

1. General

1.1 Compatibility

The 'Interface 1bis' is software compatible with the Sinclair 'ZX INTERFACE 1' at BASIC command as well as at 'hook-code' level

Necessary pre-conditions

- Same mechanism to extend the BASIC interpreter: paging a 'shadow' ROM in place of the BASIC ROM, whenever a syntax error is encountered
- Same 'extended BASIC' syntax
- Same system variables
- Same mechanism to access shadow ROM routines: 'Hook codes'
- Same data structures for handling sequential files: 'M channel' and network communication: 'N channel'

1.2 Memory layout

When activated, the interface disables the internal PROM of the ZX Spectrum and pages in its own operating system (OpSys), which resides in two contiguous 16 KB NVSRAM banks with the following layout:

Bank	Offset	Size	Address	Write protected
BASIC ROM	#0000	#4000	#0000	Yes
Shadow ROM	#4000	#2E00	#0000	Yes
Work RAM	#6E00	#0200	#2E00	No
Buffers	#7000	#1000	#3000	No

1.2.1 The 'BASIC ROM'

is a slightly modified copy of the ZX Spectrum 48k ROM, the scope of the changes being restricted to:

- enabling software-controlled memory paging by means of input/output operations to dedicated ports
- trapping the calls to the tape routines, to handle .TAP files,
- modified NMI handling, to allow the creation of (.Z80) snapshots, and optionally:
- integration of an ESC/P printer driver in the ZX Spectrum BASIC
- fixing some known ZX Spectrum 48k ROM bugs.

1.2.2 The 'shadow ROM'

The shadow ROM is fully compatible with the 8 KB ROM of the original Sinclair 'ZX INTERFACE 1' at BASIC command and 'hook code' level

- Following hook codes are not implemented:

Code	Function
#33	Read next header

1.2.3 The 'work RAM'

- The 'work RAM' is structured as below:

Address	Block	Bytes
#2E00	Variables	128
#2E80	Internal stack	64
#2EC0	Printer buffer	64
#2F00	Page buffer	256

- The internal stack is used when handling (.Z80) snapshots or loading (.TAP) tape files.

1.2.4 Buffers

- There are seven 512 bytes sector buffers, a 256 bytes 'application data' area and a 'current directory table'

2. The extended BASIC

2.1 Syntax

All 'ZX INTERFACE 1' extended BASIC statements are accepted in their original format, while some syntax enhancements are implemented.

```
2.1.1 CAT [#<str>;]{*}[<dev>;][<drv>][;<fil>|<pth>] [ABS|NOT|LN]
2.1.2 CLEAR #
2.1.3 CLOSE #<str>
2.1.4 CLS #
2.1.5 ERASE <spe>|#<hdl>
2.1.6 FORMAT [#<csz>;]{*}[<dev>;][<drv>] [NOT]
2.1.7 FORMAT [#<rsd>;]{*}[<dev>;] 0 [NOT]
2.1.8 FORMAT "B";<brt>
2.1.9 FORMAT "N";<sta>;<sid>
2.1.10 INKEY$ #<str>
2.1.11 INPUT #<str>,<var>
2.1.12 LOAD <chn>|#<hdl> [<opt>]
2.1.13 LOAD [[*<dev>;][<drv>];<pth>
2.1.14 LOAD STOP
2.1.15 MERGE <chn>|#<hdl>
2.1.16 MOVE <spe>|<nfc>|#<str> TO|AT|OVER|IN|MERGE <spe>|<nfc>|#<str>
2.1.17 MOVE #<str>|#<hdl> VAL
2.1.18 MOVE #<str>|#<hdl> POINT [<pnt>]
2.1.19 OPEN #<str>;{*}[<dev>;][<drv>];<fil> [IN|OUT|OVER|RND]
2.1.20 OPEN #<str>;<nfc>
2.1.21 PRINT #<str>;<exp>
2.1.22 SAVE <chn>|#<hdl> [<opt>]
2.1.23 SAVE [[*<dev>;][<drv>];<pth>
2.1.24 SAVE STOP
2.1.25 VERIFY <chn>|#<hdl> [<opt>]
2.1.26 VERIFY STOP
```

where:

- <str> = Stream (0-15)
- <hdl> = Handle (0-15)
- <chn> = Channel
 - = [*<dev>;][<drv>];<fil>
- <nfc> = Non-file channel
 - = {"B"|"T"
 - = {"N";<sta>
- <spe> = Specifier
 - = {*}[<dev>;][<drv>];<fil>|<pth>
- <dev> = Device specifier
 - = <typ>[<sta>]
- <typ> = Device type literal
 - = "M" - flash "M"icrodrive
 - = "R" - "R"AM drive
 - = "V" - ser"V"er drive
- <sta> = Station (device) number (0-7)
 - = 0 - Local
 - = 1..7 - over IP network
- <drv> = Drive number (1-255)
- <fil> = File (1-254 characters)
 - = [<pth>]<nam>[.<fty>]
- <dir> = Directory (1-254 characters)
 - = <nam>/
- <pth> = Path (1-254 characters)
 - = [/<dir><dir>..<dir>
- <nam> = Name (1-254 characters)
- <fty> = File type literal (1 character)
 - defined at 3.4
- <opt> = SAVE, LOAD or VERIFY options
 - = LINE <lin>
 - = DATA <ary>[<len>][<pos>]
 - = CODE [<add>[,<len>{,<pos>}]]
 - = SCREEN\$
 - = BIN [<add>[,<len>[,<pos>]]]
- <lin> = Auto-run line number (0-9999)
- <ary> = Array name
- <add> = Memory block address (0-65535)
- <len> = Memory block length (0-65535)
- <pos> = File pointer
 - = <rec>[,<pos>]
- <rec> = Record number (0-32767)
- <pos> = Position within a record (0-511)
- <var> = BASIC variable
- <exp> = BASIC expression
- <csz> = Allocation unit in sectors/cluster (2,4,8,16)
- <rsd> = Number of reserved drives (1 - 127)
- <brt> = Baud rate in bit/s (0-65535)
- <sid> = Name or IP address (1-254 characters)

- Syntax elements in square brackets are optional
- Syntax elements in curly brackets are accepted but not used
- Alternative syntax elements are separated by a vertical bar
- A (file) 'handle' is a stream opened to a file, using the option RND
- The position within a record: <pos> may be specified in the range (0-65535), because the pointer is always automatically normalized:
 $\text{<rec>} = \text{<rec>} + \text{int}(\text{<pos>} / 512)$
 $\text{<pos>} = \text{mod}(\text{<pos>}, 512)$
- A leading '*' not followed by a device type literal stands for: "M"
- A leading '@' not followed by a device type literal stands for: "E"
- A leading '!' stands for: "R";1 and a leading '\$' for: "V";1

2.2 Devices and Channels

2.2.1 Storage devices

2.2.1.1	"M"	(0) SD card	block device
2.2.1.2	"E"	(1) Tape player EAR port	block device
2.2.1.3	"R"	(2) RAM drive	block device
2.2.1.4	"V"	(3) Server	file device

2.2.2 Communication devices

2.2.2.1	"B"	(4) Asynchronous serial port
2.2.2.2	"N"	(5) 10Mbit/s Ethernet

2.2.3 Channels

2.2.3.1 The 'Microdrive' channel: "M"

- The M channel provides buffered character input/output from/to the supported storage devices
- It is compatible with the 'Microdrive' channel of the original 'ZX INTERFACE 1', having the same descriptor structure

2.2.3.2 The 'RS-232 Interface' channels: "B" and "T"

- These channels are by default implemented as output-only. Any input operation produces an error report
- Channel B sends binary data directly to the printer spooler while channel "T" behaves identically to channel "P"
- When a suitable cable is plugged into the mouse and joystick sockets channel B can be used for duplex asynchronous serial communication.
- Only the OPEN#, CLOSE# and FORMAT commands are implemented, while SAVE, LOAD, VERIFY and MERGE are not.

