PREFACE

Well, the work is done now. After 2 years of investigation and typing the DAI personal computer software manual is ready. A countless number of hours resulted in about 500 pages of Assembler listings.

Going through the DAI firmware will show you all the good (and less good) features of the DAI personal computer. Many routines can be used in your own machine language programs. Therefore, entry and exit conditions are added to the routines.

Please don't blame me if you may find some wrong interpretations of parts of the firmware. It is sometimes very difficult to trace the ideas of the one who wrotes the program.
If you may have any comments, I woud be very grateful if you could transmit these to me in writing. It may result in updates of the manual, published seperately or in the DAInamic Newletter.

I would like to acknowledge the DAInamic Users Club for the support with all the information they had available.
I would like to especially thank Mr. Gordon Wassermann for using the results of his investigations of the firmware.
The chapter 'Updates BASIC V1.1' is a result of the compare Jos Schepens did on both BASIC versions.

Jan Boerrigter - September, 1982.

Published by:
'Micro Service'
Fabritiusstraat 15,
6174 RG Sweikhuizen
The Netherlands.

SUMMARY:

1. MEMORY MAP.

2. FIRMWARE MODULES:

C000 - C6BF:	Math. utilities.
C6C0 - C718:	Bank switching.
C719 - D100:	BASIC handler.
D101 - D194:	String handler.
D195 - D23C:	Heap handler.
D23D - D8FA:	I/O handler.
D8FB - D9F4:	Interrupt handler.
D9F5 - DAD3:	Error handler.
DAD4 - DDD0:	Print routines.
DDD1 - DE01:	Encoding service routines.
DE02 - DEB4:	Single/double byte utilities.
DEB5 - 0ECAA:	BASIC execution/run-time module.
0ECAB - 0EFFF:	List handler.
1E000 - 1EE6D:	Math. package.
1EE6E - 1EFFF:	Sound module.
2E000 - 2EBF3:	Screen driving package.
2EBF4 - 2EFFF:	Editor package.
3E000 - 3E9FF:	Encoding package.
3EA00 - 3EFFF:	Utility package.

'Gaps' in these modules are filled with parts of routines from other modules.

3. UPDATES BASIC V1.1.

TABLES:

CBBF:	Strings BASIC commands.
CD8B:	Pointers to strings BASIC commands (for LIST).
CF02:	Pointers to execution routines BASIC commands.
CF86:	Table prefixes unitary operations.
CF91:	Table binary operators.
CFD8:	Table unitary operators.
CFE6:	Table strings BASIC functions.
DA94:	Pointers to strings error messages.
DB6F:	Strings machine messages.
DC1C:	Strings error messages.
0E9F0:	Function indirection table.
0ECFB:	Pointers LIST handling routines.
2E030:	Screen constants.
3E8C5:	ASCII tables.

DAI pc MEMORY MAP

BASIC V1.0/V1.1 Revision 5.1 Date: 18.9.82

INTERRUPT VECTOR ROUTINES: 0000 - 003F

```
0000-07            Interrupt vector routine 0:
                      Used by Utility (LOOK).
0008-0F            Interrupt vector routine 1:
                      Used by Utility and encoding Basic.
                      RST 1 + data: Switch to ROM-bank 3.
0010-17            Interrupt vector routine 2:
                      Used by stack interrupt.
0018-1F            Interrupt vector routine 3:
                      Used by sound interrupt.
0020-27            Interrupt vector routine 4:
                      Used for math. routines.
                      RST 4 + data: Switch to ROM-bank 1.
0028-2F            Interrupt vector routine 5:
                      Used for screen handling routines.
                      RST 5 + data: Switch to ROM-bank 2.
0030-37            Interrupt vector routine 6:
                      Used for keyboard service routines.
0038-3F            Interrupt vector routine 7:
                      Used to flash the cursor.

Interrupt vector      00  NOP
routines:             E5  PUSH H
                      2A  LHLD:
                      ..  ) vector address location
                      ..  ) see (#0062-#0071).
                      E9  PCHL
                      00  NOP
                      00  NOP
```

BANK SWITCHING AREA: 0040 - 0046

```
0040     !POROM:  ) Memory of last outputs to output ports.
         !POR1M:  ) Duplicate of (#FD06).
          POROM:  )
0041/42  RSWK1:  Save PSW during ROM bank switching.
0043/44  RSWK2:  Save HL during ROM bank switching.
0045/46          Spare.
```

