de                ;point to next location
0323  C9            1036          ret                     ;done
                    1037  ;
```

```
                1038  ;
                1039  ; PRPUNT Gets ascii-decimal figure and adds value to HL           SUBROUTINE
                1040  ;       On Entry: (DE) = pointer to figure
                1041  ;                 HL  = value to add to
                1042  ;       On Exit:   A  = figure in binary
                1043  ;                 BC  = binary value
                1044  ;                 DE  = DE+1
                1045  ;                 HL  = result
                1046  ;-------------------------------------------------------------------------
                1047  ;
0324  1A          1048  PrpUnt: ld    a,(de)          ;get ascii figure
0325  13          1049          inc   de              ;point to next
0326  D630        1050          sub   48              ;convert to binary
0328  4F          1051          ld    c,a
0329  0600        1052          ld    b,0             ;BC=binary value
032B  09          1053          add   hl,bc           ;HL=result
032C  C9          1054          ret
                1055  ;
                1056  ;
                1057  ; P U B L I C   S U B R O U T I N E S
                1058  ;=========================================================================
                1059  ;
                1060  ; RTSTOP - Disable Request To Send                        PUBLIC-SUBROUTINE
                1061  ;          Used to stop host transmitter to digest data
                1062  ;          On Entry:
                1063  ;          On Exit:
                1064  ;          Reaction time inclusive call: 18.5 Ms
                1065  ;-------------------------------------------------------------------------
                1066  ;
032D  F5          1067  RTStop: push  af              ;save register
032E  3E05        1068          ld    a,00000101b     ;WR0 00-000-101 select WR5
0330  D31B        1069          out   (siobc),a       ;address WR5
0332  3E68        1070          ld    a,01101000b     ;WR5 0-11-0-1-0-0-0 RTS off (+RTS)
0334  D31B        1071          out   (siobc),a       ;set RTS high, disable
0336  F1          1072          pop   af              ;restore register
0337  C9          1073          ret
                1074  ;
                1075  ;
                1076  ; RTSGO - Enable Request To Send                          PUBLIC-SUBROUTINE
                1077  ;         Used to relinquish control to host transmitter
                1078  ;         On Entry:
                1079  ;         On Exit:
                1080  ;-------------------------------------------------------------------------
                1081  ;
0338  F5          1082  RTSGo:  push  af              ;save register
0339  3E15        1083          ld    a,00010101b     ;WR0 00-010-101 select WR5
033B  D31B        1084          out   (siobc),a       ;address WR5
033D  3E6A        1085          ld    a,01101010b     ;WR5 0-11-0-1-0-1-0 RTS on (-RTS)
033F  D31B        1086          out   (siobc),a       ;set RTS low, enable
0341  F1          1087          pop   af              ;restore register
0342  C9          1088          ret
                1089  ;
                1090  ;
                1091  ; GETASC - Get ascii value.                               PUBLIC-SUBROUTINE
                1092  ;          On Entry: (SysByt+3) (65379 FF63) = parameter 1 or 2
                1093  ;          On Exit:  case Param0=1: Param1=received value
                1094  ;                    case Param0=2: Param2=received value
                1095  ;                    case Param0=3: Param3=received value
                1096  ;                    case Param0=4: Param4=received value
                1097  ;                    No register altered.
                1098  ;          NOTE: No check for overrun done if result exeeds 65535, neither
                1099  ;                is number of numbers limitted. The result is formed by the
                1100  ;                last 5 (or less) characters received.
                1101  ;-------------------------------------------------------------------------
                1102  ;
0343  E5          1103  getasc: push  hl              ;save all registers to be used
0344  D5          1104          push  de
0345  C5          1105          push  bc
0346  F5          1106          push  af
0347  210000      1107          ld    hl,0            ;clear accumulator-1
034A  54          1108          ld    d,h             ;clear accumulator-2
034B  5D          1109          ld    e,l
```

```
034C  E7          1110  getas0: rst   ConIn            ;A=character
034D  CDC904      1111          call  chknbr           ;NC=valid character
0350  380B        1112          jr    c,getas1         ;not valid
0352  D630        1113          sub   48               ;convert number to binary value
0354  5F          1114          ld    e,a
0355  1600        1115          ld    d,0              ;DE=accu-2
0357  CDDE04      1116          call  MulTen           ;HL * 10, BC * 2
035A  19          1117          add   hl,de            ;HL=accu-1 (old * 10 + new)
035B  18EF        1118          jr    GetAs0           ;repeat until illegal character
035D  EB          1119  getas1: ex    de,hl            ;DE=Parameter
035E  0600        1120          ld    b,0
0360  3A57FF      1121          ld    a,(SysByt + 3)   ;get parameter number (1 to 4)
0363  3D          1122          dec   a                ;make 1=0, and 2=1
0364  87          1123          add   a                ;double
0365  4F          1124          ld    c,a              ;BC=offset into SysWrd
0366  2140FF      1125          ld    hl,SysWrd + 64   ;HL points to Param1
0369  09          1126          add   hl,bc            ;HL points to Param1 or Param2
036A  73          1127          ld    (hl),e
036B  23          1128          inc   hl
036C  72          1129          ld    (hl),d           ;store result in Param1 or Param2
036D  F1          1130          pop   af               ;restore all registers used
036E  C1          1131          pop   bc               ;value can be found at either
036F  D1          1132          pop   de               ;either Param1, Param2
0370  E1          1133          pop   hl               ;  or   Param3, Param4
0371  C9          1134          ret
                  1135  ;
                  1136  ;
                  1137  ; IQPUSH  Instruction Queue Return Stack Push              PUBLIC-SUBROUTINE
                  1138  ;         Pushes current instruction onto Instruction Query Return Stack
                  1139  ;         (IQRS)<--(ILoopV)  (IQRSP)=(IQRSP+2)
                  1140  ;-------------------------------------------------------------------------
                  1141  ;
0372  E5          1142  iqpush: push  hl               ;save registers to be used
0373  D5          1143          push  de
0374  2A06FF      1144          ld    hl,(SysWrd + 6)  ;ILoopV current task
0377  5D          1145          ld    e,l
0378  54          1146          ld    d,h              ;DE=current Task (ILoopV)
0379  2A4AFF      1147          ld    hl,(SysWrd + 74) ;IQRSP
037C  73          1148          ld    (hl),e
037D  23          1149          inc   hl
037E  72          1150          ld    (hl),d           ;push (ILoopV) onto IQRS
037F  23          1151          inc   hl               ;HL=next entry
0380  224AFF      1152          ld    (SysWrd + 74),hl ;update (IQRSP)
0383  215AFF      1153          ld    hl,SysByt + 6    ;point to IQRSDC
0386  34          1154          inc   (hl)             ;depth + 1
0387  D1          1155          pop   de               ;restore registers used
0388  E1          1156          pop   hl
0389  C9          1157          ret
                  1158  ;
                  1159  ;
                  1160  ; IQPOP Instruction Queue Return Stack Pop                 PUBLIC-SUBROUTINE
                  1161  ;       Pops topmost item of Instruction Queue Return Stack back at current
                  1162  ;       (ILoopV)<--(IQRS)  (IQRSP)=(IQRSP-2)
                  1163  ;-------------------------------------------------------------------------
                  1164  ;
038A  E5          1165  iqpop:  push  hl               ;save registers to be used
038B  D5          1166          push  de
038C  2A4AFF      1167          ld    hl,(SysWrd + 74) ;IQRSP
038F  2B          1168          dec   hl
0390  56          1169          ld    d,(hl)
0391  2B          1170          dec   hl
0392  5E          1171          ld    e,(hl)           ;DE=last Task
0393  224AFF      1172          ld    (SysWrd + 74),hl ;update (IQRSP)
0396  EB          1173          ex    de,hl            ;HL= last task
0397  2206FF      1174          ld    (SysWrd + 6),hl  ;ILoopV
039A  215AFF      1175          ld    hl,SysByt + 6    ;point to IQRSDC
039D  35          1176          dec   (hl)             ;depth - 1
039E  D1          1177          pop   de               ;restore registers used
039F  E1          1178          pop   hl
03A0  C9          1179          ret
                  1180  ;
                  1181  ;
```

```
                1182  ; IQDROP Instruction Queue Return Stack Drop one entry      PUBLIC-SUBROUTINE
                1183  ;        Drops most recent pushed item from Instruction Queue Return Stack
                1184  ;        (IQRSP)=(IQRSP-2)
                1185  ;-------------------------------------------------------------------------
                1186  ;
03A1  E5        1187  iqdrop: push  hl              ;save registers to be used
03A2  2A4AFF    1188          ld    hl,(SysWrd + 74)  ;IQRSP
03A5  2B        1189          dec   hl
03A6  2B        1190          dec   hl
03A7  224AFF    1191          ld    (SysWrd + 74),hl  ;update (IQRSP)
03AA  215AFF    1192          ld    hl,SysByt + 6     ;point to IQRSDC
03AD  35        1193          dec   (hl)              ;depth - 1
03AE  E1        1194          pop   hl
03AF  C9        1195          ret
                1196  ;
                1197  ;
                1198  ; IQTEST Check for Instruction Queue Return Stack           PUBLIC-SUBROUTINE
                1199  ;        If IQR-Stack is full, changes system prompt to "<"
                1200  ;        On Entry:
                1201  ;        On Exit:  case (IQRSP) < 65534 Zero-Flag=NZ
                1202  ;                  case (IQRSP) > 65533 Zero-Flag= Z
                1203  ;                                       (SysByt+0)=(Prompt)='<'
                1204  ;-------------------------------------------------------------------------
                1205  ;
03B0  E5        1206  iqtest: push  hl              ;save register
03B1  2A4AFF    1207          ld    hl,(SysWrd + 74)  ;IQRSP
03B4  2C        1208          inc   l                 ;should not be zero after two
03B5  2C        1209          inc   l                 ;increments, else now 65536 = 0
03B6  2005      1210          jr    nz,iqtes1
03B8  2154FF    1211          ld    hl,SysByt         ;Prompt
03BB  363C      1212          ld    (hl),'<'          ;warning, IQRSP low
03BD  E1        1213  iqtes1: pop   hl
03BE  C9        1214          ret
                1215  ;
                1216  ;
                1217  ; IQDUMP Dumps one Instruction-Vector from IQRS             PUBLIC-SUBROUTINE
                1218  ;        The first item pushed onto the Instruction Queue Return Stack
                1219  ;        is removed and the whole stack contents shifted to make room
                1220  ;        for one push.
                1221  ;        On Entry:
                1222  ;        On Exit:  (IQRSP)=(IWRSP-2)
                1223  ;                  IQSP top to IQSP bottom+2 --> IQSP top-2 IQSP bottom
                1224  ;                  IQRSP points to same stack entry as before
                1225  ;-------------------------------------------------------------------------
                1226  ;
03BF  E5        1227  iqdump: push  hl              ;save registers
03C0  D5        1228          push  de
03C1  C5        1229          push  bc
03C2  F5        1230          push  af
03C3  21C2FF    1231          ld    hl,SysBuf + 66    ;IQRS+2
03C6  11C0FF    1232          ld    de,SysBuf + 64    ;IQRS bottom
03C9  013E00    1233          ld    bc,62             ;62 bytes, 31 entries
03CC  EDB0      1234          ldir                    ;move stack
03CE  F1        1235          pop   af                ;restore registers
03CF  C1        1236          pop   bc
03D0  D1        1237          pop   de
03D1  E1        1238          pop   hl
03D2  CDA103    1239          call  iqdrop            ;point to same entry as before
03D5  C9        1240          ret
                1241  ;
                1242  ;
```