2.2.3.3 The 'Local Area Network' channel: "N"

- The N channel provides buffered character input/output over a network
- It is compatible with the original 'ZX INTERFACE 1' implementation
- Only the OPEN#, CLOSE# and FORMAT commands are implemented, while SAVE, LOAD, VERIFY and MERGE are not.

2.2.3.4 The 'Handle' channel: H

- An 'H channel' is created by opening a stream to a file, using the option RND. Its descriptor is identical to bytes 0-30 of the M channel descriptor
- Such a stream can be used as a 'handle' to specify the associated file in LOAD, SAVE, VERY, MERGE and ERASE statements

2.2.3.5 The 'NULL' channel: U

- Provides no input and discards any output

2.3 File names

- Full names may be composed of segments, separated by "/". The last segment represents the actual filename, while all the other make up the path. For a block device, only the first 10 characters of a segment are significant. The total length of the path is limited to 254 characters
- A name ending with a "/" represents a directory name
- Filenames may have a trailing 'file type literal', separated by a ".", as an extension
- A leading "/" stands for the root directory of the disk and a "../" for the parent directory
- For a file device, "/A/", "/C/".."/Z/" represent the drives A, C .. Z of the server. The alternative form "a:/".. is also accepted.
- When not creating a new file, the wild cards "?" (standing for "any character") and "*" (standing for "any number of characters") may be used in regular names, except for those of channels and handles, but not in directory or path names
- Filenames are case-insensitive

2.4 File types

2.4.1 'BASIC' files

Type	Literal	Description	Extension
0	P	BASIC program	ZZP
1	N	Number array	ZZN
2	A	String array	ZZA
3	C	CODE block	ZZC

- To allow access via the SAVE, LOAD and VERIFY commands these files contain a 9-byte header, with the following structure:

- 0 File type (0-3)
- 1-2 File length (excluding the header)
- 3-4 Loading address (Code)
- 5-6 Length of program only (Program)
- Array name (Numeric or String)
- 7-8 Start line (Program)

2.4.2 'Regular' files

Type	Literal	Description	Extension
4	F	PRINT file	ZZF
5	E	Text file	ZZE
6	K	Backup file	ZZK
7	B	Binary file	ZZB

- The maximum length of a regular file is 16 MByte (32768 records of 512 bytes each).

2.4.2.1 PRINT file (type 4)

- PRINT files are implemented as in the original ZX INTERFACE 1 extended BASIC, to be accessed via the OPEN#, PRINT#, INKEY\$# and INPUT# commands

2.4.2.2 Text file (type 5)

- A 'Text' file contains no control characters besides CR and LF and has all BASIC tokens expanded
- When writing (PRINT#) to a stream opened to a 'text file', a LF is automatically inserted after each CR
- When reading (INPUT#) from a stream opened to a 'text file', any LF following a CR is discarded

2.4.2.3 Backup file (type 6)

- 'Backup' files are copies of files of any other type

2.4.2.4 Binary file (Type 7)

- 'Binary' files have no specific structure

2.4.3 'Emulator' files

Type	Literal	Description	Extension
8	S	Screen dump	SCR
9	X	'ZX Tape' file	TZX
10	T	Tape file	TAP
11	Z	'Z80' snapshot	Z80

2.4.3.1 Screen dump (type 8)

- A 'Screen dump' represents the contents of the video RAM, having the default loading address of: #4000 and the default length of: #1B00,
- Screen dumps are loaded or saved specifying the file type by means of the filename extension '.s'

2.4.3.2 'ZX Tape' file (type 9)

- A file in 'TZX' format, containing only type #10, #2A, #30 and #32 blocks, can be assigned as an 'input tape', via a LOAD command, specifying the file type by means of the filename extension '.x'

2.4.3.3 'Tape' file (type 10)

- A 'tape' file is opened for input or output via the LOAD or respectively SAVE statement, specifying the file type by means of the filename extension '.t', after which, all BASIC tape input or output is redirected to the specified file, until the end of the 'input tape' is reached, the length of the 'output tape' exceeds 16 MB or the file is closed, using the LOAD or respectively SAVE command with the option: STOP
- A reset or even a power-off does not close the tape files.
- Opening the 'input tape' to a non-existing file will generate the error report "File not found"
- Opening the 'output tape' to a non-existing file will create the file
- Opening the 'output tape' to an existing file will append to the file
- The 'input tape' and 'output tape' can be simultaneously opened to the same file, but the blocks appended after the 'input tape' was opened, will not be accessible until the 'input tape' is closed and re-opened
- Opening the 'input tape' using the extension '.T' (or '.X') will immediately perform the equivalent of NEW, followed by LOAD"".

- When device "E" is specified, a tape file can be loaded from the audio signal produced by the interface, via the Spectrum's 'EAR' input, if a suitable cable is connected.

2.4.3.4 (.Z80) Snapshot file (type 11)

- (.Z80) snapshot files are launched using the LOAD command, specifying the file type by means of the filename extension '.z'
- After loading a snapshot with the extension '.Z' (capital) the interface will switch to the 'ON - inactive' state
- To create a version 1.45 48k snapshot, a file must be first opened using the SAVE command specifying the file type by means of the filename extension '.z', after which, generating a NMI saves the snapshot and if CAPS SHIFT is not pressed, closes the corresponding file
- Closing can also be forced using the VERIFY command with the option: STOP, but the resulting file will have no usable content
- A reset or even a power-off does not close the snapshot file
- If the extension '.Z' (capital) is specified, a version 3.05 snapshot is created, for 'hardware mode' 128k, if possible, or 48k otherwise

2.4.3.5 Applying 'POKES'

- If bit 0 of (AX_FLG) is set, (BUFF_P) is expected to contain a list of 'POKES', in the following format:

Offset	Description
0	Flag - if #FF marks end of list
1-2	Address
3	Data

to be automatically applied, after loading a snapshot. For tape files, the POKES are applied only when a NMI is triggered, while the SYMBOL SHIFT key is being pressed

2.4.4 Reserved file types

Type	Literal	Description	Extension
12		Reserved	
13		Reserved	
14		Reserved	
15	Y	Any type	*

- Type numbers: 12,13 and 14 are reserved
- Type number 15 is the 'type wild card', standing for "Any type"

2.4.5 Directories (type 16)

- Directories are special files, accessed via the commands LOAD, standing for 'change', SAVE, standing for 'create' and DELETE

2.5 Error messages

The error messages are the same as those of the original ZX INTERFACE 1, except for the following:

- #07: "Missing name" not used
- #08: "Missing station number" not used
- #09: "Missing drive number" not used
- #0A: "Missing baud rate" replaced by "Communication error"
- #0B: "Header mismatch error" replaced by "Directory in use"
- #13: "Hook code error" replaced by "File exists"
- #15: "MERGE error" replaced by "Invalid path"
- #17: "Wrong file type" not used

2.6 Other syntax issues

2.6.1 Default values

- The default values for the device literal <dev>, volume literal <sta> and drive number <drv> are the ones last specified in a statement
- For statement 2.1.1, the default value of <str> is: 2
- For statement 2.1.6, the default value of <csz> is the one stored on the media when the drive was last formatted, or otherwise: 8

2.6.2 The CAT command

- The file list produced by the CAT statement has following layout:
 - Column 1-10 Filename
 - 12 File type literal
 - 14-21 File length in bytes
 - 23-27 Auto start line (Program)
 - Array letter (Numeric or String)
 - Loading address (Code)
- The number of free sectors available on the drive is given as the product of the number of free clusters and the cluster size
- If no name is specified, all files in the current directory are catalogued
- If a name is specified, then its path indicates the directory to be catalogued and the filename and extension are used as filters for the output of the command, whereby the extension '.d' is accepted, to display only directories
- If followed by the token ABS, the CAT command outputs only the absolute path
- If followed by the token LN, the CAT command outputs a long-name file list of a file device directory
- If the specified drive number is 0, then the name is considered a command and is sent to the peripheral port, to be interpreted by either the server, if it ends in a "/", or otherwise by the peripheral controller. After processing the command, these are expected to send a response, which is printed out as hex-dump, if not suppressed by a NOT option token