UTILITY WORK AREA: 0047 - 0061

0047		Store EI/DI instructions after using LOOK the first time (No clear occurs).
0048/49		High address trace window.
004A/4B		Low address trace window.
004C-4F		Store current instruction if LOOK is used, preceeded by EI. In case of a RST-instruction is stored: RSTx/data x; RST0. In case of an EI instruction is stored: EI, NOP, next instruction.
0050		Flag for Look initialisation: #FF: init. Look, else: #00.
0051/52	IADR:	I: Address current instruction.
0053	AFSAV:	A: Contents A after execution of I.
0054		F: Idem status flags.
0055	BCSAV:	B: Idem B register.
0056		C: Idem C register.
0057	DESAV:	D: Idem D register.
0058		E: Idem E register.
0059	HLSAV:	H: Idem H register.
005A		L: Idem L register.
005B/5C	SPSAV:	S: Idem stackpointer.
005D/5E	PCSAV:	P: Address next instruction to be executed.
005F	TICIM:	M: Current interrupt mask. Duplicate of (#FFF8)
0060		T: Value TICC control word. (#FC after Z2).
0061		G: Value GIC control word. (#1B after Z2).

INTERRUPT VECTOR ADDRESSES: 0062 - 0071

0062/63	I0USA:	Vector address RST 0: set by UT (Z2): 3#EB5D.
0064/65	I1USA:	Vector address RST 1: utility/encode: #C70E.
0066/67	I2USA:	Vector address RST 2: stack interrupt: #D9E2.
0068/69	I3USA:	Vector address RST 3: sound interrupt: #D755.
006A/6B	I4USA:	Vector address RST 4: math. restart: #C6C0.
006C/6D	I5USA:	Vector address RST 5: screen restart: #C6FD.
006E/6F	I6USA:	Vector address RST 6: keyb. int. serv: #D578.
0070/71	I7USA:	Vector address RST 7: clock interrupt: #D9A9.

By changing the vector addresses, other interrupt routines can be used.

SCREEN VARIABLES: 0072 - 00CF

Character mode variables:

0072/73	CURSOR:	Cursor position address.
0074	CURTY:	Cursor type: #00: cursor flashes in colour. #01: cursor alternates between actual character and contents #0075.

```
0075      CURIN:  Cursor information:
                     If type = 0: Mask which is EXOR'ed with
                                  the colour byte for that
                                  character to flash it.
                     If type = 1: Cursor alterates between
                                  actual character and this
                                  information.
0076/77   CURSV:  Contents screen RAM location indicated by
                  the cursor:
                     #0076 contains the colour byte,
                     #0077 contains the data.
0078/79   LNSTR:  Address line mode byte of currently used
                  line of the screen RAM.
007A      LNEND:  Lobyte of end of cursor line.
                  Used to check if end of line is reached.
007B      LCONT:  Number of extended lines.
007C      COLMT:  ) colours for colour    #80+X3
007D              ) registers COLORT      #90+X2
007E              )                       #A0+X1
007F              )                       #B0+X0

          Variables set to describe the current state of
          the screen:

0080/81   SCREEN: Points to first byte of screen RAM (#BFFF).
0082/83   SCTOP:  Points after header (#BFEF).
0084/85   FFB:    First free byte in this mode.
0086/87   GRR:    Points to top of rolled area. Contains the
                  line mode byte of the line where split mode starts.
0088/89   GRE:    Points after end of graphics area.
008A/8B   CHS:    Points to start of character area.
008C/8D   GAE:    Unsplit: End archive area.
          CHE:    Split: After end of character area.
008E/8F   SCE:    End of screen (after trailer).
0090/91   GTE:    End area used splitting mode.
0092/93   GAS:    Unsplit: start archive area.
          GTS:    Split: start temporary save area.
0094/95   GRC:    Number of blobs horizontally in mode.
0096      GRL:    Number of lines of graphics in mode.
0097      GAL:    Number saved lines of graphics.
0098      GXB:    Number of bytes/line this mode.
0099/9A   GREQ:   Previous end of graphics.
009B/9C   CHSO:   Previous start characters:
                  Was split: previous mode byte of 1st text line.
                  Was graphics: Previous last COLORT-byte.
009D      SMODE:  Current screen mode (updated after mode
                  changed):
                     #00 mode 1          #08 mode 5
                     #01 mode 1A         #09 mode 5A
                     #02 mode 2          #0A mode 6
                     #03 mode 2A         #0B mode 6A
                     #04 mode 3
                     #05 mode 3A         #10 during init.
                     #06 mode 4
                     #07 mode 4A         #FF mode 0
                        bits 4-7 are ignored;
                        #0C,0D,0E,0F are inhibited.
          Graphics mode variables (From #00A2-#00B5 also
          used by the EDIT mode):

009E      COLMG:  ) colours for colour #80+X3
009F              ) registers COLORG   #90+X2
00A0              )                    #A0+X1
00A1              )                    #B0+X0
00A2      SCVR:
00A3-AA   SCXBUF: Buffer used to hold contents of an 8 bit
                  field during 16 colour updates.
00AB      SBGOU:  Flags if colour is being carried out to
                  next field.
00AC      SBGOC:  Colour being carried out.
00AD-B4   COLS:   Buffer for impossible requests.

         !Edit variables:

00A2/A3  !EBUFR:  Address start EDIT buffer.
00A4/A5  !EBUFN:  Address end of text in EDIT buffer.
00A6/A7  !EBUFS:  End available space in EDIT buffer.
00A8     !EWINX:  Offset of left side of window.
00A9/AA  !EWINY:  Offset of top of window from start buffer.
00AB     !ECURX:  X-offset of cursor in document (current
                  cursor position in text line).
00AC/AD  !ECURY:  Y-offset of cursor in document (count of
                  current cursor line).
00AE/AF  !CURPT:  Pointer to cursor position in buffer.
00B0/B1  !CURLS:  Pointer to line mode byte of cursor line on screen.
00B2/B3  !CURLB:  Pointer to start of cursor line in buffer.
00B4/B5  !TABTP:  Address tab position table.

          Line drawing variables:

00B5/B6   DELTA:  Amount to add into count.
00B7/B8   RT:     Count.
00B9/BA   COR:    Adjustments for long sectors.
00BB/BC   SECT:   Lower of 2 possible sector lengths.
00BD      SECTC:  Number of sectors.
00BE      TRIM:   Amount to trim off last sector.
00BF      DIRN1:  Set if Y-direction is negative.
00C0      DIRN2:  Set if swap X,Y directions.
00C1      ANIM:   Set if animate in 4 colour mode.
00C2/C3   FCOLR:  Details of colour required.


00C4/C5   ASMKRM: Address memory management routine (#CA01).
                  Checks available RAM space.
00C6/C7   AESTOP: Address emergency stop routine (#CA25).
                  Return-routine for 'Out of space for mode'.
00C8-CF           Spare.
```

MATH. WORKING AREA: 00D0 - 00FF

```
==============================

00D0/D1   EVECT:  Pointer to table with error routines (#C7F2).
00D2/D3   AGETC:  Pointer to input routine (#DDE0).
00D4      MVECA:  Math. chip flag: offset of start HW/SW vector:
                  (offset for RST 4 restart routines):
                     #00  No math. chip.
                     #7B  math. chip present.
```

```
00D5-D8   FPAC:    ) Arithmetic FPT/INT accumulator.
          IAC:     )
00D9      SF:      Subtraction flag.
00DA      OP4:     Operand 4th byte.
00DB      OP3:     Operand 3rd byte.
00DC      OP2:     Operand 2nd byte.
00DD      OP1:     Operand 1st byte.
00DE      EXPDF:   Difference in exponents for last FPT
                   add/sub operations.

          Work area for math. operations:

00DF/E0   FWORK:   ) Also used for data save during stack operations.
          XPRAS:   )
00E1/E2   XPHLS:   )

          FPOLY variables (RAM shared with SQRT):

00E3-E6   XN:      Running power of X (X^K).
00E7-EA   XK:      Power multiplier (X^J).
00EB-EE   SUM:     Running sum.

          SQRT variables (RAM shared with FPOLY):

00E3-E6   F:       Mantissa.
00E7-EA   P:       Polynomial approximation.

          EXP variables (RAM shared with TRIG, INVTRIG):

00EF      SIGN:    Input sign.

          TRIG variables (RAM shared with EXP, INVTRIG):

00EF-F2   FTWRK:   Work location for TAN.

          Inverse TRIG variables (RAM shared with EXP, TRIG):

00EF-F2   FATZX:   Z,X. Used by ATAN, ASIN, ACOS.

          Number input variables:

00E3-E6   ICBWK:   Number to add for each digit.

          Number output variables:

00E3-F1:  DECBUF:  Decimal output buffer.
                       MAXSIG: #0A: Max. possible significant figures.
                       FPTSIG: #06: Number of significant digits for FPT.
00E4      DECBS:   Sign.
00E5      DECBD:   Decimal point.
00E6-F0   DECBF:   Digits, most significant one in 00E6.
00F1      DECBE:   Exponent.
00F2/F3   DECBP:   Buffer pointer.
00F4-FF            Spare variable space.
```