```
                1243  ; BINASC Converts 1 byte into 2 hex-ascii characters        PUBLIC-SUBROUTINE
                1244  ;        On Entry: A-register has character to be converted
                1245  ;        On Exit:  A-register as on entry, F-register may be different
                1246  ;                  (SysWrd+72)=HexChr, converted byte in 2 ascii chars
                1247  ;-------------------------------------------------------------------------------
                1248  ;
03D6  E5        1249  binasc: push  hl              ;save registers used
03D7  F5        1250         push  af
03D8  67        1251         ld    h,a             ;save byte
03D9  CB3F      1252         srl   a               ;shift high nibble to low nibble
03DB  CB3F      1253         srl   a               ;and remove high
03DD  CB3F      1254         srl   a               ;by logical right shifting
03DF  CB3F      1255         srl   a
03E1  FE0A      1256         cp    10              ;check 0...9
03E3  3802      1257         jr    c,binas1        ;jump if 0...9
03E5  C607      1258         add   a,7             ;offset for A...F
03E7  C630      1259  binas1: add   a,48            ;convert to ascii character
03E9  6F        1260         ld    l,a             ;high nibble first
03EA  7C        1261         ld    a,h             ;get byte again
03EB  E60F      1262         and   15              ;remove high nibble
03ED  FE0A      1263         cp    10              ;check 0...9
03EF  3802      1264         jr    c,binas2        ;jump if 0...9
03F1  C607      1265         add   a,7             ;offset for A...F
03F3  C630      1266  binas2: add   a,48            ;convert to ascii character
03F5  67        1267         ld    h,a             ;low nibble last
03F6  2248FF    1268         ld    (SysWrd + 72),hl  ;HexChr
03F9  F1        1269         pop   af              ;restore registers
03FA  E1        1270         pop   hl
03FB  C9        1271         ret
                1272  ;
                1273  ;
                1274  ; ASCBIN Converts 2 hex-ascii characters into 1 byte        PUBLIC-SUBROUTINE
                1275  ;        On Entry: (SysWrd+72)=HexChr, 2 chars to convert to binary
                1276  ;        On Exit:  A-register has byte, F-register may be altered
                1277  ;                  (SysWrd+72)=HexChr
                1278  ;-------------------------------------------------------------------------------
                1279  ;
03FC  E5        1280  ascbin: push  hl              ;save registers used
03FD  2A48FF    1281         ld    hl,(SysWrd + 72)  ;HexChr
0400  7D        1282         ld    a,l             ;A=high nibble in ascii
0401  CDC904    1283         call  chknbr          ;CY: (A) not ['1','9']
0404  DCD004    1284         call  c,chkaf         ;CY: (A) not ['A','F'] or ['a','f']
0407  3829      1285         jr    c,ascbi4        ;illegal nibble, store 0
0409  CB77      1286         bit   6,a             ;if set, not '0'...'9'
040B  2804      1287         jr    z,ascbi1        ;not set, number
040D  CBAF      1288         res   5,a             ;convert to uppercase
040F  D607      1289         sub   a,7             ;subtract offset
0411  D630      1290  ascbi1: sub   a,48            ;convert to binary nibble
0413  CB27      1291         sla   a               ;move to high nibble position
0415  CB27      1292         sla   a               ;and clear low
0417  CB27      1293         sla   a               ;by logical left shifting
0419  CB27      1294         sla   a
041B  6F        1295         ld    l,a             ;replace char by high nibble
041C  7C        1296         ld    a,h             ;get low nibble character
041D  CDC904    1297         call  chknbr          ;CY: (A) not ['1','9']
0420  DCD004    1298         call  c,chkaf         ;CY: (A) not ['A','F'] or ['a','f']
0423  380D      1299         jr    c,ascbi4        ;illegal nibble, store 0
0425  CB77      1300         bit   6,a             ;if set, not '0'...'9'
0427  2804      1301         jr    z,ascbi2        ;not set, number
0429  CBAF      1302         res   5,a             ;convert to uppercase
042B  D607      1303         sub   a,7             ;subtract offset
042D  D630      1304  ascbi2: sub   a,48            ;convert to binary nibble
042F  B5        1305         or    l               ;combine low with high - A=result
0430  E1        1306  ascbi3: pop   hl              ;restore registers
0431  C9        1307         ret
0432  AF        1308  ascbi4: xor   a               ;invalid byte, returns 0
0433  18FB      1309         jr    ascbi3          ;store 0
                1310  ;
                1311  ;
```

```
                1312  ; LinDmp - Line Dump, dumps 16 Bytes                        PUBLIC-SUBROUTINE
                1313  ;          On Entry: Param3=aaaa - start address to dump from
                1314  ;          On Exit:  Param3=aaaa
                1315  ;                    SysBuf has 58 bytes as
                1316  ;          "aaaa : bb bb bb bb bb bb bb bb  bb bb bb bb bb bb bb bb | "
                1317  ;-------------------------------------------------------------------------
                1318  ;
0435  E5         1319  lindmp: push  hl               ;save registers
0436  D5         1320          push  de
0437  C5         1321          push  bc
0438  F5         1322          push  af
0439  1180FF     1323          ld    de,SysBuf         ;destination
043C  2A44FF     1324          ld    hl,(SysWrd + 68)  ;Param3, start address
043F  7C         1325          ld    a,h               ;get high byte of address
0440  CDD603     1326          call  binasc            ;HexChr, high byte
0443  7D         1327          ld    a,l               ;get low byte of address
0444  2148FF     1328          ld    hl,SysWrd + 72    ;HexChr, source
0447  010200     1329          ld    bc,2              ;BC=counter for 2 bytes
044A  EDB0       1330          ldir
044C  2B         1331          dec   hl
044D  2B         1332          dec   hl                ;HL=HexChr
044E  0E02       1333          ld    c,2               ;BC=counter for 2 bytes
0450  CDD603     1334          call  binasc            ;convert low address byte
0453  EDB0       1335          ldir                    ;copy it
0455  3E20       1336          ld    a,blank
0457  12         1337          ld    (de),a
0458  13         1338          inc   de
0459  3E3A       1339          ld    a,':'             ;address delimiter
045B  12         1340          ld    (de),a            ;insert it
045C  13         1341          inc   de                ;next buffer address
045D  3E20       1342          ld    a,blank
045F  12         1343          ld    (de),a
0460  13         1344          inc   de
0461  0610       1345          ld    b,16              ;16 bytes to prepare
0463  2A44FF     1346          ld    hl,(SysWrd + 68)  ;Param3, current address
0466  7E         1347  lindm0: ld    a,(hl)            ;get byte
0467  23         1348          inc   hl                ;next location
0468  E5         1349          push  hl                ;save source pointer
0469  CDD603     1350          call  binasc            ;convert byte
046C  2148FF     1351          ld    hl,SysWrd + 72    ;point to HexChr
046F  78         1352          ld    a,b               ;save 16-counter
0470  010200     1353          ld    bc,2              ;BC=counter for 2 bytes
0473  EDB0       1354          ldir                    ;copy byte
0475  E1         1355          pop   hl                ;restore source pointer
0476  47         1356          ld    b,a               ;restore 16-counter
0477  3E20       1357          ld    a,blank           ;blank
0479  12         1358          ld    (de),a            ;insert a blank
047A  13         1359          inc   de                ;point to next destination
047B  3E09       1360          ld    a,9
047D  B8         1361          cp    b                 ;if in middle, insert a 2nd blank
047E  2004       1362          jr    nz,lindm1
0480  3E20       1363          ld    a,blank           ;blank
0482  12         1364          ld    (de),a            ;insert a blank
0483  13         1365          inc   de                ;point to next destination
0484  10E0       1366  lindm1: djnz  lindm0            ;convert 16 bytes
0486  3E7C       1367          ld    a,'|'
0488  12         1368          ld    (de),a
0489  13         1369          inc   de
048A  3E20       1370          ld    a,blank
048C  12         1371          ld    (de),a
```

```
048D  F1          1372          pop   af              ;restore registers
048E  C1          1373          pop   bc
048F  D1          1374          pop   de
0490  E1          1375          pop   hl
0491  C9          1376          ret                   ;58 bytes ready in SysBuf+0
                  1377  ;
                  1378  ;
                  1379  ; DmpAsc - DumpAscii, dumps 16 Bytes as ascii              PUBLIC-SUBROUTINE
                  1380  ;          On Entry: Param3=aaaa - start address to dump from
                  1381  ;          On Exit:  Param3=aaaa+16
                  1382  ;                    SysBuf has 18 bytes as
                  1383  ;          "abcdefghijklmno"<cr><lf>
                  1384  ;-------------------------------------------------------------------------
                  1385  ;
0492  E5          1386  dmpasc: push  hl              ;save registers
0493  D5          1387          push  de
0494  C5          1388          push  bc
0495  F5          1389          push  af
0496  1180FF      1390          ld    de,SysBuf       ;destination
0499  2A44FF      1391          ld    hl,(SysWrd + 68)  ;Param3, start address
049C  0610        1392          ld    b,16            ;16 bytes to do
049E  0E2E        1393          ld    c,'.'           ;replaces character if unprintable
04A0  7E          1394  dmpas0: ld    a,(hl)
04A1  CBBF        1395          res   7,a             ;convert to [0,127]
04A3  FE20        1396          cp    blank           ;must be >= blank
04A5  3804        1397          jr    c,dmpas1        ;is < blank
04A7  FE7F        1398          cp    127             ;only codes are [32,126]
04A9  2001        1399          jr    nz,dmpas2       ;is not 127
04AB  79          1400  dmpas1: ld    a,c
04AC  12          1401  dmpas2: ld    (de),a          ;store character
04AD  13          1402          inc   de              ;next dest
04AE  23          1403          inc   hl              ;next srce
04AF  10EF        1404          djnz  dmpas0          ;repeat 16 times
04B1  3E0D        1405          ld    a,cr            ;insert cr/lf-sequence
04B3  12          1406          ld    (de),a
04B4  13          1407          inc   de              ;point to next
04B5  3E0A        1408          ld    a,lf            ;insert line feed
04B7  12          1409          ld    (de),a
04B8  2244FF      1410          ld    (SysWrd + 68),hl  ;update Param3
04BB  F1          1411          pop   af              ;restore registers
04BC  C1          1412          pop   bc
04BD  D1          1413          pop   de
04BE  E1          1414          pop   hl
04BF  C9          1415          ret                   ;19 bytes ready in SysBuf+0
                  1416  ;
                  1417  ;
                  1418  ; CRLFS  - Sends carriage return-linefeed sequence         PUBLIC-SUBROUTINE
                  1419  ;          On Entry:
                  1420  ;          On Exit:
                  1421  ;-------------------------------------------------------------------------
                  1422  ;
04C0  F5          1423  crlfs:  push  af
04C1  3E0D        1424          ld    a,cr            ;carriage return
04C3  EF          1425          rst   conout          ;send CR
04C4  3E0A        1426          ld    a,lf            ;line feed
04C6  EF          1427          rst   conout          ;send LF
04C7  F1          1428          pop   af              ;restore register
04C8  C9          1429          ret
                  1430  ;
1431  ;
                  1432  ; CHKNBR - Check Number                                        RST-28
                  1433  ;          On Entry:  A = character to test
                  1434  ;          On Exit:   A = character tested
                  1435  ;                    NC = '0' <= (A) <= '9' valid number
                  1436  ;                     C = character in A is not a number (0...9)
                  1437  ;          Bytes:  7  t-cycles worst case: 36  execution time: 9 Ms
                  1438  ;-------------------------------------------------------------------------
                  1439  ;
04C9  FE30        1440  ChkNbr: cp    48              ;A >= '0'?
04CB  D8          1441          ret   c               ;A < '0', CY set, no number
04CC  FE3A        1442          cp    58              ;A <= '9'?
04CE  3F          1443          ccf                   ;A > '9', CY set, no number
04CF  C9          1444          ret                   ;NC  '0' >= A >= '9'
                  1445  ;
                  1446  ;
```