2.6.3 The FORMAT command

- The statement 2.1.6 and 2.1.7 do not apply to file devices.
- The statement 2.1.7 for device "M", identifies the flash card and sets the number of reserved logical drives if a <rsd> value is specified. For device "R", it clears the application data area and the 'current directories' table.
- The option token NOT suppresses the screen output.
- The allowed values of the cluster size in statement 2.1.6 are: 2, 4, 8 and 16. Any other number is disregarded and the default value: 8 is used instead
- The baud rate specified in statement 2.1.8 is rounded up to the next standard value in the set: 300, 600, 1200, 2400, 4800, 9600, 19200, 57600, 115200

2.6.4 The MOVE command

- If both source and destination are files, the operation is performed sector by sector, rather than byte by byte.
- If source and destination device and logical drive are the same, the source file can be renamed, if source and destination are located in the same directory, or otherwise moved. Following separators can be used:

Separator	Operation	Overwrite existing files
TO	copy	No
OVER	copy	Yes
AT	move/rename	No
IN	move/rename	Yes
MERGE	append	

- Both source and destination file names are considered from the perspective of the current location: device, drive, directory
- The statement 2.1.16 is repetitive. It processes all files that match the specified source name.
- The statement 2.1.17 copies the record pointer and the record number of the file, to which the stream is currently opened, to the system variables: HD__0F and respectively HD__11
- The statement 2.1.18 sets the file pointer of the file, to which the stream is currently opened, to a specified position. If the stream is opened to a "M" channel and the specified position is out of range, the file pointer is set to EOF

2.6.5 The OPEN command

- Any file can be opened for sequential access, not only PRINT files
- The optional keywords IN, OUT or OVER force the opening of the file for reading or respectively writing
- Opening a non-existing file for reading, using option IN, generates the error report "File not found"
- Writing to an existing file will either append to it, if it was opened with option OUT, or overwrite it, if option OVER was used.

- Opening a file with the option RND creates a random access 'handle' for that file

2.6.6 The SAVE, LOAD and VERIFY commands

- The auto-run feature can be suppressed by specifying the file extension '.p' (lower case) when loading a program
- The option BIN allows to load, save or verify a memory block from/to a given position of any type of file
- If the file is accessed via a 'handle' rather than a specifier, the pointer entered with the option BIN is not used, but instead the one stored in the corresponding "H" channel descriptor, which is set to 0 when the file is opened and subsequently updated automatically following each operation.

2.6.7 The ERASE command

- The form: ERASE <fsq> is repetitive. It processes all files that match the specified name.

2.6.8 The printer commands.

- The printer commands: LPRINT, LLIST and COPY work as expected with a ESC/P printer.
- The block graphics and UDG characters are printed as bitmaps at a density of 80 DPI.
- The system variables P_POSN and PR_CC are used as follows:

Variable	Address	Length	Description
P_POSN	#5C7F 23679	1	Column number
PR_CC	#5C80 23680	1	Lines per page minus Line number
	#5C81 23681	1	Bit 7 reset = 64 columns set = 32 columns
			Bits 0-6 = Lines per page

- OPEN #<str>,"P" sends an initialization string to the printer
- While the interface is connected to a server PC, the print jobs are forwarded to the server application, which directs them to a printer or a spool file.

3. Data structures

3.1 The ZX INTERFACE 1 system variables

Variable	Address	Length	Replaces
FLAGS3	#5CB6	23734	1
VECTOR	#5CB7	23735	2
..			
SER_FL	#5CC7	23751	2
..			
DRV_NR	#5CD6	23766	1 D_STR1
PTH_LN	#5CD7	23767	1
STR_NR	#5CD8	23768	1 S_STR1
DEV_LT	#5CD9	23769	1 L_STR1
NAM_LN	#5CDA	23770	1 N_STR1
FIL_TY	#5CDB	23771	1
NAM_AD	#5CDC	23772	2 P_STR1
DRV_N2	#5CDE	23774	1 D_STR2
PTH_L2	#5CDF	23775	1
STR_N2	#5CE0	23776	1 S_STR2
DEV_L2	#5CE1	23777	1 L_STR2
NAM_L2	#5CE2	23778	1 N_STR2
FIL_T2	#5CE3	23779	1
NAM_A2	#5CE4	23780	2 P_STR2
HD__00	#5CE6	23782	1 HD_00
HD__0B	#5CE7	23783	2 HD_0B
HD__0D	#5CE9	23785	2 HD_0D
HD__0F	#5CEB	23787	2 HD_0F
HD__11	#5CED	23789	1 HD_11
HD__DV	#5CEE	23790	1
HD__DR	#5CEF	23791	1 COPIES

- The variables not shown are not used

3.1.1 FLAGS3

Bits 0-4 have the same significance as in the original ZX INTERFACE 1 'Shadow ROM'

- Bit 0 Shadow ROM entered the second time for the same error
- Bit 1 Shadow ROM entered the first time after creation of the new system variables, or CLEAR# command in progress
- Bit 2 Shadow ROM entered by means of a hook-code
- Bit 3 CAT command in progress
- Bit 4 Character by character MOVE command in progress, or Destination name in MOVE command contains wild cards, or A filename was specified in the CAT command, or Suppress auto-run of a loaded BASIC program, or SAVE / LOAD option specified in upper case
- Bit 5 H(andle) channel SAVE / LOAD / VERIFY in progress
- Bit 6 Find the 'last match' in a search operation
- Bit 7 Find the 'next match' in a search operation

3.1.2 VECTOR and SER_FL

Same as in the original ZX INTERFACE 1 'Shadow ROM'

3.1.3 File specifiers

The two 8-byte file specifiers at DSTR_1 and DSTR_2 have the same function as in the original ZX INTERFACE 1 'Shadow ROM', except for the drive number's high byte, which is used to store the path name's length and the file name's length high byte, which is used to store the file type

3.1.4 BASIC header: HD__00 .. HD__11

Same as in the original ZX INTERFACE 1 'Shadow ROM'

3.1.5 HD__DV and HD__DR

Replace HD_11 high byte and COPIES. Store the device and drive number

3.2 The M channel descriptor

Offset	Name	Description
0		Address of error handling routine (0008)
2		Address of error handling routine (0008)
4		Channel type ("M" or "M"+128 for 'ad-hoc' channels)
5		Address of output subroutine
7		Address of input routine
9		Length of channel (595)
11	CHBYTE	Record pointer (0-512).
13	CHREC	Record number, lower byte
14	CHNAME	10 byte filename with trailing spaces
24	CHFLAG	Flag byte: bit 0 - file open for write bit 1 - file doesn't exist
25	CHDRIV	Drive number
26	CHMAP	- Parent directory number, for a block device, or - File handle, for a file device
28		File type literal.
29		Record number, upper byte.
30		Device code
..		Not used
67	RECFLG	Flag byte: bit 0 = 0 bit 1 = last record bit 2 = not a PRINT file
68	RECNUM	Not used
69	RECLEN	Number of bytes of data in the current record (0-512)
71	RECNAM	Not used
81	DESCHK	Not used
82	CHDATA	512 bytes of data
594	DCHK	Not used

3.3 The N channel descriptor

Offset	Name	Description
0		Address of error handling routine (0008)
2		Address of error handling routine (0008)
4		Channel type ("N" or "N"+128 for 'ad-hoc' channels)
5		Address of output subroutine
7		Address of input routine
9		Length of channel (276)
11	NCIRIS	Destination station number
12	NCSELF	Not used
13	NCNUMB	Block number
15	NCTYPE	Packet type code... 0 data, 1 EOF
16	NCOBL	Number of bytes in the data block
17	NCDCS	Not used
18	NCHCS	Not used
19	NCCUR	Position of the last character taken from the buffer
20	NCIBL	Number of bytes in the input buffer
21	NCB	255 byte data