BASIC VARIABLES: 0100 - 02EB

```
          User state:

          Following are saved by soft break: (SFRAME = SYSTOP - SYSBOT)

0100/01   SYSBOT: ) Start of current line. Points to first
          CURRNT: ) byte of line number.
0102/03   BRKPT:  Start of current command.
0104/05   LOPVAR: Points to current loop variable. Points to
                  position of variable in symbol table.
                     #00 if no running loop.
0106      LSTPF:  Flag for integer/fpt loop and
                  implicit/explicit step.
                     bit 0: 0 = implicit step.
                            1 = expicit step.
                     bit 7: 0 = FPT loop variable.
                            1 = INT loop variable.
0107-0A   LSTEP:  Step value if explicit.
010B-0E   LCOUNT: Loop iteration count.
010F-10   LOPPT:  Pointer to start address loop.
0111/12   LOPLN:  Pointer to start loop line.
0113/14   STKGOS: Stack level at last GOSUB.
                     #00 if no active call.
0115      SYSTOP: )
         (STRFL:  ) Trace/step flag together)
          TRAFL:  ) Trace flag (#FF if set).
0116      STEPF:  Step flag (#FF if set).
0117      RDIPF:  Flag set while running input (set: #FF).
0118      RUNF:   Flag set while running program.
                  (Previous 2 bytes must be consecutive)

          Runtime scratch area:

0119/1A   GSNWK:  Scratch area for GOSUB/NEXT (2 bytes).
                  Points to destination address last GOSUB.
          LISW1:  Startaddress of listed area.
          COLWK:  Scratch area for SCOLG, SCOLT (4 bytes).
                  Contains last selected COLORT/COLORG values.
011B/1C   LISW2:  End address listed area.

          Save area for restart on error:

011D/1E   ERSSP:  Stack pointer.
011F-21
0122      ERSFL:  Set if encoding a stored line (set: #01).

          Data/read variables:

0123      DATAC:  Offset of next character to encode.
0124/25   DATAP:  Pointer to address current data line.
         !DATAQ:  Pointer after current data line.
0126      CONFL:  Set if there is a suspended program (set: #01).
0127/28   STACK:  Current base stack level.

          Scratch location for expression/function evaluation.

0129-2C   WORKE:  Scratch area. Contains also the argument A of
                  the last software random RND(A).
```

```
        Random number kernel:

012D-30 RNUM:   Random number kernel.
        RNDLY:  Random number delay count (1 byte).

        Output switching:

0131    OTSW:   #00  output to screen + RS232.
                #01  output to screen only.
                #02  output to edit buffer.
                #03  output via DOUTC.

        Encoding input source switching:

0132/33 EFEPT:  Encoding input pointer. Points to start-
                address of Basic-line just being encoded.
0134    EFECT:  Encoded input count. Counts length of line.
0135    EFSW:   Encoded input switching:
                    #00  Input from keyboard/DINC.
                    #01  Input from string.
                    #02  From edit buffer to program area.

        Variables used during expression encoding
        (could overlap with runtime variables):

0136    TYPE:   Type of latest expression or item:
                    #00  FPT
                    #10  INT
                    #20  STR
                    #30  Boolean
0137    RGTOP:  Latest priority operator:
                    #00  no operation       #6A  IOR
                    #38  AND                #6C  IXOR
                    #39  OR                 #8D  SHL
                    #50  >=                 #8E  SHR
                    #51  >                  #A0  +
                    #52  <>                 #A1  -
                    #53  <=                 #C2  /
                    #54  <                  #C3  *
                    #55  =                  #CF  MOD
                    #69  IAND               #E4  ^
0138    OLDOP:  Old priority operator.
0139/3A HOPPT:  Pointer to place in encoded input buffer
                for next operator.
013B/3C RGTPT:  Pointer to place in encoded input buffer
                of operand latest operator.

        Mask to select cassette 1 or 2:

013D    CASSL:  #10  Cassette 1 activated.
                #20  Cassette 2 activated.

        Encoded input buffer:

013E-BD EBUF:   128 bytes buffer. Also used by
                utility.
```