```
                    1447  ; CHKAF  Tests whether byte is a...f or A...F              PUBLIC-SUBROUTINE
                    1448  ;        On Entry: (A)=byte to test
                    1449  ;        On Exit:  (A)=(A)
                    1450  ;                  case NC: (A)=['A','F'] or ['a','f']
                    1451  ;                  case CY: (A) not within tested range
                    1452  ;-------------------------------------------------------------------------
                    1453  ;
04D0  C5            1454  chkaf:  push  bc              ;save BC
04D1  4F            1455          ld    c,a             ;copy byte
04D2  CBAF          1456          res   5,a             ;convert to upper case
04D4  FE41          1457          cp    'A'             ;CY: (A) < 'A'
04D6  38F8          1458          jr    c,chkaf         ;not valid
04D8  FE47          1459          cp    'G'             ;NC: (A) > 'F'
04DA  3F            1460          ccf                   ;CY: (A) > 'F'
04DB  79            1461  chkaf1: ld    a,c             ;restore original byte
04DC  C1            1462          pop   bc              ;restore BC
04DD  C9            1463          ret
                    1464  ;
                    1465  ;
                    1466  ; MULTEN - Multiplicate by Ten (10)                       PUBLIC-SUBROUTINE
                    1467  ;          On Entry: (HL) = value to multiply by 10
                    1468  ;          On Exit:  (HL) = Entry-HL times ten [HL=HL*10]
                    1469  ;                    (BC) = Entry-HL times two [BC=HL* 2]
                    1470  ;          Bytes:    7  t-cycles: 62  execution time: 15.5 Ms
                    1471  ;-------------------------------------------------------------------------
                    1472  ;
04DE  29            1473  MulTen: add   hl,hl           ;HL = HL *  2
04DF  44            1474          ld    b,h
04E0  4D            1475          ld    c,l             ;BC = HL *  2
04E1  29            1476          add   hl,hl           ;HL = HL *  4
04E2  29            1477          add   hl,hl           ;HL = HL *  8
04E3  09            1478          add   hl,bc           ;HL = HL * 10
04E4  C9            1479          ret                   ;HL = HL * 10, BC = BC * 2
                    1480  ;
                    1481  ;
                    1482  ; DIV16  Divides BC by 16                                 PUBLIC-SUBROUTINE
                    1483  ;        On Entry: (BC)=value
                    1484  ;        On Exit:  (BC)=value/16
                    1485  ;-------------------------------------------------------------------------
1486  ;
04E5  F5            1487  div16:  push  af              ;save register
04E6  78            1488          ld    a,b             ;[FEDC|BA98 7654|3210] save B
04E7  CB39          1489          srl   c               ;[-765|4321]
04E9  CB39          1490          srl   c               ;[--76|5432]
04EB  CB39          1491          srl   c               ;[---7|6543]
04ED  CB39          1492          srl   c               ;[----|7654] C=C/16
04EF  CB27          1493          sla   a               ;[EDCB|A98-]
04F1  CB27          1494          sla   a               ;[DCBA|98--]
04F3  CB27          1495          sla   a               ;[CBA9|8---]
04F5  CB27          1496          sla   a               ;[BA98|----]
04F7  B1            1497          or    c               ;[BA98|7654]
04F8  4F            1498          ld    c,a             ;C: high nibble of C, low of B
04F9  CB38          1499          srl   b               ;[-FED|CBA9]
04FB  CB38          1500          srl   b               ;[--FE|DCBA]
04FD  CB38          1501          srl   b               ;[---F|EDCB]
04FF  CB38          1502          srl   b               ;[----|FEDC|BA98|7654] BC=BC/16
0501  F1            1503          pop   af              ;restore register
0502  C9            1504          ret
                    1505  ;
                    1506  ;
                    1507  ; BINDEC Converts an integer to ascii-decimal number       PUBLIC-SUBROUTINE
                    1508  ;        On Entry: (SysBuf+0) = unsigned integer
                    1509  ;        On Exit:  (SysBuf+2) = result in 5 characters
                    1510  ;-------------------------------------------------------------------------
                    1511  ;
0503  E5            1512  BinDec: push  hl              ;save registers
0504  D5            1513          push  de
0505  C5            1514          push  bc
0506  F5            1515          push  af
0507  2A80FF        1516          ld    hl,(SysBuf)     ;HL=integer
050A  1182FF        1517          ld    de,SysBuf + 2   ;address of result
050D  011027        1518          ld    bc,10000        ;[6]5535
0510  CD1603        1519          call  DivHBA          ;find ten-thousands
0513  01E803        1520          ld    bc,1000         ;6[5]535
0516  CD1603        1521          call  DivHBA          ;find thousands
```

```
0519  016400      1522          ld    bc,100            ;65[5]35
051C  CD1603      1523          call  DivHBA            ;find hundreds
051F  0E0A        1524          ld    c,10              ;655[3]5
0521  CD1603      1525          call  DivHBA            ;find tens
0524  7D          1526          ld    a,l               ;6553[5] = L
0525  C630        1527          add   a,48              ;convert to ascii
0527  12          1528          ld    (de),a            ;Store units
0528  F1          1529          pop   af                ;restore registers
0529  C1          1530          pop   bc
052A  D1          1531          pop   de
052B  E1          1532          pop   hl
052C  C9          1533          ret
                  1534  ;
                  1535  ;
                  1536  ; DECBIN Converts an ascii-decimal number to an integer     PUBLIC-SUBROUTINE
                  1537  ;        On Entry: (SysBuf+2) = ascii-number in 5 characters
                  1538  ;        On Exity: (SysBuf+0) = resulting unsigned integer
                  1539  ;-------------------------------------------------------------------------
                  1540  ;
052D  E5          1541  DecBin: push  hl              ;save registers
052E  D5          1542          push  de
052F  C5          1543          push  bc
0530  F5          1544          push  af
0531  1182FF      1545          ld    de,SysBuf + 2     ;ten-thousands
0534  210000      1546          ld    hl,0              ;clear accumulator
0537  CD2403      1547          call  PrpUnt            ;get ten-thousands at unit-pos
053A  CDDE04      1548          call  MulTen            ;TT to tenth-position
053D  CD2403      1549          call  PrpUnt            ;get thousands at unit-pos
0540  CDDE04      1550          call  MulTen            ;TT to hundreth-position
0543  CD2403      1551          call  PrpUnt            ;get hundrets at unit-pos
0546  CDDE04      1552          call  MulTen            ;TT to thousandth-position
0549  CD2403      1553          call  PrpUnt            ;get tens at unit-pos
054C  CDDE04      1554          call  MulTen            ;TT to ten-thousandth-position
054F  CD2403      1555          call  PrpUnt            ;get units at unit-pos
0552  EB          1556          ex    de,hl             ;DE=result
0553  2180FF      1557          ld    hl,SysBuf         ;point to result buffer
0556  73          1558          ld    (hl),e            ;insert lower byte
0557  23          1559          inc   hl
0558  72          1560          ld    (hl),d            ;insert higher byte
0559  F1          1561          pop   af                ;restore registers
055A  C1          1562          pop   bc
055B  D1          1563          pop   de
055C  E1          1564          pop   hl
055D  C9          1565          ret
                  1566  ;
                  1567  ;
                  1568  ; SETPAR Sets up Param1 and Param2                          PUBLIC-SUBROUTINE
                  1569  ;        Prepares parameters entered so that C?pntr becomes C?addr and
                  1570  ;        C?cntr zero.
                  1571  ;        On Entry:
                  1572  ;        On Exit:  Param3=Param1
                  1573  ;                  Param4=0
                  1574  ;-------------------------------------------------------------------------
                  1575  ;
055E  E5          1576  setpar: push  hl              ;save register
055F  2A40FF      1577          ld    hl,(SysWrd + 64)  ;HL=Param1
0562  2244FF      1578          ld    (SysWrd + 68),hl  ;Param3=Param1
0565  210000      1579          ld    hl,0              ;0
0568  2246FF      1580          ld    (SysWrd + 70),hl  ;Param4=0
056B  E1          1581          pop   hl                ;restore register
056C  C9          1582          ret
                  1583  ;
                  1584  ;
```