3.4 The 'work RAM' (512 bytes) Mapped at address #2E00 of the 'Shadow ROM'

- 3.4.1 Main logical drive descriptor (13 bytes)
 - #2E00 CRT_DV Current device code
 - #2E01 CRT_DR Current drive number
 - #2E02 PRV_DV Previous device code
 - #2E03 PRV_DR Previous drive number
 - #2E04 CLU_SZ Cluster size - 1
 - #2E05 RES_DR Number of reserved logical drives
 - #2E06 ROOT_D First sector number of root directory
 - #2E07 ALC_SN Number of last allocated sector
 - #2E09 ICL_SN In-cluster sector number
 - #2E0B FAT_SN Pointer to the FAT sector number
- 3.4.2 Alternate logical drive descriptor (13 bytes)
Same structure as the main descriptor
 - #2E0D ALT_DV
- 3.4.3 Default block and file device numbers (2 bytes)
 - #2E1A DEF_ST
- 3.4.4 Flash media sizes, in logical drives (8 bytes)
 - #2E1C FLA_DN
- 3.4.5 Spare initialized variable space (12 bytes)
 - #2E24 VARS_E Spare initialized variables

- 3.4.6 Sector buffer pointers (16 bytes)
 - #2E30 SECT_0 Pointer for buffer 0
 - #2E32 SECT_1 Pointer for buffer 1
 - #2E34 SECT_A Pointer for buffer A
 - #2E36 SECT_3 Pointer for buffer 3
 - #2E38 SECT_L Pointer for buffer L
 - #2E3A SECT_S Pointer for buffer S
 - #2E3C SECT_F Pointer for buffer F
 - #2E3E SECT_Z Parent directory number

- 3.4.7 'Output tape' variables (10 bytes)
 - #2E40 S_FLAG Flag
 - #2E41 S_FSEC First sector
 - #2E43 S_DIRN Directory nr
 - #2E45 S_PNTR Record pointer
 - #2E47 Not used
 - #2E48 S_DEVN Device code
 - #2E49 S_DRVN Drive number

- 3.4.8 'Input tape' variables (10 bytes)
 - #2E4A L_FLAG Flag
 - #2E4B L_FSEC First sector
 - #2E4D L_FPNT Record pointer
 - #2E4F L_LEN_L Length (low)
 - #2E51 L_LEN_H Length (high)
 - #2E52 L_DEVN Device code
 - #2E53 L_DRVN Drive number

- 3.4.9 Snapshot variables (10 bytes)
 - #2E54 Z_FLAG Flag
 - #2E55 Z_FSEC First sector
 - #2E57 TMP_HL Temporary store for
 - #2E59 TMP_AD the HL registers
 - #2E5B Z_TYPE Snapshot type
 - #2E5C Z_DEVN Device code
 - #2E5D Z_DRVN Drive number

- 3.4.10 Printer buffer pointer (2 bytes)
 - #2E5E PBF_PT

- 3.4.11 File device descriptor (16 bytes)
 - #2E60 N_DESC File type
 - #2E6B N_HNDL File handle
 - #2E6D N_FLEN File length

- 3.4.12 Auxiliary (4 bytes)
 - #2E70 AX_FLG Flags
 - #2E71 AX_CMD Control
 - #2E72 AX_ERR Error
 - #2E73 Station

- 3.4.13 Flash drive block number (6 bytes)
 - #2E74 BLK_LO Low word
 - #2E76 BLK_HI High word
 - #2E78 DAT_LN Data length

- 3.4.14 Copy/Rename destination file parameters (5 bytes)
 - #2E7A DST_TY File type
 - #2E7B DST_LN Filename length
 - #2E7D DST_AD Filename address

- 3.4.15 Temporary Filename buffer (11 bytes)
 - #2E7F TMP_TY Type
 - #2E80 TMP_NA Name

- 3.4.16 Directory entry location (6 bytes)
 - #2E8A D_NUMB Directory number
 - #2E8C D_SECT Sector number
 - #2E8E D_PNTR Pointer

- 3.4.17 Internal stack (48 bytes)
 - #2E90 ST_BOT Stack bottom

- 3.4.18 Printer buffer (64 bytes)
 - #2EC0 PR_BUF

- 3.4.19 Page buffer (256 bytes)
 - #2F00 PAGE_B

3.5 sector buffers (4 KB)

3.5.1 Sector buffers

#3000	BUFF_0	Main sector
#3200	BUFF_1	Allocation
#3400	BUFF_A	Alternate FAT
#3600	BUFF_3	Work
#3620	BUFF_P	POKE buffer
#3800	BUFF_L	'Input tape'
#3A00	BUFF_S	'Output tape'
#3C00	BUFF_F	Main FAT

3.5.2 Application data (256 bytes)

#3E00	AP_DAT
-------	--------

3.5.3 Current directories table (256 bytes)

Stores the last 64 block device directories used

#3F00	DIR_TB
-------	--------

4. Peripheral Controller

All Data Input/Output operations are handled by sending commands to the interface's peripheral controller

4.1 Communication protocol

4.1.1 The structure of the command block is:

Byte Nr.	Description
Header 0	Command byte: bits 0-3 = command parameter bits 4-7 = command code
1	Control byte: bits 0-2 = station number bits 3-7 = flags
2,3	Length of data: n = 0-512
Data 4..(n+4)	(n) bytes of data

4.1.2. After processing the command, the peripheral controller sends back a 'reply block':

Byte Nr.	Description
Header 0	Error code or: 0 = No error
1	Flags
2,3	Length of data: n = 0-512
Data 4..(n+4)	(n) bytes of 'reply' data

4.2 General Commands

Code	Command	Parameter	Data	Response
13	Server command	0	Command	Result
14	Receive-Transmit	0		
15	Peripheral command	0	Command	Result

The Receive-Transmit command relays data blocks from a file opened for reading on a file device, to a new file created on a different file device.

4.3 Server Commands

Code	Command	Parameter	Data	Response
13	Request OpSys	1		Sector
13	Dump OpSys	2	Sector	
13	Print buffered data	3	Data	
13	Set page	4	Dir, Pag	Dir Info
13	Get next line	4		Type, Name
13	Search from line	4	Line	Pag, Line
13	Get file	5	Type, Name	Type, Name
13	Update index	6		
13	Set search string	6	String	
13	Send chat message	7	Message	
13	Receive chat message	8		Message

- The 'Update index' command creates a new index for an online archive
- The 'Set page' command sets the current position of an online archive index, from which subsequent 'Get next line' commands retrieve file names and types
- The 'Get file' command retrieves a file from an online archive
- The 'Search from line' command searches the index of an online archive for a string defined in a previous 'Set search string' command

4.4 Communication port commands

Code	Command	Parameter	Data	Response
14	Serial port transmit	1	Data	
14	Serial port receive	2	Length	Data
14	Network port transmit	3	Data	
14	Network port receive	4	Length	Data
14	Link network station	5	Name/IP	
14	Check network station	6		

- The 'Serial port transmit' command sends 1-512 bytes of data
- The maximum number of bytes to be received by the 'Serial port receive' command: 'Length' can be specified in the range (1-256)
- The data length for the network transmit and receive commands is limited to 255 bytes
- The check network station command returns the zero flag set if the corresponding station is linked

5. File systems

Two different file systems are implemented for:

- Block devices: SD card flash memory and RAM drive
- File devices: Computers running a suitable server application

5.1 Block devices

A block device can be either:

- local (station 0): the on-board SD card or RAM drive, or
- remote (station 1-7): the SD card of a unit on an IP network

5.1.1 Filesystem features

- A SD card volume is implicitly partitioned into fixed-sized logical disks of 32 MB (65536 sectors of 512 bytes).
- The size of a SD card volume is limited to 255 logical disks = 8 GB.
- The allocation unit (cluster) can be of: 2, 4, 8 or 16 sectors
- The FAT contains (65536 / cluster size) entries, occupying sectors 1 through (256 / cluster size).
- Sector number (256 / cluster size) + 1 contains the first record of the root directory.
- The RAM drive has a capacity of 256 sectors = 128 KB. The cluster size is of 1 sector. Sector 1 contains the FAT and sector 2 the first record of the root directory.
- The logical drives are formatted according to a simplified 16-bit FAT system, the FAT entries being sector, rather than cluster numbers.
- Sector number 0 of any logical drive is not used.
- As cluster 0 is always occupied by the FAT, the corresponding FAT entry (bytes 0 and 1 of sector 1) is used to store the cluster size.
- As sectors 0 and 1 are not available for allocation, the corresponding FAT entry values are used for marking:
 - 0000 = Free cluster
 - 0001 = Last cluster of the file
- Directories contain sequences of 16-byte 'file specifiers':

Offset	Length	Description
0	1	File type
1	10	File name
11	2	First sector of the file
13	3	File length

with the byte #FF as an end marker.