Interrupt handler variables:

01BE/BF TIMER: Timer location. Also used in WAIT TIME.

01C0 CTIMR: Cursor clock. Used for cursor flashing.
CTIMV: #0F: Flash time in 20 ms units.
If #00, cursor flashes.

01C1 KBXCT: Extend keyboard scan time counter. When #00, keyboard scan will be performed.
KBXCK: #02: Keyboard scan time (16 ms units). Also used by RAND routine.

Sound control block storage:

01C2-CF: Sound control block 0.
SCBL: Length of a sound block (14 bytes).

01C2 SCB0: Elapsed count of current volume:
#FF: channel off.
#FE: current volume forever.

01C3/C4 Pointer to required count at this volume in envelope table.

01C5/C6 Pointer to start envelope table being used.

01C7 Sound-volume *8. Multiplier for volume, between 0 and 16, shifted 3 places left.

01C8 Basic volume at this moment, calculated from sound-volume and present envelope volume.

01C9 Counter for tremolo. 0 if no tremolo.

01CA Actual volume, calculated from volume and tremolo fluctuations.

01CB Glissando flag:
#00 Endperiod reached.
#01 Set frequency.
#02 Endperiod not reached.

01CC/CD Current period of output.

01CE/CF Required final period of output.

01D0-DB SCB1: Sound control block 1 (see SCB0).

01DE-EB SCB2: Sound control block 2 (idem).

01EC-F4 NCB: Noise control block.
NCBL: Length of noise block (9 bytes).
The noise control block is identical to the sound control block, but without period-values and tremolo.

Envelope storage:

01F5-
-0274 ENVST: Envelope storage (128 bytes).
ENVLL: #40: Number of bytes/envelope
NUMENV: #02: Number of envelopes.
Two envelope tables of each 64 bytes:
#01F5-#0234 and #0235-#0274.

Type storage:

0275-
-28E IMPTAB: Implicit type table.