```
                    1585  ; SETVAR Sets up Command variables                        PUBLIC-SUBROUTINE
                    1586  ;        Copies content of Param1 to 4 to C?parameter_block
                    1587  ;        On Entry: (SysByt+5)=CmdNbr
                    1588  ;        On Exit:  C?addr=Param1
                    1589  ;                  C?byte=Param2
                    1590  ;                  C?pntr=Param3, usually Param1
                    1591  ;                  C?cntr=Param4, usually 0
                    1592  ;-------------------------------------------------------------------------------
                    1593  ;
056D  E5            1594  setvar: push  hl                ;save registers used
056E  D5            1595          push  de
056F  C5            1596          push  bc
0570  F5            1597          push  af
0571  3A59FF        1598          ld    a,(SysByt + 5)    ;CmdNbr
0574  87            1599          add   a,a               ;*2 for words
0575  87            1600          add   a,a               ;*4 for parameters
0576  87            1601          add   a,a               ;*8 total 8 bytes
0577  4F            1602          ld    c,a
0578  0600          1603          ld    b,0               ;BC=8, number of bytes
057A  2108FF        1604          ld    hl,SysWrd + 8     ;start of vars-block CTaddr
057D  09            1605          add   hl,bc             ;HL points to appropriate block
057E  1140FF        1606          ld    de,SysWrd + 64    ;start of para-block Param1
0581  EB            1607          ex    de,hl             ;HL=parablk, DE=varsblk
0582  0E08          1608          ld    c,8               ;8 bytes for 4 words
0584  EDB0          1609          ldir                    ;copy Para1-4 to C?nnnn
0586  F1            1610          pop   af                ;restore registers
0587  C1            1611          pop   bc
0588  D1            1612          pop   de
0589  E1            1613          pop   hl
058A  C9            1614          ret
                    1615  ;
                    1616  ;
                    1617  ; SVINIT - System Variables Initialization table
                    1618  ;===============================================================================
                    1619  ;
                    1620  ;
                    1621  ; SYSTEM WORDS AT FF00 TO FF5F (48 unsigned word variables).
                    1622  ;   Label in RAM: SysWrd (IX-register).
                    1623  ;-------------------------------------------------------------------------------
                    1624  ;
058B  0000          1625  svinit: dw      0              ;+ 0  1:MCFadr Memory Check Fail address
058D  3C00          1626          dw    im1ret           ;+ 2  2:IMode1 interrupt mode-1 vector
058F  AD00          1627          dw    warmst           ;+ 4  3:NMIVec non-maskable int vector
0591  7000          1628          dw    dumyap           ;+ 6  4:ILoopV Infinite Loop Vector
0593  0000          1629          dw      0              ;+ 8  5:CTaddr Command TASK address
0595  0000          1630          dw      0              ;+10  6:CTbyte Commmad TASK bytes expected
0597  0000          1631          dw      0              ;+12  7:CTbptr Command TASK buffer pointer
0599  0000          1632          dw      0              ;+14  8:CTcntr Command TASK counter
059B  0000          1633          dw      0              ;+16  9:CEaddr Command EXEC address
059D  0000          1634          dw      0              ;+18 10:CEbyte Commmad EXEC bytes expected
059F  0000          1635          dw      0              ;+20 11:CEbptr Command EXEC buffer pointer
05A1  0000          1636          dw      0              ;+22 12:CEcntr Command EXEC counter
05A3  0000          1637          dw      0              ;+24 13:CDaddr Command DNLD address
05A5  0000          1638          dw      0              ;+26 14:CDbyte Commmad DNLD bytes expected
05A7  0000          1639          dw      0              ;+28 15:CDbptr Command DNLD buffer pointer
05A9  0000          1640          dw      0              ;+30 16:CDcntr Command DNLD counter
05AB  0000          1641          dw      0              ;+32 17:CUaddr Command UPLD address
05AD  0000          1642          dw      0              ;+34 18:CUbyte Commmad UPLD bytes to send
05AF  0000          1643          dw      0              ;+36 19:CUpntr Command UPLD buffer pointer
05B1  0000          1644          dw      0              ;+38 20:CUcntr Command UPLD counter
05B3  54FF          1645          dw    SysByt           ;+40 21:CHaddr Command HELO address
05B5  0100          1646          dw      1              ;+42 22:CHbyte Commmad HELO bytes to send
05B7  54FF          1647          dw    SysByt           ;+44 23:CHpntr Command HELO buffer pointer
05B9  0000          1648          dw      0              ;+46 24:CHcntr Command HELO counter
05BB  80FF          1649          dw    SysBuf           ;+48 25:CMaddr Command MONI address
05BD  0000          1650          dw      0              ;+50 26:CMbyte Command MONI bytes
05BF  80FF          1651          dw    SysBuf           ;+52 27:CMpntr Command MONI pointer
05C1  0000          1652          dw      0              ;+54 28:CMcntr Command MONI counter
05C3  3008          1653          dw    ExtCmd           ;+56 29:CAaddr Command AUXY address
05C5  0000          1654          dw      0              ;+58 30:CAbyte Command AUXY byte
05C7  0000          1655          dw      0              ;+60 31:CApntr Command AUXY pointer
05C9  0000          1656          dw      0              ;+62 32:CAcntr Command AUXY counter
05CB  0000          1657          dw      0              ;+64 33:Param1 1st parameter
05CD  0000          1658          dw      0              ;+66 34:Param2 2nd parameter
05CF  0000          1659          dw      0              ;+68 35:Param3 3rd parameter
```

```
05D1  0000        1660          dw      0               ;+70 36:Param4 4th parameter
05D3  0000        1661          dw      0               ;+72 37:HexChr can hold ascii-hex value
05D5  C0FF        1662          dw    SysBuf + 64       ;+74 38:IQRSP  Instruction Queue Ret Stk Ptr
05D7  3400        1663          dw    im0ret            ;+76 39:IMode0 interrupt mode-0 30h vector
05D9  0000        1664          dw      0               ;+78 40:spare
05DB  0000        1665          dw      0               ;+80 41:spare
05DD  0000        1666          dw      0               ;+82 42:spare
                  1667  ;
                  1668  ;
                  1669  ; SYSTEM BYTES AT FF54 TO FF61 (14 unsigned byte variables) SysWrd+84
                  1670  ;  Label in RAM: SysByt (IY-register).
                  1671  ;-------------------------------------------------------------------------
                  1672  ;
05DF  3A          1673          db    ':'               ;+ 0  1:Prompt Initial system prompt
05E0  2D          1674          db    '-'               ;+ 1  2:CurCmd Current Command
05E1  2D          1675          db    '-'               ;+ 2  3:OldCmd Last Command
05E2  00          1676          db      0               ;+ 3  4:Param0 flags Param1 or Param2
05E3  00          1677          db      0               ;+ 4  5:MonMod Monitor Mode flags
05E4  00          1678          db      0               ;+ 5  6:CmdNbr Command Number
05E5  00          1679          db      0               ;+ 6  7:IQRSDC IQRS-Depth Counter
05E6  00          1680          db      0               ;+ 7  8:
05E7  00          1681          db      0               ;+ 8  9:
05E8  00          1682          db      0               ;+ 9 10:
05E9  00          1683          db      0               ;+10 11:
05EA  00          1684          db      0               ;+11 12:
05EB  00          1685          db      0               ;+12 13:
05EC  00          1686          db      0               ;+13 14:
                  1687  ;
                  1688  ;
                  1689  ; ALLISR All Interrupt service routines jump table (SysWrd+98, SysByt+14)
                  1690  ;        IM2TBL at 0050 (ROM-Start+80) point to here. All routines must
                  1691  ;        return with a RETI-instruction
                  1692  ;        FF62 to FF7F (65378-65407)
                  1693  ;-------------------------------------------------------------------------
                  1694  ;
05ED  C3AD00      1695          jp    warmst            ;SIO-A  0050=80                         FF62
05F0  C30401      1696          jp    sioisr            ;SIO-B  0052=82                         FF65
05F3  C3AD00      1697          jp    warmst            ;PIO-A  0054=84                         FF68
05F6  C3AD00      1698          jp    warmst            ;PIO-B  0056=86                         FF6B
05F9  C3AD00      1699          jp    warmst            ;CTC-0  0058=88                         FF6E
05FC  C3AD00      1700          jp    warmst            ;CTC-1  005A=90                         FF71
05FF  C3AD00      1701          jp    warmst            ;CTC-2  005C=92                         FF74
0602  C3AD00      1702          jp    warmst            ;CTC-3  005E=94                         FF77
0605  C3AD00      1703          jp    warmst            ;Spare1 0060=96                         FF7A
0608  C3AD00      1704          jp    warmst            ;Spare2 0062=98                         FF7D
                  1705  ;
                  1706  ;
                  1707  ; BUFFER AREA FF80 TO FFFF (SysBuf+0 - SysWrd+128)
                  1708  ;   Label in RAM: SysBuf
                  1709  ;-------------------------------------------------------------------------
                  1710  ;
060B  (0040)      1711          ds     64               ;+ 0 DEFBUF default buffer
064B  (003F)      1712          ds     63               ;+64 IQRS Instruction Queue Return Stack
068A  C9          1713          db    201               ;65535: RET instruction for A command
                  1714  ;
                  1715  ;
```

```
                1716  ; PUBPRO Public Procedures (PubVec points to PubPro)
                1717  ;        Execution addresses of procedures that can be used from any
                1718  ;        application.
                1719  ;        DO NOT CHANGE SEQUENCE
                1720  ;      ! Firmware description must identify table and procedures.
                1721  ;-------------------------------------------------------------------------
                1722  ;
068B  1908      1723  pubpro: dw    useend          ;address of first unused location
068D  C506      1724          dw    uzever          ;address of EPROM-Version
                1725  ;
068F  EE02      1726          dw    crlfp           ;public procedure
0691  A002      1727          dw    dumpit          ;public procedure M<4-7>,s,b,s,n
0693  6002      1728          dw    whowho          ;public procedure E<whowho> H
                1729  ;
0695  FC03      1730          dw    ascbin          ;public subroutine
0697  D603      1731          dw    binasc          ;public subroutine
0699  0305      1732          dw    bindec          ;public subroutine
069B  D004      1733          dw    chkaf           ;public subroutine
069D  C904      1734          dw    chknbr          ;public subroutine
069F  C004      1735          dw    crlfs           ;public subroutine
06A1  2D05      1736          dw    decbin          ;public subroutine
06A3  E504      1737          dw    div16           ;public subroutine
06A5  9204      1738          dw    dmpasc          ;public subroutine
06A7  4303      1739          dw    getasc          ;public subroutine
06A9  A103      1740          dw    iqdrop          ;public subroutine
06AB  BF03      1741          dw    iqdump          ;public subroutine
06AD  8A03      1742          dw    iqpop           ;public subroutine
06AF  7203      1743          dw    iqpush          ;public subroutine
06B1  B003      1744          dw    iqtest          ;public subroutine
06B3  3504      1745          dw    lindmp          ;public subroutine
06B5  DE04      1746          dw    multen          ;public subroutine
06B7  3803      1747          dw    rtsgo           ;public subroutine
06B9  2D03      1748          dw    rtstop          ;public subroutine
06BB  5E05      1749          dw    setpar          ;public subroutine
06BD  6D05      1750          dw    setvar          ;public subroutine
                1751  ;
06BF  1B08      1752          dw    sdelay          ;public subroutine
06C1  2308      1753          dw    ldelay          ;public subroutine
06C3  1D09      1754          dw    versi0          ;publich subroutine
                1755  ;
                1756  ; UZEVER Version of the firmware
                1757  ;        This has a special format. If the first 2 bytes are 255, the
                1758  ;        version string follows immediately afterwards. These 2 bytes
                1759  ;        can be re-programmed to point to a newer version. Thus, if
                1760  ;        the first two bytes are not 255, they contain an address
                1761  ;        where an updated version number can be found.
                1762  ;        The version string can have any length. It must be terminated
                1763  ;        by a byte 255.
                1764  ;-------------------------------------------------------------------------
                1765  ;
06C5  1A07      1766  uzever: dw    uzeve1          ;vector to linked list
06C7  0D0A555A  1767          db    cr,lf,'UZE control firmware V.0.01, 940910 (c) HoroSoft, '
06FB  43482D33  1768          db    'CH-3952 Susten, Switzerland.',cr,lf
0719  FF        1769          db    255             ;end byte
071A  6F07      1770  uzeve1: dw    uzeve2          ;vector to next: none
071C  0D0A555A  1771          db    cr,lf,'UZE control firmware V.1.00, 941022 (c) HoroSoft, '
0750  43482D33  1772          db    'CH-3952 Susten, Switzerland.',cr,lf
076E  FF        1773          db    255             ;end byte
076F  C407      1774  uzeve2: dw    uzeve3          ;vector to next: none
0771  0D0A555A  1775          db    cr,lf,'UZE control firmware V.1.02, 941229 (c) HoroSoft, '
07A5  43482D33  1776          db    'CH-3952 Susten, Switzerland.',cr,lf
07C3  FF        1777          db    255             ;end byte
07C4  FFFF      1778  uzeve3: dw    65535           ;vector to next: none
07C6  0D0A555A  1779          db    cr,lf,'UZE control firmware V.2.01, 951030 (c) HoroSoft, '
07FA  43482D33  1780          db    'CH-3952 Susten, Switzerland.',cr,lf
0818  FF        1781          db    255             ;end byte
                1782  ;
                1783  ;
                1784  ; USEEND Just a label that points to the first unprogrammed ROM-location.
                1785  ;   DUMMY-label is also last word in named routines.
                1786  ;-------------------------------------------------------------------------
                1787  ;
0819  A60A      1788  useend: dw    Dummy           ;point to first free location
                1789  ;
                1790  ;
```