- The size of a (sub)directory is limited only by the available space and the depth of the directory tree by the maximum length of the path name: 255 bytes.
- The first entry of the first record of a directory has the following structure:

Offset	Length	Description
0	1	Type: 16
1	10	Directory name
11	2	First sector of the parent directory, or 00 00 for the root directory
13	3	00 00 00

- When a file is deleted, the parent directory is compacted by reclaiming the corresponding entry and shifting all further entries to its right, which can span several records, downwards by 16 bytes.

5.1.2 Block device commands

Code	Command	Parameter	Data	Response
15	Play tape file	3	Add Len Typ	
15	Receive - Write	4	Address	
15	Set write address	5	Address	
15	Read multiple blocks	6	Address	
15	Align to block start	6		
15	Read - Transmit	7	Address	
15	Read single block	8	Address	Sector
15	Read to buffer	9	Address	
15	Write block	10	Sector	
15	Write from buffer	11	Address	
15	Identify card	12		Identifier
15	First erase address	13	Address	
15	Last erase address	14	Address	
15	Erase block	15		

5.1.3 Notes

- 'Address' is a 4-bytes sector number
- A sector number with the highest bit (31) set, is interpreted as a RAM drive address
- 'Identifier' is an 8-bytes card identifier string followed by the 4-bytes capacity (last sector number) of the card
- 'Sector' is a 512-bytes block of data
- Writing a sector requires two steps:
 1. The sector number is specified in a 'Set write address' command
 2. The data is sent via a 'Write sector' command
- The 'write from buffer' writes the sector read by a previous 'Read to buffer' command
- Erasing a block requires three steps:
 1. The start of the block is specified in a 'First erase address' command
 2. The end of the block is specified in a 'Last erase address' command
 3. The 'Erase block' command is sent to actually erase the block
- The 'Receive - write' command sends a 'Read sector' command to a file device and writes the received block to the specified address of a block device
- The 'Read - Transmit' command reads a block from the specified address of a block device and sends it to a file device by issuing a 'Write sector' command
- After a 'Read multiple blocks' command, a stream of bytes can be read at the rate of the INI instruction (16 T-states) until stopped by re-issuing the command (with an arbitrary address). The read pointer can be anytime advanced to the beginning of a block, by issuing the same command without an address.
- The 'Play tape file' command needs as data:
 - Address of the file's first sector : 4 bytes
 - File length : 3 bytes
 - File type (#09 or #0A) : 1 byte

5.1.4 Large Sequential Files partition

- In order to accomodate multimedia files (types 12 -14), of up to 4GB length, command 2.1.7 can be used to reserve <rsd> (1 - 127) logical drives, for a 'Large Sequential Files' partition, which also includes the SD card storage space exceeding the maximum volume size of 8 GB.
- The directory of the partition is stored starting at sector 0 of logical drive: (<volume size in sectors>/65536 - <rsd> + 1) and spans 4 sectors containing a maximum of 128 entries with the following structure:

Offset	Length	Description
0	1	File type (12 - 14)
1	10	File name
11	2	00 00
13	3	File length (in sectors)

and the byte #FF as an end marker.

It is followed by files, stored contiguously in the order in which they appear in the directory

- The sector number of a files's first record needs to be calculated by parsing the directory and adding 4 + the cumulated length of the preceding ones to the start of the partition

5.2 File devices

A file device can be either:

- local (station 0): a server machine connected via the USB port, or
- remote (station 1-7): a server machine on an IP network

5.2.1 File device commands

Code	Command	Parameter	Data	Response
0	Close File	Handle		
0	Open reserved handle	Handle	Type	Handle,Size
1	Read sector	Handle		Sector
2	Write sector	Handle	Sector	
3	Set file pointer	Handle	Position	
3	Get file size	Handle		File size
3	Get handle info	Handle	Byte	Handle,Size
4	Create temp. file	Type	Name	Handle,Size
5	Create perm. file	Type	Name	Handle,Size
5	Reserve handle	15	Name	Handle
5	Get long name	1		Long name
5	Get disk free space	2		Free space
5	Set 'Overwrite'flag	3		
5	LSQ file mode on	4		
5	LSQ file mode off	5		
6	Open temp. file	Type	Name	Descriptor
6	Open next file	0-15		Descriptor
7	Open perm. file	Type	Name	Descriptor
8	Find file	Type	Name	Descriptor
8	Find next file	0-15		Descriptor
9	Delete file	Type	Name	
9	Delete file	Handle		
10	Rename file	Type	Name	
11	Copy file	Type	Name	
12	First file list	0	Name	List
12	First BASIC file list	1	Name	List
12	First long name line	2	Name	Line
12	Next file list/line	0-2		List/line
12	Get absolute path	3	Name	Path
12	Get current path	3		Path
12	Select directory	13	Name	
12	Make directory	14	Name	
12	Remove directory	15	Name	

5.2.2 Notes

- A 'Handle' is a number in the range: 0-15
- The 'Type' is a number in the range: 0-15 defined at (3.4)
- A 'permanent file' is allocated a handle = 1-15. The same handle is not re-allocated before the file is explicitly closed.
- A 'temporary file' is always allocated the handle = 0. Creating or opening another 'temporary file' automatically closes the previous one
- The 'Open reserved handle' function creates the file specified when the handle was reserved, with the given type
- For the 'Set file pointer' command the position can be specified either on two, or on four bytes: [<Pos>]<Rec>, where <Pos>= record pointer (0-511) and <Rec>= record number (0-32767). If only one byte is sent, the pointer is set at the start of the file and its handle and length are returned as a response, on five bytes
- 'Get long name' returns the long name of the last file found using the 'Find [next] file' command

5.2.3 The 'Find [next] file' command returns a 16-bytes descriptor:

Offset	Length	Description
0	1	Actual file type (0-11)
1	10	Actual file name
11	2	Invalid file handle (255)
13	3	File length

5.2.4 The descriptor returned by the 'Open temporary|permanent [next] file' command contains also the file handle

Offset	Length	Description
0	1	Actual file type (0-11)
1	10	Actual file name
11	2	File handle (0-15)
13	3	File length

5.2.5 The handle returned by the 'Create temporary|permanent file' command is a two-byte number in the range 0-15.
- After a 'Find file' or 'Open temp. file' command, a subsequent 'Find next file' or 'Open next file' command will attempt to find/open the next file with a matching name.

5.2.6 A copy|rename operation requires two steps:
- The source file is found by issuing a 'Find [next] file' command.
- The destination name is specified in a 'Rename file' or 'Copy file' command.

5.4.7 The list returned by the 'First|Next file list' command consists of a sequence of 16-bytes file descriptors, as returned by the 'Find [next] file' command, with #FF as an end marker. The descriptors returned by the 'First|Next BASIC file list' command contain, for file types 0-3, also information from the 9-byte BASIC header of the files:

Offset	Length	Description
0	1	File type (0-3)
1	10	File name
11	1	File type (from BASIC header)
12	2	File length
14	2	Start line (Program), or Array name (Numeric or String), or Loading address (Code)

5.2.7 'First long list line' and 'Next long list line' are similar to 'First directory list' and respectively 'Next directory list' but return a single directory list line with the following layout:

Col	Length	Description
0	1	File type literal
1	1	" " (space) character
2	1-253	Long filename

6. Hook Codes

The ZX INTERFACE 1 hook codes.