#0275	A	#027C	H	#0283	O	#028A	V
#0276	B	#027D	I	#0284	P	#028B	W
#0277	C	#027E	J	#0285	Q	#028C	X
#0278	D	#027F	K	#0286	R	#028D	Y
#0279	E	#0280	L	#0287	S	#028E	Z
#027A	F	#0281	M	#0288	T		
#027B	G	#0282	N	#0289	U		

```
028F      IMPTYP: Default number type. Selected by IMP command.
                      #00  FPT
                      #10  INT
                      #20  STR
0290      REQTYP: Required number type for present operation.
                      #00  FPT
                      #10  INT
                      #20  STR
                      #30  Variable name argument
                      #40  Array without arguments


0291/92   DATAQ:  Pointer to begin current data line.
0293      RNDLY:
0294      POROM:  Duplicate of (#FD04).
0295      POR1M:  Duplicate of (#FD05).
0296      INSW:   Input switching:
                      If #00, input from keyboard.
                      If <>#00, input from DINC (Default: RS232).
0297-9A           Spare.

          Heap/text buffer/symbol table pointers:

029B/9C   HEAP:   Start address of HEAP.
029D/9E   HSIZE:  Size of HEAP.
                      HSIZD: #100: Default size.
029F/A0   TXTBGN: Start address of text buffer.
02A1/A2   TXTUSE: End text buffer and.
          STBBGN: Start symbol table.
02A3/A4   STBUSE: End of symbol table.
02A5/A6   SCRBOT: Bottom screen RAM area (48K):
                      mode 0:        #B350
                      mode 1/2(A):   #B7A0
                      mode 3/4(A):   #A65C
                      mode 5/6(A):   #63B8

          Keyboard variables + constants:

02A7/A8   KBTPT:  Pointer to table with ASCII-codes.
02A9-B0   MAP1:   Latest scan of keys (key-codes).
                  (row 0 in #02A9, row 7 in #02B0)
   02AF   RPLOC:  Byte containing REPT key.
                      RPMSK: #20: Rept key bit.
                      BRSEL: #40: Column select mask for BREAK.
                      BRMSK: #40: Break key bit.
   02B0   SHLOC:  Byte containing SHIFT.
                      SHMSK: #40: Shift key bit.
02B1-B8   MAP2:   Previous scanning of keyboard.
02B9      KNSCAN: Set to scan for BREAK only. When (#02B9)
                  is #FF: scan for BREAK only.
02BA-BD   KLIND:  4 byte circular buffer to store the ASCII
                  values for keys pressed.
                      KBLEN/KEYL: #04: length rollover buffer.
02BE/BF   KLIIN:  Next position for input to KLIND.
02C0/C1   KLIOU:  Next position for output from KLIND.
02C2      RPCNT:  Count for REPT. #01 if REPT is not
                  pressed. Else it is used as timer for the
                  repeat function.
02C3      SHLK:   Set to #FF if CTRL is pressed to
                  invert SHIFT. Else #00. Used to
                  calculate the offset for the ASCII code
                  table.
```

```
02C4      KBRFL:  Break flag. #FF indicates BREAK pressed
                  (Only if suspended program). If BREAK is pressed,
                  #02C4 counts from 00 to #0F before stopping
                  the program.

          Data/cassette switching vectors:

          Copy of ROM (#D7A4 - #D7CA) for cassette and RS232.
          Can be loaded with other I/O vectors.

02C5-EB   IOVEC:  02C5  WOPEN:  C3 B8 D2  JMP: D2B8
                  02C8  WBLK:   C3 F1 D2  JMP: D2F1
                  02CB  WCLOSE: C3 27 D4  JMP: D427
                  02CE  ROPEN:  C3 25 D3  JMP: D325
                  02D1  RBLK:   C3 40 D3  JMP: D340
                  02D4  RCLOSE: C3 45 D4  JMP: D445
                  02D7  MBLK:   C3 A2 D3  JMP: D3A2
                  02DA  RESET:  C9 00 00  RET
                  02DD  DOUTC:  C9 00 00  RET
                  02E0  DINC:   C3 B4 DD  JMP: DDB4
                  02E3          C9 00 00  RET
                  02E6  TAPSL:  24 24     Tape speed leader.
                  02E8  TAPSD:  24 3C     Tape speed data.
                  02EA  TAPST:  24 18     Tape speed trailer.
```

HEAP, PROGRAM AREA, SCREEN RAM: 02EC - BFFF

```
02EC-  (RAM:     HEAP (Strings + arrays)   - See (#029B/9C).
 -BFFF (VAREND:  Program (compiled Basic)  - See (#029F/A0).
       (VARLAST: Symbol table              - See (#02A1/A2).
                 Not used RAM              - See (#02A3/A4).
                 Screen RAM                - See (#02A5/A6).
```

ROM AND CPU AREA: C000 - FBFF

```
C000-            24K ROM:
  -EFFF             #C000-#DFFF: 8K non-switched ROM.
        VECA:       #E000-#EFFF: 4 banks of each 4K ROM.
                                 (switchable).


F000-            Can be used for ROM extension (reading only).
  -F7FF          Is already completely used by Memocom MDCR-D.


FB00-            Microcomputer stack.
  -FBFF          Incl. vector for MDS jump instructions.
                    #F800  SRBOT    Bottom of stack RAM.
                    #F900  STTOP    Top of stack RAM.
```

I/O DEVICE ADDRESSES: F900 - FFFF

```
F900-              Spare I/O device addresses.
  -FAFF            (Not wired on pC board).
```

MATH. CHIP AMD 9511: FB00 - FBFF

```
FB00      MTHAD:   ) Data math.chip.
          MDATA:   )
FB02      MCOMD:   ) Command + status.
          MSTATUS:)
```

AMD9511 operator and status bytes:

ODADD:	#2C	Int addition	OFADD:	#10	Fpt addition
ODSUB:	#2D	Int subtract	OFSUB:	#11	Fpt subtract
ODMUL:	#2E	Int multiply	OFMUL:	#12	Fpt multiply
ODDIV:	#2F	Int division	OFDIV:	#13	Fpt division
OSQRT:	#01	Square root	OFIXD:	#1E	Fix
OSIN:	#02	Sine	OFLTD:	#1C	Float
OCOS:	#03	Cosine	OCHSD:	#34	Change sign int
OTAN:	#04	Tangent	OCHSF:	#15	Change sign fpt
OASIN:	#05	Arc sine	OPTOD:	#37	Push int/fpt
OACOS:	#06	Arc cosine	OPOPD:	#38	Pop int/fpt
OATAN:	#07	Arc tangent			
OLOG:	#08	Log base 10			
OLN:	#09	Log base e	MBUSY:	#80	Busy status bit
OEXP:	#0A	Expotential	MERRB:	#1E	All error bits
OPWR:	#0B	X^Y	MZERO:	#20	Top of stack

PROGRAMMABLE INTERVAL TIMER 8253: FC00 - FCFF

Used for sound generator. 3 independent 16 bits down counters with programmable counter modes.

```
FC00/01  SNDAD:   )
         SNDO:    ) Counter 0 (oscillator channel 0).
        !PDLCH:   Used as counter for paddle operations.
FC02/03  SND1:    Counter 1 (oscillator channel 1).
FC04/05  SND2:    Counter 2 (oscillator channel 2).
                  (16 bit data; LSB first)
FC06     SNDC:    Command 8253. To be loaded prior to freq.
                  selection with resp. #36, #76 and #B6.
                  Command word format:
                  bit 0    : 0   binairy counter 16 digits.
                             1   BCD counter (4 decades).
                      3,2,1: 000 mode 0: Interrupt on end count.
                             001 mode 1: Programmable one shot.
                             x10 mode 2: Rate generator.
                             x11 mode 3: Sq.wave rate generator.
                             100 mode 4: SW trig. strobe.
                             101 mode 5: HW trig. strobe.
```

```
                    5,4  : 00  Counter latch operation.
                           01  Read/load MSB only.
                           10  Read/load LSB only.
                           11  Read/load LSB first, then
                               MSB.
                    7,6  : 00  Select counter 0.
                           01  Select counter 1.
                           10  Select counter 2.
                           11  Illegal.
                 Several control words:
                    COFIX: #00  Fix count on channel 0.
                    COMO:  #30  Chan.0, mode 0, 2 byte op.
                    COM1:  #32  Chan.0, mode 1, 2 byte op.
                    COM3:  #36  Chan.0, mode 3, 2 byte op.
                    C1M3:  #76  Chan.1, mode 3, 2 byte op.
                    C2M3:  #B6  Chan.2, mode 3, 2 byte op.
```

```
           DISCRETE I/O DEVICE ADDRESSES: FD00 - FDFF
           ========================================

FD00       PORI:    IN (1)  bit 0: -
                                1: -
                                2: PIPGE: Page signal
                                3: PIDTR: Serial output ready
                                4: PIBU1: Button on paddle 1
                                          (1 = closed)
                                5: PIBU2: Button on paddle 2
                                          (1 = closed)
                                6: PIRPI: Random data
                                7: PICAI: Cassette input data

FD01       PDLST:   OUT (3) Single pulse used to trigger
                            paddle timer circuit.

FD04       PORO:    OUT (2) bit 0 - 3: volume osc. channel 0
                                4 - 7: volume osc. channel 1
FD05       POR1:    OUT (2) bit 0 - 3: volume osc. channel 2
                                4 - 7: volume random noise
                                       generator.
FD06       PORO:    OUT (3) bit 0: POCAS: Cassette data output
                              1,2: PDLMSK: Paddle select
                                3: POPNA:  Paddle enable
                                4: POCM1:  Cassette 1 motor
                                           control.(0 = run)
                                5: POCM2:  Cassette 2 motor
                                           control.(0 = run)
                              7,6:         ROM bank switching:
                                              00  bank 0
                                              01  bank 1
                                              10  bank 2
                                              11  bank 3
```

PROGR. PERIPHERAL INTERFACE 8255 : FE00 - FEFF

Used for DCE-bus (GIC Controller).