```
                1791  ;==================-<VERSION 2.xx EXTENSIONS>-================================
                1792  ; SDELAY - Short Delay                                        PUBLIC SUBROUTINE
                1793  ;          Produces a delay with a resolution of approx. 6.5 Ms
                1794  ;          With HL=0 produces abaut HL=0 on LongDelay routine
                1795  ;          HL = 1:  17.25 Ms (incl. call SDelay)
                1796  ;          HL = 0: 426 ms
                1797  ;          On Entry: HL = Delay
                1798  ;          On Exit:  HL = 0
                1799  ;-------------------------------------------------------------------------------
                1800  ;
081B  F5        1801  SDelay: push  af              ;save af
081C  2B        1802  sdela1: dec   hl              ;1.5 Ms
081D  7C        1803          ld    a,h             ;1.0 Ms  repeat until HL=0
081E  B5        1804          or    l               ;1.0 Ms
081F  20FB      1805          jr    nz,sdela1       ;3.0 Ms / 1.75 Ms
0821  F1        1806          pop   af              ;restore af
0822  C9        1807          ret                   ;done
                1808  ;
                1809  ;
                1810  ; LDELAY - Long Delay                                         PUBLIC SUBROUTINE
                1811  ;          Produces a delay with a resolution of approx. 425 ms
                1812  ;          or longest delay of ShortDelay routine.
                1813  ;          HL= 1: ca. 425 ms
                1814  ;          HL= 0: ca.  27.85 s
                1815  ;          See EDN (Electronic Design News), August 18, 1983:
                1816  ;            Cass R. Leward; Z80 routine provides wide delay range.
                1817  ;          On Entry: HL = delay
                1818  ;          On Exit:  HL = 0
                1819  ;-------------------------------------------------------------------------------
                1820  ;
0823  F5        1821  LDelay: push  af              ;save register used
0824  C5        1822          push  bc
0825  45        1823  ldela0: ld    b,l             ;get delay counter from HL
0826  10FE      1824  ldela1: djnz  ldela1          ;main delay loop
0828  2B        1825          dec   hl
0829  7D        1826          ld    a,l
082A  B4        1827          or    h               ;check HL=0
082B  20F8      1828          jr    nz,ldela0
082D  C1        1829          pop   bc              ;restore registers used
082E  F1        1830          pop   af
082F  C9        1831          ret                   ;done
                1832  ;
                1833  ;
```

```
                1834  ;==============================================================================
                1835  ; A-COMMAND EXTENSIONS
                1836  ;==============================================================================
                1837  ;
                1838  ; AUXY   loads execution address into ILoopV.                          COMMAND
                1839  ;        An application can be executed by its name, provided it can be
                1840  ;        found in the dictionary. The NAME has to be supplied and it is
                1841  ;        loaded into SysBuf. This name is then parsed and when a match
                1842  ;        was found, its execution adress loaded into ILoopV. The Instruction
                1843  ;        is executed in exactly the same manner as when it were called by
                1844  ;        the EXEC command. If command could not be found, uploads a ?
                1845  ;        All commands MUST end with a JumP to HOME.
                1846  ;        AF, BC, DE and HL are saved in primed registers AF', BC', DE' DE'
                1847  ;        and HL' by the interrupt service routine. If they are used, they
                1848  ;        must be saved by the application command, otherwise, the system
                1849  ;        most probably crashes! If IX and IY registers are used, they
                1850  ;        should be saved by the application program needing them.
                1851  ;        On Entry:
                1852  ;        On Exit:  CAaddr=ExtCmd  (must not be altered!)
                1853  ;                  CAbyte=unknown
                1854  ;                  CApntr=<addr>
                1855  ;                  CAcntr=0
                1856  ;                  Param1=<addr>
                1857  ;                  Param2=unknown
                1858  ;                  Param3=unknown
                1859  ;                  Param4=pointer into SysBuf where last char is
                1860  ;                  ILoopV=Param1=<addr>
                1861  ;                  (HL)=<addr>
                1862  ; Syntax: ANAME{,param1{,param2{,param3}}}.
                1863  ;------------------------------------------------------------------------------
                1864  ;
0830  CD6608      1865  ExtCmd: call  CmdNam            ;get name of command into SysBuf
0833  2180FF      1866          ld    hl,SysBuf
0836  2B          1867          dec   hl                ;one less to compensate first inc de
0837  2246FF      1868          ld    (SysWrd + 70),hl  ;start of buffer (param4)
083A  211509      1869          ld    hl,Versio         ;first word in list of extended words
083D  CD9308      1870          call  ExtAdr            ;address is in (param1) if found
0840  3D          1871          dec   a                 ;A=1->0 if found, else A<>0
0841  2806        1872          jr    z,ExtCm1          ;ok, go on
0843  3E3F        1873          ld    a,'?'             ;unknown command
0845  CD2800      1874          call  ConOut            ;send question mark
0848  C9          1875          ret                     ;return via RetIsr
0849  CDB003      1876  ExtCm1: call  IQTest            ;test instruction queue return stack depth
084C  2003        1877          jr    nz,ExtCm2         ;jump if still free space
084E  CDBF03      1878          call  IQDump            ;else, dump oldest application
0851  CD7203      1879  ExtCm2: call  IQPush            ;push interrupted application onto stack
0854  2A40FF      1880          ld    hl,(SysWrd + 64)  ;get address of new application
0857  2206FF      1881          ld    (SysWrd + 6),hl   ;into (ILoopV)
085A  C9          1882          ret                     ;pop RetIsr, JP RetIsr
                1883  ;
                1884  ;
                1885  ; HOME - Is the proper return from any named application, since the
                1886  ;        registers are reloaded and the stack adjusted accordingly
                1887  ;------------------------------------------------------------------------------
                1888  ;
085B  CD8A03      1889  HOME:   call  IQPop              ;remove terminated program from stack
085E  CD0509      1890          call  VoidSB            ;clear input buffer
0861  FB          1891          ei                      ;enable interrupts in any case
0862  2A06FF      1892          ld    hl,(SysWrd + 6)   ;get ILoopV
0865  E9          1893          jp    (hl)              ;enter eternal loop
                1894  ;
                1895  ;
```

```
                1896  ; SUBROUTINE gets text input and loads it into SysBuf until either 63
                1897  ;   characters or a full stop is entered. Valid characters are [0;9], [,],
                1898  ;   [.] and [A;Z], lower case characters are converted to upper case. If
                1899  ;   no full stop is entered, SysBuf+63 holds a full stop.
                1900  ;   ON ENTRY:
                1901  ;   ON EXIT: (SysBuf+0 ... SysBuf+63) filled.
                1902  ;-------------------------------------------------------------------------
                1903  ;
0866  E5        1904  CmdNam: push  hl              ;save HL
0867  C5        1905          push  bc              ;save BC
0868  F5        1906          push  af              ;save accu and flags
0869  2180FF    1907          ld    hl,SysBuf       ;start of buffer
086C  063F      1908          ld    b,63            ;not more than 63 characters
086E  E7        1909  CmdNa1: rst   ConIn           ;wait for a key, return it in A
086F  FE41      1910          cp    'A'             ;check if < "A"
0871  380E      1911          jr    c,CmdNa3        ;if < A, check numeric
0873  CBAF      1912          res   5,a             ;convert to upper case
0875  FE5B      1913          cp    '['             ;check if < "["
0877  30F5      1914          jr    nc,CmdNa1       ;if > Z ignor input
0879  77        1915  CmdNa2: ld    (hl),a          ;store character [A;Z], [0;9], [,]
087A  23        1916          inc   hl              ;next address
087B  10F1      1917          djnz  CmdNa1          ;repeat max. 63 times
087D  3E2E      1918          ld    a,'.'           ;force full stop at end of buffer
087F  180D      1919          jr    CmdNa4          ;buffer full
0881  CDC904    1920  CmdNa3: call  ChkNbr         ;if CY, no number
0884  30F3      1921          jr    nc,CmdNa2       ;store number
0886  FE2C      1922          cp    ','             ;check delimitter
0888  28EF      1923          jr    z,CmdNa2        ;store delimiter
088A  FE2E      1924          cp    '.'             ;check end of entry character
088C  20E0      1925          jr    nz,CmdNa1       ;ignore character
088E  77        1926  CmdNa4: ld    (hl),a          ;store delimitter, That's it.
088F  F1        1927          pop   af              ;restore accu and flag
0890  C1        1928          pop   bc              ;restore BC
0891  E1        1929          pop   hl              ;restore HL
0892  C9        1930          ret
                1931  ;
                1932  ;
                1933  ; SUBROUTINE returns in (param1) execution address of instruction found
                1934  ;   and a pointer into SysBuf (or any other) in (param4).
                1935  ;   The SysBuf is parsed for the name entered.
                1936  ;   This subroutine is derived from Extended BASIC Issue 2, Version 2.2 from
                1937  ;   June 24, 1984 for SBCS. The 1st word does not hold the routines length,
                1938  ;   however, but the address of the following command.
                1939  ;   The routine has also been designed to be more universal. It can search
                1940  ;   for any word.
                1941  ;   The search stops when the word is found. If more alpha characters follow
                1942  ;   in the input buffer, they are ignored. This can be used to add an
                1943  ;   additional character, since (param4) points to it.
                1944  ;   ON ENTRY: HL = Address of label of 1st word (e.g. VERSIO)
                1945  ;             (param4) = Address of one byte before the start of the word
                1946  ;                        to be parsed.
                1947  ;   ON EXIT:  A: 1=Instruction found, A<>1: Instruction not found
                1948  ;             (param1) = CFA or PFA (code or parameter field address)
                1949  ;             (param4) = Address of 1st delimitter in buffer
                1950  ;             NOTE: only upper case alpha characters [A;Z] and numbers [0;9]
                1951  ;                   are legal.
                1952  ;-------------------------------------------------------------------------
                1953  ;
0893  D5        1954  ExtAdr: push  de              ;save DE
0894  C5        1955          push  bc              ;save BC
0895  E5        1956  NxtRtn: push  hl              ;save E-pointer
0896  2A46FF    1957          ld    hl,(SysWrd + 70)  ;get buffer address
0899  EB        1958          ex    de,hl           ;DE=P-pointer
089A  E1        1959          pop   hl              ;restore E-pointer
089B  4E        1960          ld    c,(hl)          ;get length of routine
089C  23        1961          inc   hl
089D  46        1962          ld    b,(hl)          ;BC=start of next routine
089E  23        1963  Match:  inc   hl              ;E-pointer to 1st char in name
089F  13        1964          inc   de              ;P-pointer to 1st char in buffer
08A0  1A        1965          ld    a,(de)          ;get character from buffer
08A1  BE        1966          cp    (hl)            ;compare against defined character
08A2  28FA      1967          jr    z,Match         ;if match, get next character
08A4  CB6E      1968          bit   5,(hl)          ;check if defined is lower case
08A6  200A      1969          jr    nz,Invrs        ;lower case is last char, check match
```