Nr	Label	Description
#1B	WAI_KY	Console input
#1C	S_PRNT	Console output
#1D	BCHN_I	R232 input
#1E	BCHN_O	R232 output
#1F	L_PRNT	Printer output
#20	TST_KY	Keyboard test
#21	SET_DN	Select drive
#22	OPN_CH	Open channel
#23	CLO_CH	Close channel
#24	ERAS_F	Delete file
#25	RD_SQE	Read sequential
#26	WR_SQE	Write sequential
#27	RD_REC	Read random
#28	RD_CSC	Read sector
#29	NX_CSC	Read next sector
#2A	WR_CSC	Write sector
#2B	CRE_CH	Create channel
#2C	DEL_CH	Delete channel
#2D	OPEN_N	Open network channel
#2E	CLOS_N	Close network channel
#2F	GET_PK	Get packet
#30	SND_PK	Send packet
#31	PA_INI	Create system variables
#32	EXEC_C	Call shadow ROM routine
#33		Read next header - Not implemented
#34	OPEN_B	Open B channel

Additional hook codes

Nr	Label	Description
#35	RD_SEC	Read sector
#36	WR_SEC	Write sector
#37	NEXT_R	Next sector
#38	RCLM_A	Reclaim 'ad-hoc' channels
#39	FIND_F	Find file
#3A	FILE_N	Find next file
#3B	OPEN_S	Open stream
#3C	CLOS_S	Close stream
#3D	SAV_LD	SAVE / LOAD / VERIFY / MERGE
#3E	MOVE_E	Copy file or Set file pointer
#3F	CAT_LG	Catalogue of a directory
#40	FORM_T	Format drive
#41	CLR_SC	Clear screen
#42	CLOS_A	Close all streams
#43	SV_CMD	Custom peripheral command
#44	A_PATH	Get absolute path
#45	DIR_FL	Get first directory list
#46	DIR_NL	Get next directory list
#47	PR_INT	Print 3-byte integer
#48	NDT_CM	Send command only
#49	DAT_CM	Send data command
#4A	DEV_PA	Get device size
#4B	ASY_RX	Serial port Receive
#4C	ASY_TX	Serial port Transmit
#4D	GET_LN	Read line from M channel
#4E	PUT_LN	Write line to M channel
#4F	DV_PRE	check device presence
#50	NET_RX	Ethernet Receive
#51	NET_TX	Ethernet Transmit
#52	LONG_N	Get long name

- A 'File type literal' is a character as defined at (3.4)
- A 'Device type literal' is a character as defined at (3.2)
- A 'Device code' is a byte defined as:
 - Bits 0-2: Device (station) number (0-7)
 - Bit 3 : 0
 - Bits 4-6: 'Device type' (0-7) as defined at (3.2)
 - Bit 7 : 1

6.1 Console input (#1B)

6.1.1 Action: wait for a key to be pressed

6.1.2 Input data: None

6.1.3 Output data:

- (A) = Character code

6.2 Console output (#1C)

6.2.1 Action: Send a character to the screen

6.2.2 Input data:

- (A) = Character code

6.2.3 Output data: None

6.3 RS232 input (#1D)

6.3.1 Action: Receive a character from the RS232 output
Times out after 1 second

6.3.2 Input data: None

6.3.3 Output data:

- CY = A character has been received
- (A) = Character code

6.4 RS232 output (#1E)

6.4.1 Action: Send a character to the RS232 output
wait indefinitely

6.4.2 Input data:

- (A) = Character code

6.4.3 Output data: None

6.5 Printer output (#1F)

6.5.1 Action: Print a character to the printer

6.5.2 Input data:

- (A) = Character code

6.5.3 Output data: None

6.6 Keyboard test (#20)

6.6.1 Action: Test if a key is being pressed

6.6.2 Input data: None

6.6.3 Output data:

- CY = A key is being pressed

6.7 Select drive (legacy) (#21)

6.7.1 Action: Set drive (A) as current for device (DEV_LT)

6.7.2 Input data:

- (A) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code

6.7.3 Output data: None

6.8 Open channel (#22)

6.8.1 Action: Open a channel.

6.8.2 Input data:

File channel

- (A) = #BF (IN) - Open for read
- = #DE (OVER) - Overwrite
- = #DF (OUT) - Open for write
- = #A5 (RND) - Create a file handle
- (DRV_NR) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)

Non-file channel

- (DRV_NR) = Station number (0-7) only for channel N
- (DEV_LT) = Channel type: B, T, N or U
- (NAM_LN) = Invalid name length (0 or 255)

6.8.3 Output data:

- (HL) = channel offset
- (IX) = Address of the channel descriptor

6.9 Close channel (#23)

6.9.1 Action: Close a channel

6.9.2 Input data:

- (IX) = Address of the channel descriptor

6.9.3 Output data: None

6.10 Delete file (#24)

6.10.1 Action: Delete a file

6.10.2 Input data:
File parameters, specified by a descriptor or a handle
Descriptor:
- (DRV_NR) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)
Handle:
- (STR_NR) = Handle (0-15)
- (NAM_LN) = Invalid name length (0 or 255)

6.10.3 Output data: None

6.11 Read sequential (#25)
6.11.1 Action: Read the next record
6.11.2 Input data:
- (IX) = Address of the channel descriptor
6.11.3 Output data: None

6.12 Write sequential (#26)
6.12.1 Action: Write the current record
6.12.2 Input data:
- (IX) = Address of the channel descriptor
6.12.3 Output data: None

6.13 Read record (#27)
6.13.1 Action: Read the current record
6.13.2 Input data:
- (IX) = Address of the channel descriptor
6.13.3 Output data: None

6.14 Read sector (legacy) (#28)
6.14.1 Action: Read sector CHREC into channel buffer
6.14.2 Input data:
- (IX) = Address of the channel descriptor
6.14.3 Output data:

6.15 Read next sector (legacy) (#29)
6.15.1 Action: Read next sector into channel buffer
6.15.2 Input data:
- (IX) = Address of the channel descriptor
6.15.3 Output data:

6.16 Write sector (legacy) (#2A)
6.16.1 Action: Write channel buffer to sector CHREC
6.16.2 Input data:
- (IX) = Address of the channel descriptor
6.16.3 Output data:

6.17 Create channel (#2B)
6.17.1 Action: Create a channel descriptor
6.17.2 Input data:
File channel
- (A) = #BF (IN) - Open for read
= #DE (OVER) - Overwrite
= #DF (OUT) - Open for write
= #A5 (RND) - Create a file handle
- (DRV_NR) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)
Non-file channel
- (DRV_NR) = Station number (0-7) only for channel N
- (DEV_LT) = Channel type: B, T, N or U
- (NAM_LN) = Invalid name length (0 or 255)
6.17.3 Output data:
- (HL) = Channel offset
- (IX) = Address of the channel descriptor

6.18 Delete channel (#2C)
6.18.1 Action: Delete a channel descriptor
6.18.2 Input data:
- (IX) = Address of the channel descriptor
6.18.3 Output data: None

6.19 Open network channel (#2D)
6.19.1 Action; Create a N channel
6.19.2 Input data: None
6.19.3 Output data:
- (HL) = Channel offset
- (IX) = Address of the channel descriptor