FE00 GIC: (1) I/O port A
FE01 (1) I/O port B
FE02 (1) I/O port C
FE03 (6) Command word 8255:

Contr.	PA	PCH	PCL	PB	(mode 0)
#80	out	out	out	out	RWMOP
#81	out	out	in	out	
#82	out	out	out	in	
#83	out	out·	in	in	
#88	out	in	out	out	
#89	out	in	in	out	
#8A	out	in	out	in	
#8B	out	in	in	in	
#90	in	out	out	out	RWMIP
#91	in	out	in	out	
#92	in	out	out	in	
#93	in	out	in	in	
#98	in	in	out	out	
#99	in	in	in	out	
#9A	in	in	out	in	
#9B	in	in	in	in	

TICC: TIMER + INTERRUPT CONTROLLER 5501: FF00-FFFF

FFF0 (4) Serial input buffer. Contains the last character received on the RS232 interface.

FFF1 (4) Keyboard input port. Bottom 7 bits are data input from the keyboard. Bit 7 is the IN7 line from the DCE-bus and is attached to the page-blanking signal for the TV. Every 20 ms. an impulse is present.

FFF2 (2) Interrupt address register:

```
bits 5,4,3: Number of pending
            interrupt.
     7,6  : )
     2,1,0: ) always '1'
```

FFF3 (4) Status register:

bit 0: Frame error. Set by a BREAK on the RS232 input.

1: Overrun error. Set if a character has been received but not taken by the CPU.

2: Serial input. Set if no data is received.

3: Receive buffer loaded. Set if a character has been received.

4: Transmit buffer empty. Set if RS232 output is ready to accept another character.

5: Interrupt pending. Set if one or more of the enabled interrupts has occured.

```
                    6: Full bit detected. Set if the first
                       data bit of an incoming character
                       has been detected.
                    7: Start bit detected. Set if the start
                       bit of an incoming character has been
                       detected.
 FFF4        (2) Command register:
                bit 0: TICC reset.
                    1: Send Break. If set, the serial output
                       is high impedance.
                    2: Interrupt 7 select. A '1' selects IN7
                       of the DCE-bus, a '0' selects Timer 5.
                    3: Interrupt acknowledge enable.
                       A '1' enables TICC to accept a INTA
                       signal from the CPU.
                    4 - 7: Always 0.
 FFF5        (6) Communications rate register:
                bit 0:   110 baud
                    1:   150 baud
                    2:   300 baud
                    3:  1200 baud
                    4:  2400 baud
                    5:  4800 baud
                    6:  9600 baud
                    7:  1 - one stop bit
                        0 - two stop bits
 FFF6        (6) Serial output buffer. Write byte to this
                location to send it on the RS232 output.
                Use only when #FFF3-bit 4 is high.
 FFF7        (7) Keyboard output port. Data output to scan
                keyboard.
 FFF8        (2) Interrupt mask register:
                bit 0: timer 1 has expired (UTIM).
                    1: timer 2 has expired.
                    2: External interrupt (STKIM).
                    3: Timer 3 has expired (SNDIM).
                    4: Serial receiver loaded.
                    5: Serial transmitter empty.
                    6: Timer 4 has expired (KBIM).
                    7: Timer 5 has expired or IN7 (CLKIM).
                (react only on low-high transition)
 FFF9        (2) UTIAD:   Timer 1 address (UT).
 FFFA        (1)          Timer 2 address.
 FFFB        (2) SNDIAD:  Timer 3 address (sound).
 FFFC        (2) KBIAD:   Timer 4 address (keyboard).
 FFFD        (1)          Timer 5 address.
 FFFE            not used.
 FFFF            not used.
```

NOTES:

(1) Read and write allowed.
(2) Reading allowed. Writing too, but may be overwritten by BASIC program.
(3) No writing allowed.
(4) Reading allowed, writing not.
(5) Should not be accessed.
(6) Writing allowed, reading not.
(7) Reading not allowed, writing is harmless but useless; keyboard scanner will overwrite it.

REMARKS:

ADDRESSES FB00 - FFFF:

The highest byte of the address is used for the chip select signal CS of the peripheral equipment 8253, 8255, 5501 etc. The lowest byte is used to address the several registers of the peripheral. Its high nibble does not have any value. So addresses in this range can be read as FBx0 - FFxF, in which x is a don't care.