```
08A8  3EFF        1970  NxtAdr: ld    a,255             ;A=end of list code
08AA  BE          1971          cp    (hl)              ;end of list check
08AB  281B        1972          jr    z,NotFnd          ;if end of list, stop searching
08AD  09          1973          add   hl,bc             ;calculate start of next instruction
08AE  60          1974          ld    h,b
08AF  69          1975          ld    l,c               ;HL=address of next routine
08B0  18E3        1976          jr    NxtRtn            ;continue search
08B2  C5          1977  Invrs:  push  bc                ;save start of next routine
08B3  46          1978          ld    b,(hl)            ;get lower case character
08B4  CBA8        1979          res   5,b               ;convert to upper case
08B6  B8          1980          cp    b                 ;check wether last one matches
08B7  C1          1981          pop   bc                ;restore start of next routine
08B8  20EE        1982          jr    nz,NxtAdr         ;no match, try with next name
08BA  23          1983          inc   hl                ;HL=start of routine (CFA or PFA)
08BB  13          1984          inc   de                ;DE=addr of 1st uncompared char in SysBuf
08BC  2240FF      1985          ld    (SysWrd + 64),hl  ;exec address into param1
08BF  EB          1986          ex    de,hl             ;HL=addr of 1st uncompared char in SysBuf
08C0  2246FF      1987          ld    (SysWrd + 70),hl  ;pointer into param4
08C3  3E01        1988          ld    a,1               ;flag successfully found
08C5  C1          1989  ExtEnd: pop   bc                ;restore BC
08C6  D1          1990          pop   de                ;restore DE
08C7  C9          1991          ret                     ;back
08C8  AF          1992  NotFnd: xor   a                 ;flag not found
08C9  18FA        1993          jr    ExtEnd            ;return without found
                  1994  ;
                  1995  ;
                  1996  ; FETPAR - Fetch parameter
                  1997  ;          ON ENTRY: A = parameter number [1;3]
                  1998  ;                    (param4)   = pointer to delimitter byte in SysBuf
                  1999  ;          ON EXIT:  case A=1: (param1) = parameter 1
                  2000  ;                    case A=2: (param2) = parameter 2
                  2001  ;                    case A=3: (param3) = parameter 3
                  2002  ;                    (param4): updated to point to next delimitter byte
                  2003  ;                    A = low byte offset into param1
                  2004  ;          NOTE: see note for GetAsc.
                  2005  ;-------------------------------------------------------------------------
                  2006  ;
08CB  C5          2007  FetPar: push  bc                ;save BC
08CC  D5          2008          push  de                ;save DE
08CD  E5          2009          push  hl                ;save HL
08CE  F5          2010          push  af                ;save parameter number
08CF  2A46FF      2011          ld    hl,(SysWrd + 70)  ;HL=(param4)=pointer into SysBuf
08D2  44          2012          ld    b,h
08D3  4D          2013          ld    c,l               ;BC=points to delimitter byte in SysBuf
08D4  210000      2014          ld    hl,0              ;clear accu-1
08D7  54          2015          ld    d,h
08D8  5D          2016          ld    e,l               ;clear accu-2
08D9  03          2017  FetPa0: inc   bc                ;point to next char
08DA  0A          2018          ld    a,(bc)            ;A=character
08DB  CDC904      2019          call  ChkNbr            ;NC=valid character
08DE  380D        2020          jr    c,FetPa1          ;jump if not [0;9]
08E0  D630        2021          sub   48                ;convert ascii to binary number
08E2  5F          2022          ld    e,a
08E3  1600        2023          ld    d,0               ;DE=accu-2
08E5  C5          2024          push  bc                ;save pointer
08E6  CDDE04      2025          call MulTen             ;HL * 10, BC * 2
08E9  C1          2026          pop   bc                ;restore pointer
08EA  19          2027          add   hl,de             ;HL=accu-1 (old * 10 + new)
08EB  18EC        2028          jr    FetPa0            ;repeat until illegal char
08ED  EB          2029  FetPa1: ex    de,hl             ;DE=parameter
08EE  2146FF      2030          ld    hl,SysWrd + 70    ;param4
08F1  71          2031          ld    (hl),c            ;BC points to illegal character, just as
08F2  23          2032          inc   hl                ;when called, but after parameter
08F3  70          2033          ld    (hl),b            ;update pointer for next parameter (param4)
08F4  0600        2034          ld    b,0               ;clear B
08F6  F1          2035          pop   af                ;get parameter number
08F7  3D          2036          dec   a                 ;1, 2, 3 --> 0, 1, 2
08F8  87          2037          add   a,a               ;double (2 or 4)
08F9  4F          2038          ld    c,a               ;BC=offset into SysWrd (param2 or param3)
08FA  2140FF      2039          ld    hl,SysWrd + 64    ;HL points to param1
08FD  09          2040          add   hl,bc             ;HL points to param1, param2 or param3
08FE  73          2041          ld    (hl),e
08FF  23          2042          inc   hl
0900  72          2043          ld    (hl),d            ;store integer in param1, param2 or param3
```

```
0901  E1          2044          pop   hl              ;restore HL
0902  D1          2045          pop   de              ;restore DE
0903  C1          2046          pop   bc              ;restore BC
0904  C9          2047          ret
                  2048  ;
                  2049  ;
                  2050  ; SUBROUTINE voids/clears SysBuf. This subroutine is called by HOME.
                  2051  ;-------------------------------------------------------------------------------
                  2052  ;
0905  E5          2053  VoidSB: push  hl              ;save HL
0906  C5          2054          push  bc              ;save BC
0907  2180FF      2055          ld    hl,SysBuf       ;get start of buffer
090A  0640        2056          ld    b,64            ;buffer size
090C  0E00        2057          ld    c,0             ;clear character
090E  71          2058  VoidS1: ld    (hl),c          ;clear location
090F  23          2059          inc   hl              ;next location
0910  10FC        2060          djnz  VoidS1          ;clear buffer
0912  C1          2061          pop   bc              ;restore BC
0913  E1          2062          pop   hl              ;restore HL
0914  C9          2063          ret
                  2064  ;
                  2065  ;
                  2066  ;===============================================================================
                  2067  ; START OF NAMED ROUTINES
                  2068  ;   Convention:
                  2069  ;     - VER(sion is to be the 1st word in the dictionary.
                  2070  ;     - At the label, a word must be defined that points to the next word.
                  2071  ;     - After this word, the characters that make up the word must be
                  2072  ;       defined.
                  2073  ;     - For the word, only upper case letters may be used [A;Z].
                  2074  ;     - The last character of the word must be lower case [a;z].
                  2075  ;     - After the word, code or data start.
                  2076  ;     - After the last defined word, a dumy word must be defined with the
                  2077  ;       name of character code 255. The address for the non existing word
                  2078  ;       following the dummy may be (preferrably but not mandatory) 65535.
                  2079  ;       This makes it possible to 'afterburn' more words.
                  2080  ;     - There can be up to 3 parameters. They are stored in param1...3.
                  2081  ;       The parameter number [1;3] has to be passed in the A-register
                  2082  ;       before FetPar is called.
                  2083  ;===============================================================================
                  2084  ;
                  2085  ; VERSION - Version: returns version number of Extended UZE
                  2086  ;           Text must end with chr$(255), HL must point to text.
                  2087  ;           Can be used from label Versi0 for other string outs.
                  2088  ;
                  2089  ;   Syntax: aVer.
                  2090  ;-------------------------------------------------------------------------------
                  2091  ;
0915  7109        2092  Versio: dw    LISTVW          ;next word
0917  564572      2093          db    'VEr'           ;VER
091A  212709      2094          ld    hl,VersiN       ;HL=text start
091D  7E          2095  Versi0: ld    a,(hl)          ;get character
091E  FEFF        2096          cp    255             ;is it end?
0920  CA5B08      2097          jp    z,Home          ;if end, done
0923  EF          2098          rst   ConOut          ;send character
0924  23          2099          inc   hl              ;point to next
0925  18F6        2100          jr    Versi0          ;repeat
0927  0D0A        2101  VersiN: db    cr,lf
0929  555A4520    2102          db    'UZE Extended A-Commands, '
0942  562E312E    2103          db    'V.1.01; 951103 (c) HoroSoft, '
095F  43482D33    2104          db    'CH-3952 Susten.',cr,lf,255
                  2105  ;
                  2106  ;
```