6.20 Close network channel (#2E)

6.20.1 Action; Send buffered data and delete a N channel descriptor

6.20.2 Input data:

- (IX) = Address of the channel descriptor

6.20.3 Output data: None

6.21 Get packet (#2F)

6.21.1 Action; Receive a packet of 255 bytes into a N channel. Times out after 1 second

6.21.2 Input data:

- (IX) = Address of the channel descriptor

6.21.3 Output data:

- CY = Time-out

6.22 Send packet (#30)

6.22.1 Action: Send a packet of maximum 255 bytes from a N channel. wait indefinitely

6.22.2 Input data:

- (IX) = Address of the channel descriptor

6.22.3 Output data:

6.23 Create system variables (#31)

6.23.1 Action: Create system variables

6.23.2 Input data: None

6.23.3 Output data: None

6.24 Execute code (#32)

6.24.1 Action: Execute code from address (HD__11)

6.24.2 Input data:

- (HD__11) = Address of the executable code

6.24.3 Output data: None

6.25 Read next header (#33)

6.25.1 Action: Not implemented

6.25.2 Input data: None

6.25.3 Output data: None

6.26 Create B channel (#34)

6.26.1 Action: Create a B channel

6.26.2 Input data: None

6.26.3 Output data:

- (HL) = channel offset
- (IX) = Address of the channel descriptor

6.27 Read sector (#35)

6.27.1 Action: Read sector into buffer.

- Does not apply to file devices

6.27.2 Input data:

- (BC) = Sector Number (0-65535)
- (HL) = Buffer address (0-65536)

If CY reset:

- (A) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code

6.27.3 Output data: None

6.28 Write sector (#36).

6.28.1 Action: write sector from buffer.

- Does not apply to file devices

6.28.2 Input data:

- (BC) = Sector Number (0-65535)
- (HL) = Buffer address (0-65536)

If CY reset:

- (A) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code

6.28.3 Output data: None

6.29 Next sector (#37).

6.29.1 Action: Find the sector number of the next record of a file.

- Initially it should be called with (BC)=0 to flush the FAT buffer.
- Does not apply to file devices

6.29.2 Input data:

- (BC) = sector number (0-65535)

If CY reset:

- (A) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code

6.29.3 Output data:

- Zero flag set = No more records
- (BC) = Next sector number (0-65535)

6.30 Reclaim all 'ad-hoc' channels (#38).

6.30.1 Action: Reclaim all channels not associated with streams.

6.30.2 Input data: None

6.30.3 Output data: None

6.31 Find file (#39).

6.31.1 Action: Find a file or directory.

6.31.2 Input data:
File parameters, specified by a descriptor or a handle
Descriptor:
- (DRV_NR) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)
Handle:
- (STR_NR) = Handle (0-15)
- (NAM_LN) = Invalid name length (0 or 255)

6.31.3 Output data:
- CY = File not found
- (HL) = Pointer to the file descriptor

6.32 Find next file (#3A).

6.32.1 Action: Find the next file with a name matching the one specified in a previous 'Find file' call

6.32.2 Input data: none

6.32.3 Output data:
- CY = File not found
- (HL) = Pointer to the file descriptor

6.33 Open stream (#3B).

6.33.1 Action: Open a stream or handle..

6.33.2 Input data:
File channel
- (A) = #BF (IN) - Open for read
= #DE (OVER) - Overwrite
= #DF (OUT) - Open for write
= #A5 (RND) - Create a file handle
- (DRV_NR) = Drive number (1-255)
- (STR_NR) = Stream or handle number (0-15).
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)
Non-file channel
- (DRV_NR) = Station number (0-7) only for channel N
- (STR_NR) = Stream number (0-15).
- (DEV_LT) = Channel type: B, T, N or U
- (NAM_LN) = Invalid name length (0 or 255)

6.33.3 Output data: None.

6.34 Close stream (#3C).

6.34.1 Action: Close a stream or a handle.

6.34.2 Input Data:
- (A) = Stream or handle number (0-15)

6.34.3 Output data: None

6.35 SAVE / LOAD (#3D).

6.35.1 Action:
- Read/write memory contents from/ to a file.
- Create (SAVE) / change (LOAD) directory.
- Close the 'input tape', 'output tape' or 'snapshot file'

6.35.2 Input data:
Operation type:
- (A) = 0,4 - SAVE , clear file pointer and SAVE
= 1,5 - LOAD , clear file pointer and LOAD
= 2,6 - VERIFY, clear file pointer and VERIFY
= 3,7 - MERGE , clear file pointer and MERGE
= 8 - Close the output tape
= 9 - Close the input tape
= 10 - Close the snapshot file
File parameters, specified by a descriptor or a handle
Descriptor:
- (DRV_NR) = Drive number (1-255)
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)
Handle:
- (STR_NR) = Handle (0-15)
- (NAM_LN) = Invalid name length (0 or 255)
Data type code
- (HD__00) = File type number as defined at (3.4)

If not 7 (binary data), overridden by the specified file type literal.
Parameters of BASIC program, only for data type: 0
- (HD__11) = Auto-run line number
Parameters of BASIC array, only for data types: 1 & 2
- (HD__0F) = Array name ("a"- "z").
Address of memory block, only for data types: 3 - 7
- (HD__0D) = Address of memory block.
Length of memory block, only for data types: 3 - 7
- (HD__0B) = Length of memory block.
File pointer, only for data types: 4 - 7
- (HD__0F) = Record pointer (0-511)
- (HD__11) = Record number (0-32767)
The pointer is automatically updated after the operation

6.35.3 Output data: None

6.36 Copy /rename file (#3E).

6.36.1 Action: Copy or rename files / rename directory

6.36.2 Input data:

Operation type (only for two file channels):

- (A) = #CC (TO) - Copy
- = #AC (AT) - Rename
- = #DE (OVER) - Copy Overwrite
- = #BF (IN) - Rename Overwrite
- = #D5 (MERGE) - Append

Source: specified by a stream or a channel

Stream:

- (STR_NR) = Stream number (0-15)
- (NAM_LN) = Invalid name length (0 or 255)

File channel:

- (DRV_NR) = Drive number (1-255)
- (STR_NR) = 255
- (DEV_LT) = Device type literal or device code
- (NAM_LN) = Length of filename (1-254)
- (FIL_TY) = File type literal
- (NAM_AD) = Address of filename (0-65535)

Non-file channel:

- (DRV_NR) = Station number (0-7) only for channel "N"
- (STR_NR) = 255
- (DEV_LT) = Channel type: B, T, N or U
- (NAM_LN) = Invalid name length (0 or 255)

Destination, specified by a stream or a file descriptor

Stream:

- (STR_N2) = Stream number (0-15)
- (NAM_L2) = Invalid name length (0 or 255)

File channel:

- (DRV_N2) = Drive number (1-255)
- (STR_N2) = 255
- (DEV_L2) = Device type literal or device code
- (NAM_L2) = Length of filename (1-254)
- (FIL_T2) = File type literal
- (NAM_A2) = Address of filename (0-65535)

Non-file channel:

- (DRV_N2) = Station number (0-7) only for channel "N"
- (STR_N2) = 255
- (DEV_L2) = Channel type: B, T, N or U
- (NAM_L2) = Invalid name length (0 or 255)

6.36.3 Output data: None

6.37 Get file pointer (#3E).

6.37.1 Action: Get the file pointer of a M or H channel to which a given stream is opened.

6.37.2 Input data:

- (A) = Operation type: #B0 (VAL)

Channel, specified by its address or stream number
Channel address:

- (STR_NR) = Invalid stream number (16-255)
- (IX) = Channel address (0-65535)

Stream number:

- (STR_NR) = Stream number (0-15)

6.37.3 Output data:

Channel pointer:

- (HD__0F) = Record pointer (0-511)
- (HD__11) = Record number (0-32767)

6.38 Set file pointer (#3E).

6.38.1 Action: Set the file pointer of a M or H channel to which a given stream is opened.

6.38.2 Input data:

- (A) = Operation type: #A9 (POINT)

Channel, specified by its address or stream number
Channel address:

- (STR_NR) = Invalid stream number (16-255)
- (IX) = Channel address (0-65535)

Stream number:

- (STR_NR) = Stream number (0-15)

Channel pointer:

- (HD_0F) = Record pointer (0-511)
 - (HD_11) = Record number (0-32767)
- 6.38.3 Output data: None
- 6.39 Catalogue (#3F).
- 6.39.1 Action: Produce a file catalogue.
- 6.39.2 Input data:
- (DRV_NR) = Drive number (1-255)
 - (STR_NR) = Stream number (0-15)
 - (DEV_LT) = Device type literal or device code
 - (NAM_LN) = Invalid name length (0 or 255)
- A directory or file name may be specified to be used as a filter for command's output
- (NAM_LN) = Length of name (1-254)
 - (FIL_TY) = File type literal
 - (NAM_AD) = Address of name (0-65535)
- 6.39.3 Output data:
- (HD_0D) = Number of files catalogued
 - Cumulated length of catalogued files
 - (HD_0F) = lower word
 - (HD_11) = upper word
- 6.40 Format logical drive (#40).
- 6.40.1 Action: Format logical drive
- Does not apply to file devices
- 6.40.2 Input data:
- (A) = Drive number (1-255)
 - (STR_NR) = Cluster size: 2, 4, 8 or 16
 - (DEV_LT) = Device type literal or device code
- Or
- (A) = 0
 - (STR_NR) = Number of reserved logical drives
 - (DEV_LT) = Device type literal or device code
- 6.40.3 Output data: None
- 6.41 Clear Screen (#41)
- 6.41.1 Action: Same as the extended BASIC 'CLS #' Command
- 6.41.2 Input data: None
- 6.41.3 Output data: None
- 6.42 Close all streams (#42)
- 6.42.1 Action: Same as the extended BASIC 'CLEAR #' command
- 6.42.2 Input data: None
- 6.42.3 Output data: None
- 6.43 Peripheral Module Command (#43)
- 6.43.1 Action: Sends a command to the server or the peripheral controller and prints the response as hex-dump
- 6.43.2 Input data:
- (BC) = Length of command string
 - (HL) = Address of command string
 - ZR = Server - /Peripheral module
 - CY = Print hex-dump
- 6.43.3 Output data: None
- 6.44 Get absolute path (#44).
- 6.44.1 Action: Get parameters of absolute path name
- 6.44.2 Input data:
- (DRV_NR) = Drive number (1-255)
 - (DEV_LT) = Device type literal or device code
- Relative path name
- (NAM_LN) = Length of filename (1-254) or Zero for the current directory
 - (NAM_AD) = Address of filename (0-65535)
- 6.44.3 Output data: Absolute path name in BUFF_3 (3.5.1)
- (HL) = Address of absolute file name
 - (BC) = Length of absolute file name
- 6.45 Get first directory list (line) (#45).
- 6.45.1 Action: Get the first directory list or list line specified 5.4.7 and 5.4.8
- 6.45.2 Input data:
- (A) = List type, only for file devices
 - = 0 Regular file list
 - = 1 with BASIC header information
 - = 2 Long name list line
 - (DRV_NR) = Drive number (1-255)
 - (DEV_LT) = Device type literal or device code
- Relative path name
- (NAM_LN) = Length of directory name (1-254) or Zero for the current directory
 - (NAM_AD) = Address of directory name (0-65535)
- 6.45.3 Output data: First directory list/line in BUFF_3

- (HL) = Address of first directory list/line
- (BC) = List/line length
- (DE) = 0, for the root directory

6.46 Get next directory list (line) (#46).
 6.46.1 Action: Get the next directory list or list line,
 6.46.2 Input data: none
 6.46.3 Output data: Next directory list/line in BUFF_3
 - (HL) = Address of next directory list/line
 - (BC) = List/line length

6.47 Print integer (#47).
 6.47.1 Action: Print the 3-byte integer (A) (DE)
 on 3, 6 or 8 digits with leading spaces
 6.47.2 Input data:
 - (DE) = Lower bytes
 - (A) = Upper byte
 - The flags specify the number of digits and the
 count & add option:

ZR	CY	Width	Used for
reset	reset	3	Drive number
reset	set	6	Address
set	reset	8	File length
set	set	8	File length with count & add

6.47.3 Output data:
 If both ZR and CY are set, the printed
 numbers are counted in (HD_0D) and their
 sum is calculated in (HD_0F and HD_11)

6.48 Send command only (#48)
 6.48.1 Action: Send only a command code to the peripheral
 controller.

6.48.2 Input data:
 - (A) = Command code
 - (HL) = Response address (0-65535)

6.48.3 Output data:
 - CY = Error
 - (A) = Error code or 0 if no error
 - (HL) = Response address

6.49 Send data command (#49)
 6.49.1 Action: Send and receive data to/from the
 peripheral controller.

6.49.2 Input data:
 - (A) = Command code
 - (HL) = Data block address (0-65535)
 - (BC) = Length of data block (0-512)
 - (DE) = Response address (0-65535)

6.49.3 Output data:
 - (A) = Error code
 - (HL) = Response address
 - (BC) = Response length

6.50 Get device size (#4A)
 6.50.1 Action: Get the parameters of the current device
 6.50.2 Input data: None
 6.50.3 Output data:
 - (H) = Available volume size, in logical drives
 - (L) = Number of reserved logical drives
 or #FF for RAM drive and file devices
 - (BC) = Actual SD card size, in logical drives

6.51 Serial port Receive (#4B)
 6.51.1 Action: Receive from RS 232 port into buffer.
 The function times out after 1 second,
 returning 0 bytes

6.51.2 Input data:
 - (C) = Number of bytes (1-256)
 - (HL) = Buffer address (0-65536)

6.51.3 Output data:
 - (C) = Number of bytes received (0-240)

6.52 Serial port Transmit (#4C)
 6.52.1 Action: Transmit to RS 232 port from buffer.
 The function times out after 12 seconds,
 signaling: 'Communication error'

6.52.2 Input data:
 - (BC) = Number of bytes (1-512)
 - (HL) = Buffer address (0-65536)

6.52.3 Output data: None

6.53 Read line from "M" channel (#4D)

6.53.1 Action: Read a line, terminated with a CR (#0D), if CY is reset, or otherwise a block of maximum (BC) bytes from the "M" channel at (IX) to the buffer at (DE)

6.53.2 Input data:

- (IX) = Channel address
- (BC) = Maximum number of bytes (0-65536)
- (DE) = Buffer address (0-65536)
- CY = Do not check for line end (CR)

6.53.3 Output data:

- Zero flag set if (BC) = 0
- (BC) = Number of bytes read (1-65536)

6.54 Write line to "M" channel (#4E)

6.54.1 Action: Write a line, terminated with a CR (#0D), if CY is reset, or otherwise a block of maximum (BC) bytes from the buffer at (HL) to the "M" channel at (IX)

6.54.2 Input data:

- (IX) = Channel address
- (BC) = Maximum number of bytes (0-65536)
- (HL) = Buffer address (0-65536)
- CY = Do not check for line end (CR)

6.54.3 Output data: None

6.55 Check device presence (#4F)

6.57.1 Action: Check if device (A) is present

6.55.2 Input data:

- (A) = Device code

6.55.3 Output data:

- ZR = Device is present
- CY = Block device

6.56 Ethernet Receive (#50)

6.56.1 Action: Receive a block of maximum (C) bytes from station (A). Time out after 12 second

6.56.2 Input data:

- (A) = Station number (1-7)
- (C) = Number of bytes (1-255)
- (HL) = Buffer address (0-65536)

6.56.3 Output data:

- CY = time-out
- (C) = Number of bytes received (0-255)

6.57 Ethernet Transmit (#51)

6.57.1 Action: Transmit a block of maximum 255 bytes to station (A). Time out after 12 second

6.57.2 Input data:

- (A) = Station number (1-7)
- (C) = Number of bytes (1-255)
- (HL) = Buffer address (0-65536)

6.57.3 Output data:

- CY = time-out

6.58 Get long name (#52)

6.58.1 Action: Get the long name of the last found file on a file device or the contents of the last found 'LNF' auxiliary file on a block device, in the buffer (HL)

6.58.2 Input data:

- (HL) = Buffer address (0-65536)

6.58.3 Output data:

- CY = Error - no name returned