```
                2107  ; LIST lists all defined Extended A-Command or Variable names
                2108  ;
                2109  ;      If ListV: system variable names are listed
                2110  ;      If List$: where $ any character except V: lists words in dictionary
                2111  ;
                2112  ;   Syntax: aList{,v|n}.
                2113  ;-------------------------------------------------------------------------
                2114  ;
0971  B609        2115  LISTVW:  dw    VARSET           ;next routine
0973  4C495374    2116           db    'LISt'
0977  CDC004      2117           call  crlfs            ;send a cr/lf sequence
097A  2A46FF      2118           ld    hl,(SysWrd + 70) ;get pointer
097D  23          2119           inc   hl               ;point after delimitter
097E  7E          2120           ld    a,(hl)           ;get character after LIST
097F  CBAF        2121           res   5,a              ;convert to upper case
0981  FE56        2122           cp    'V'              ;is it for variables
0983  2008        2123           jr    nz,List0         ;no, Word list is requested
0985  11EF09      2124           ld    de,Var001 + 2    ;get start
0988  2AED09      2125           ld    hl,(Var001)      ;get address of next
098B  1806        2126           jr    List1            ;search
098D  117309      2127  List0:   ld    de,LISTVW + 2    ;get start
0990  2A7109      2128           ld    hl,(LISTVW)      ;get address of next
0993  1A          2129  List1:   ld    a,(de)           ;get character of name
0994  13          2130           inc   de               ;next address
0995  CB6F        2131           bit   5,a              ;is it last?
0997  2003        2132           jr    nz,List2         ;yes, jump
0999  EF          2133           rst   ConOut           ;send character
099A  18F7        2134           jr    List1            ;repeat
099C  FEFF        2135  List2:  cp    255              ;is list ended
099E  2810        2136           jr    z,List3          ;all done
09A0  CBAF        2137           res   5,a              ;convert to upper case
09A2  EF          2138           rst   ConOut           ;send this one as well
09A3  3E2C        2139           ld    a,','
09A5  EF          2140           rst   ConOut           ;send a comma
09A6  3E20        2141           ld    a,' '
09A8  EF          2142           rst   ConOut           ;send a blank
09A9  5E          2143           ld    e,(hl)           ;HL=NEXT, E=(NEXT)
09AA  23          2144           inc   hl               ;HL=NEXT+1
09AB  56          2145           ld    d,(hl)           ;D=(NEXT+1): DE=address of next after next
09AC  23          2146           inc   hl               ;HL points to 1st char of NEXT
09AD  EB          2147           ex    de,hl            ;DE points to Name, HL to start of next
09AE  18E3        2148           jr    List1            ;repeat
09B0  CDC004      2149  List3:   call  crlfs            ;issue a cr/lf sequence
09B3  C35B08      2150           jp    Home             ;all done
                2151  ;
                2152  ;
                2153  ; VARSET - Set System Variable.
                2154  ;   varname is the name of the variable, subvar specifies which of the 4
                2155  ;   variables for T, E, U, D, H, A and paramX should be accessed:
                2156  ;     subvar is 0 for C?addr and param1 (address)
                2157  ;               1 for C?byte and param2 (byte count)
                2158  ;               2 for C?pntr and param3 (pointer)
                2159  ;               3 for C?cntr and param4 (counter)
                2160  ;   Not all variables are listed. Unlisted ones can be accessed through
                2161  ;   the subvar-offset. See example below. Actually, SETVAR is an easier
                2162  ;   form of the U(pload command.
                2163  ;
                2164  ;   EXAMPLE: aSetVar,MCFadr,74,10. will set IQRSP to 10 and the system will
                2165  ;            crash! The unnamed variables should not be "poked"!
                2166  ;
                2167  ;   Syntax: aSetVar,varname,subvar,value.
                2168  ;-------------------------------------------------------------------------
                2169  ;
09B6  630A        2170  VarSet:  dw    VARRET           ;next routine
09B8  53455456    2171           db    'SETVAr'         ;set a system variable to a value
09BE  21ED09      2172           ld    hl,Var001        ;start of table, param4 points to buffer
09C1  CD9308      2173           call  ExtAdr           ;(param1) = PFA
09C4  3D          2174           dec   a                ;A=1=0: found
09C5  2806        2175           jr    z,VarSe0         ;continue if found
09C7  3E3F        2176           ld    a,'?'            ;unknown
09C9  EF          2177           rst   ConOut           ;issue ?
09CA  C35B08      2178           jp    Home             ;return if not found
```

```
09CD  2A40FF      2179  VarSe0: ld    hl,(SysWrd + 64)  ;get PFA
09D0  5E          2180          ld    e,(hl)            ;get low byte of address
09D1  23          2181          inc   hl
09D2  56          2182          ld    d,(hl)            ;DE=address of variable
09D3  3E02        2183          ld    a,2               ;parameter 2
09D5  CDCB08      2184          call  FetPar            ;(param2) = sub-variable
09D8  3E03        2185          ld    a,3               ;parameter 3
09DA  CDCB08      2186          call  FetPar            ;(param3) = value
09DD  2A42FF      2187          ld    hl,(SysWrd + 66)  ;get (param2)
09E0  29          2188          add   hl,hl             ;2 bytes
09E1  19          2189          add   hl,de             ;DE=base address, HL points to variable
09E2  EB          2190          ex    de,hl             ;DE points to variable
09E3  2A44FF      2191          ld    hl,(SysWrd + 68)  ;get (param3)
09E6  EB          2192          ex    de,hl             ;HL=address, DE=value
09E7  73          2193          ld    (hl),e            ;insert low byte
09E8  23          2194          inc   hl
09E9  72          2195          ld    (hl),d            ;insert high byte
09EA  C35B08      2196          jp    Home              ;that's it folks
                  2197  ;
                  2198  ;
                  2199  ; Variable Names and addresses in ExtAdr-Convention.
                  2200  ;   HexChr and IQRSP are not included in the list.
                  2201  ;   For the commands (T, E, U, D, H and A) and the 4 parameters, only the
                  2202  ;   1st variable is included.
                  2203  ;-------------------------------------------------------------------------
                  2204  ;
09ED  F709        2205  Var001: dw    Var002            ;pointer to next
09EF  4D434641    2206          db    'MCFADr'          ;Memory Check Fail address
09F5  00FF        2207          dw    SysWrd            ;address
09F7  010A        2208  Var002: dw    Var003
09F9  494D4F44    2209          db    'IMODEo'          ;Interrupt Mode-1 vector
09FF  02FF        2210          dw    SysWrd + 2
0A01  0B0A        2211  Var003: dw    Var004
0A03  4E4D4956    2212          db    'NMIVEc'          ;Non-maskable Interrupt vector
0A09  04FF        2213          dw    SysWrd + 4
0A0B  150A        2214  Var004: dw    Var005
0A0D  494C4F4F    2215          db    'ILOOPv'          ;Infinite Loop Variable
0A13  06FF        2216          dw    SysWrd + 6
0A15  1D0A        2217  Var005: dw    Var006
0A17  5441536B    2218          db    'TASk'            ;Command TASK address
0A1B  08FF        2219          dw    SysWrd + 8
0A1D  250A        2220  Var006: dw    Var007
0A1F  45584563    2221          db    'EXEc'            ;Command EXECute address
0A23  10FF        2222          dw    SysWrd + 16
0A25  2D0A        2223  Var007: dw    Var008
0A27  444E4C64    2224          db    'DNLd'            ;Command DowNLoaD address
0A2B  18FF        2225          dw    SysWrd + 24
0A2D  350A        2226  Var008: dw    Var009
0A2F  55504C64    2227          db    'UPLd'            ;Command UPLoaD address
0A33  20FF        2228          dw    SysWrd + 32
0A35  3D0A        2229  Var009: dw    Var010
0A37  48454C6F    2230          db    'HELo'            ;Command HELlO address
0A3B  28FF        2231          dw    SysWrd + 40
0A3D  450A        2232  Var010: dw    Var011
0A3F  4D4F4E69    2233          db    'MONi'            ;Command MONItor address
0A43  30FF        2234          dw    SysWrd + 48
0A45  4D0A        2235  Var011: dw    Var012
0A47  41555879    2236          db    'AUXy'            ;Command AUXilliarY address
0A4B  38FF        2237          dw    SysWrd + 56
0A4D  560A        2238  Var012: dw    Var013
0A4F  50415241    2239          db    'PARAm'           ;Parameter-1 address
0A54  40FF        2240          dw    SysWrd + 64
0A56  600A        2241  Var013: dw    Var000
0A58  494D4F44    2242          db    'IMODEz'          ;Interrupt Mode-0 Vector
0A5E  4CFF        2243          dw    SysWrd + 76
0A60  FFFF        2244  Var000: dw    65535
0A62  FF          2245          db    255               ;end of list
                  2246  ;
                  2247  ;
```

```
                2248  ; RETVAR - Returns Value in System Variable.
                2249  ;   varname is the name of the variable, subvar specifies which of the 4
                2250  ;   variables for T, E, U, D, H, A and paramX should be accessed:
                2251  ;     subvar is 0 for C?addr and param1 (address)
                2252  ;               1 for C?byte and param2 (byte count)
                2253  ;               2 for C?pntr and param3 (pointer)
                2254  ;               3 for C?cntr and param4 (counter)
                2255  ;   Not all variables are listed. Unlisted ones can be accessed through
                2256  ;   the subvar-offset. See example below. Actually, SETVAR is an easier
                2257  ;   form of the D)ownload command.
                2258  ;
                2259  ;   EXAMPLE: aRetVar,MCFadr,74,10. will return IQRSP
                2260  ;
                2261  ;   Syntax: aRetVar,varname,subvar.
                2262  ;-------------------------------------------------------------------------
                2263  ;
0A63  A60A      2264  VarRet: dw    DUMMY             ;next routine
0A65  52455456  2265          db    'RETVAr'          ;Return value of a system variable
0A6B  21ED09    2266          ld    hl,Var001         ;start of table, param4 points to buffer
0A6E  CD9308    2267          call  ExtAdr            ;(param1) = PFA
0A71  3D        2268          dec   a                 ;A=1=0: found
0A72  2806      2269          jr    z,VarRe0          ;continue if found
0A74  3E3F      2270          ld    a,'?'             ;unknown
0A76  EF        2271          rst   ConOut            ;issue ?
0A77  C35B08    2272          jp    Home              ;return if not found
0A7A  2A40FF    2273  VarRe0: ld    hl,(SysWrd + 64)  ;get PFA
0A7D  5E        2274          ld    e,(hl)            ;get low byte of address
0A7E  23        2275          inc   hl
0A7F  56        2276          ld    d,(hl)            ;DE=address of variable
0A80  3E02      2277          ld    a,2               ;parameter 2
0A82  CDCB08    2278          call  FetPar            ;(param2) = sub-variable
0A85  2A42FF    2279          ld    hl,(SysWrd + 66)  ;get (param2)
0A88  29        2280          add   hl,hl             ;2 bytes
0A89  19        2281          add   hl,de             ;DE=base address, HL points to variable
0A8A  5E        2282          ld    e,(hl)            ;get low byte
0A8B  23        2283          inc   hl
0A8C  56        2284          ld    d,(hl)            ;get high byte
0A8D  EB        2285          ex    de,hl             ;HL=value
0A8E  2280FF    2286          ld    (SysBuf),hl       ;store it in SysBuf
0A91  CD0305    2287          call  BinDec            ;SysBuf +2 has 5 characters
0A94  2182FF    2288          ld    hl,SysBuf + 2     ;point to 5 byte string
0A97  0605      2289          ld    b,5               ;5 characters
0A99  7E        2290  VarRe1: ld    a,(hl)            ;get 1st character
0A9A  CD2800    2291          call  ConOut            ;send it
0A9D  23        2292          inc   hl                ;point to next
0A9E  10F9      2293          djnz  VarRe1            ;do full string
0AA0  CDC004    2294          call  crlfs             ;send a cr/lf sequence
0AA3  C35B08    2295          jp    Home              ;that's it folks
                2296  ;
                2297  ;
                2298  ; DUMMY - 1st undefined word in list.
                2299  ;-------------------------------------------------------------------------
                2300  ;
0AA6  FFFF      2301  Dummy:  dw    65535             ;First undefined
0AA8  FF        2302          db    255               ;name=255
                2303  ;
                2304  ;
                2305  ;==========================-<951030>-<HoroSoft>-============================
                2306  ;
                2307  end


Errors          0
Range Count     4
```

Source File: UZE201
Symbol Table

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| AllIsr | FF62 | ascbi1 | 0411 | ascbi2 | 042D | ascbi3 | 0430 | ascbi4 | 0432 |
| ascbin | 03FC | AUXY | 0257 | binas1 | 03E7 | binas2 | 03F3 | binasc | 03D6 |
| BinDec | 0503 | blank | 0020 | chkaf | 04D0 | chkaf1 | 04DB | chknbr | 04C9 |
| CmdLst | 0040 | CmdNa1 | 086E | CmdNa2 | 0879 | CmdNa3 | 0881 | CmdNa4 | 088E |
| CmdNam | 0866 | cmdvec | 012F | ColdSt | 0000 | ConIn | 0020 | ConOut | 0028 |
| cools0 | 0086 | cools1 | 0092 | cools2 | 0093 | cools3 | 0099 | coolst | 0080 |
| cr | 000D | crlfp | 02EE | crlfs | 04C0 | ctc0 | 0010 | ctc1 | 0011 |
| ctc2 | 0012 | ctc3 | 0013 | ctcset | 00CC | CTStat | 0018 | dcipr | 00F4 |
| DecBin | 052D | div16 | 04E5 | divhb0 | 0317 | DivHBA | 0316 | dmpas0 | 04A0 |
| dmpas1 | 04AB | dmpas2 | 04AC | dmpasc | 0492 | DNLD | 0181 | dnld1 | 019F |
| dnld2 | 01B2 | dnld3 | 01BA | Dummy | 0AA6 | dumpi0 | 02B3 | dumpi1 | 02B9 |
| dumpi2 | 02C2 | dumpi3 | 02CF | dumpit | 02A0 | dumyap | 0070 | end | 0AA9 |
| EXEC | 0162 | exec1 | 0175 | ExtAdr | 0893 | ExtCm1 | 0849 | ExtCm2 | 0851 |
| ExtCmd | 0830 | ExtEnd | 08C5 | FetPa0 | 08D9 | FetPa1 | 08ED | FetPar | 08CB |
| getas0 | 034C | getas1 | 035D | getasc | 0343 | getpar | 0303 | HELO | 020A |
| helo0 | 0215 | helo1 | 021D | HOME | 085B | im0ret | 0034 | im1ret | 003C |
| im2tbl | 0050 | InStat | 0008 | Invrs | 08B2 | iqdrop | 03A1 | iqdump | 03BF |
| iqpop | 038A | iqpush | 0372 | iqtes1 | 03BD | iqtest | 03B0 | ldela0 | 0825 |
| ldela1 | 0826 | ldelay | 0823 | lf | 000A | lindm0 | 0466 | lindm1 | 0484 |
| lindmp | 0435 | List0 | 098D | List1 | 0993 | List2 | 099C | List3 | 09B0 |
| LISTVW | 0971 | Match | 089E | MONI | 0227 | moni1 | 023E | moni2 | 0247 |
| moni3 | 024A | MulTen | 04DE | nmivec | 0066 | NotFnd | 08C8 | NxtAdr | 08A8 |
| NxtRtn | 0895 | OtStat | 0010 | pioac | 001D | pioad | 001C | piobc | 001F |
| piobd | 001E | PrpUnt | 0324 | pubpro | 068B | pubvec | 0064 | RetIsr | 0141 |
| rstsio | 0074 | RTSGo | 0338 | RTStop | 032D | sdela1 | 081C | sdelay | 081B |
| setpar | 055E | setvar | 056D | sioac | 0019 | sioad | 0018 | siobc | 001B |
| siobd | 001A | siobie | 0102 | sioisr | 0104 | siotbl | 00EE | svinit | 058B |
| swpcmd | 030A | SysBuf | FF80 | SysByt | FF54 | SysWrd | FF00 | TASK | 014F |
| UPLD | 01C5 | upld1 | 01E3 | upld2 | 01F6 | upld3 | 01FF | useend | 0819 |
| uzeve1 | 071A | uzeve2 | 076F | uzeve3 | 07C4 | uzever | 06C5 | Var000 | 0A60 |
| Var001 | 09ED | Var002 | 09F7 | Var003 | 0A01 | Var004 | 0A0B | Var005 | 0A15 |
| Var006 | 0A1D | Var007 | 0A25 | Var008 | 0A2D | Var009 | 0A35 | Var010 | 0A3D |
| Var011 | 0A45 | Var012 | 0A4D | Var013 | 0A56 | VarRe0 | 0A7A | VarRe1 | 0A99 |
| VARRET | 0A63 | VarSe0 | 09CD | VARSET | 09B6 | versi0 | 091D | VersiN | 0927 |
| Versio | 0915 | VoidS1 | 090E | VoidSB | 0905 | warmst | 00AD | wdtcr | 00F1 |
| wdtsr | 00F0 | whowh0 | 026D | whowho | 0260 | | | | |

| Symbol | Value | Defn | References |
|---|---|---|---|
| AllIsr | FF62 | 0176 | 0337 0338 0339 0340 0341 0342 0343 0344 0345 0346 |
| ascbi1 | 0411 | 1290 | 1287 |
| ascbi2 | 042D | 1304 | 1301 |
| ascbi3 | 0430 | 1306 | 1309 |
| ascbi4 | 0432 | 1308 | 1285 1299 |
| ascbin | 03FC | 1280 | 0674 1730 |
| AUXY | 0257 | 0837 | 0328 |
| binas1 | 03E7 | 1259 | 1257 |
| binas2 | 03F3 | 1266 | 1264 |
| binasc | 03D6 | 1249 | 0723 1326 1334 1350 1731 |
| BinDec | 0503 | 1512 | 1732 2287 |
| blank | 0020 | 0135 | 1336 1342 1357 1363 1370 1396 |
| chkaf | 04D0 | 1454 | 1284 1298 1458 1733 |
| chkaf1 | 04DB | 1461 | |
| chknbr | 04C9 | 1440 | 1111 1283 1297 1734 1920 2019 |
| CmdLst | 0040 | 0322 | 0553 |
| CmdNa1 | 086E | 1909 | 1914 1917 1925 |
| CmdNa2 | 0879 | 1915 | 1921 1923 |
| CmdNa3 | 0881 | 1920 | 1911 |
| CmdNa4 | 088E | 1926 | 1919 |
| CmdNam | 0866 | 1904 | 1865 |
| cmdvec | 012F | 0548 | 0528 0531 0534 0537 0540 0543 0546 |
| ColdSt | 0000 | 0161 | |
| ConIn | 0020 | 0165 | 0524 0669 0671 0682 1110 1909 |
| ConOut | 0028 | 0166 | 0726 0728 0736 0769 0935 0942 0971 0973 1425 1427 1874 2098 2133 2138 2140 2142 2177 2271 2291 |
| cools0 | 0086 | 0412 | 0420 |
| cools1 | 0092 | 0421 | 0415 |
| cools2 | 0093 | 0422 | 0426 |
| cools3 | 0099 | 0427 | 0419 |
| coolst | 0080 | 0409 | 0200 |
| cr | 000D | 0134 | 0970 1405 1424 1767 1768 1771 1772 1775 1776 1779 1780 2101 2104 |
| crlfp | 02EE | 0968 | 1726 |
| crlfs | 04C0 | 1423 | 0919 1735 2117 2149 2294 |
| ctc0 | 0010 | 0141 | |
| ctc1 | 0011 | 0142 | |
| ctc2 | 0012 | 0143 | |
| ctc3 | 0013 | 0144 | |
| ctcset | 00CC | 0461 | 0464 |
| CTStat | 0018 | 0164 | 0281 0391 |
| dcipr | 00F4 | 0155 | |
| DecBin | 052D | 1541 | 1736 |
| div16 | 04E5 | 1487 | 0923 1737 |
| divhb0 | 0317 | 1027 | 1030 |
| DivHBA | 0316 | 1026 | 1519 1521 1523 1525 |
| dmpas0 | 04A0 | 1394 | 1404 |
| dmpas1 | 04AB | 1400 | 1397 |
| dmpas2 | 04AC | 1401 | 1399 |
| dmpasc | 0492 | 1386 | 0938 1738 |
| DNLD | 0181 | 0658 | 0324 |
| dnld1 | 019F | 0669 | 0680 |
| dnld2 | 01B2 | 0682 | 0668 0688 |
| dnld3 | 01BA | 0689 | 0681 |
| Dummy | 0AA6 | 2301 | 1788 2264 |
| dumpi0 | 02B3 | 0928 | 0926 |
| dumpi1 | 02B9 | 0930 | 0949 |
| dumpi2 | 02C2 | 0934 | 0937 |
| dumpi3 | 02CF | 0941 | 0944 |
| dumpit | 02A0 | 0916 | 1727 |
| dumyap | 0070 | 0380 | 0450 0475 1628 |
| end | 0AA9 | 2307 | |
| EXEC | 0162 | 0629 | 0323 |
| exec1 | 0175 | 0636 | 0634 |
| ExtAdr | 0893 | 1954 | 1870 2173 2267 |
| ExtCm1 | 0849 | 1876 | 1872 |
| ExtCm2 | 0851 | 1879 | 1877 |
| ExtCmd | 0830 | 1865 | 1653 |
| ExtEnd | 08C5 | 1989 | 1993 |
| FetPa0 | 08D9 | 2017 | 2028 |
| FetPa1 | 08ED | 2029 | 2020 |
| FetPar | 08CB | 2007 | 2184 2186 2278 |
| getas0 | 034C | 1110 | 1118 |
| getas1 | 035D | 1119 | 1112 |
| getasc | 0343 | 1103 | 0998 1739 |
| getpar | 0303 | 0997 | 0597 0630 0659 0661 0711 0713 0806 0815 0821 |

```
upld1       01E3   0721  0732
upld2       01F6   0734  0720 0741
upld3       01FF   0742  0733
useend      0819   1788  1723
uzeve1      071A   1770  1766
uzeve2      076F   1774  1770
uzeve3      07C4   1778  1774
uzever      06C5   1766  1724
Var000      0A60   2244  2241
Var001      09ED   2205  2124 2125 2172 2266
Var002      09F7   2208  2205
Var003      0A01   2211  2208
Var004      0A0B   2214  2211
Var005      0A15   2217  2214
Var006      0A1D   2220  2217
Var007      0A25   2223  2220
Var008      0A2D   2226  2223
Var009      0A35   2229  2226
Var010      0A3D   2232  2229
Var011      0A45   2235  2232
Var012      0A4D   2238  2235
Var013      0A56   2241  2238
VarRe0      0A7A   2273  2269
VarRe1      0A99   2290  2293
VARRET      0A63   2264  2170
VarSe0      09CD   2179  2175
VARSET      09B6   2170  2115
versi0      091D   2095  1754 2100
VersiN      0927   2101  2094
Versio      0915   2092  1869
VoidS1      090E   2058  2060
VoidSB      0905   2053  1890
warmst      00AD   0446  1627 1695 1697 1698 1699 1700 1701 1702 1703 1704
wdtcr       00F1   0154
wdtsr       00F0   0153
whowh0      026D   0865  0874
whowho      0260   0854  1728
```