

Getting started guide

Interface 1bis version 4b

Starten des IF1bis:

1. Pushbutton fuer > 1.2 s halten -> O-LED = on
 2. Pushbutton fuer < 0.5 s tippen -> B-LED = on
 3. Fertig.
- Wenn in 2. der Button zulange gehalten wird, kommt es zu Fehlfunktionen! Dann neuer Versuch oder
4. Pushbutton fuer >2,5 s halten erzeugt Neuladen OS

1. Inspection

The 'Interface 1bis' is shipped in a corrugated cardboard box, sealed in an ESD-safe bag and bubble-wrapped. Upon receipt it should be first checked for possible gross mechanical damage incurred during transport and not accepted if broken. The delivery should also be rejected if the ESD-safe bag was opened and / or the blister pack containing the SDHC card is missing.

After removal from the ESD-safe bag the interface is best left for half an hour at room temperature, to allow eventual condensed moisture to evaporate.

2. Connecting to the ZX Spectrum

2.1 If the interface is fitted with a 'Spectrum model selector' header, the two jumpers should be set to the appropriate position:

- '48' (1-2, 3-4), for 'one ROM' machines (48k, 128, +2) or respectively
- '+3' (2-3, 4-5), for 'two ROMs' machines (+2A, +2B, +3).

2.2 Before connecting (or disconnecting) the 'Interface 1bis', it is compulsory to **remove the plug** of the power supply unit from the ZX Spectrum's (9V) power socket, in order to prevent **serious damage**.

2.3 The interface must be first aligned, by moving it sideways until the guide, in position 5 of its edge receptacle, engages the slot of the ZX Spectrum's edge connector and then, sufficient force needs to be applied on the interface for it to fully slide and snap into place.

2.4 It has been observed that plugging and unplugging the 9V supply into/from the Spectrum's power socket can occasionally produce spikes large enough to corrupt the contents of the interface's non-volatile memory. It is therefore **strongly recommended to plug the Spectrum's power supply into a switch mains socket** and to use this one for turning the power on and off.

2.5 When power is applied, the leftmost (U) LED of the interface lights up for 2 seconds, indicating the initialization stage of the microcontroller. The ZX Spectrum should **boot up normally**. If it does not, then the power must be **immediately disconnected** and before any further power-up attempts are made, the edge connector and the interface need to be thoroughly cleaned of any loose **metallic particles**, using a toothbrush dipped in isopropyl alcohol.

2.6 After power-up the 'Interface 1bis' enters its '**OFF**' state, in which it behaves as if it was **not present**, apart from drawing some 30mA from the Spectrum's internal regulated 5V supply.

3. Controls and states

3.1 The **pushbutton** performs four different functions, depending on the duration of the push:

- ON/OFF If pressed for longer than 1.2 seconds, it toggles the state of the interface between 'OFF' and 'ON Inactive'.
- RESET If pressed for 0.5 to 1.2 seconds, it produces a 5ms-wide, active-low pulse on the ZX Spectrum's /RESET line.
- NMI If pressed for less than 0.5 seconds, it produces a 5ms-wide, active-low pulse on the ZX Spectrum's /NMI line.
- SYSLD If pressed for longer than 2.5 seconds the interface reloads its operating system from a file server or the SD card.

3.2 The six **control LEDs** of the interface indicate (see also section 9):

- LED 'O' the 'ON/OFF' state of the interface
- LED 'B' the interface's 'BASIC ROM' is paged-in
- LED 'S' the interface's 'Shadow ROM' is paged-in
- LED 'M' when on: the mouse is active as 'Kempston' mouse
when blinking: the mouse is active as 'Kempston' joystick
- LED 'C' the SD card has been identified
- LED 'U' the connection to the server has been established

3.3 The 'OFF' state

The 'OFF' state is indicated by the 'O' LED being off. The interface enters this state after power-up or at any time, if it is 'switched off' using the pushbutton.

In this state the interface **does not respond** to any I/O requests and implicitly cannot page memory. Besides the pushbutton, no other functions of the interface are available.

3.4 The 'ON Inactive' state

The 'ON' state is indicated by the 'O' LED being on while both 'B' and 'S' LEDs are off. The interface can be switched from 'OFF' to 'ON' (and vice-versa) by pressing the pushbutton for **longer than 1.2 seconds.**

In this state the interface **does respond** to I/O request, but its RAM is not paged in. The ZX Spectrum is running the '48k BASIC', from its internal ROM and the joystick and mouse ports are operational.

3.5 The 'Active' state

The 'Active' state is indicated by either the 'B' or 'S' LEDs being on. The interface switches from the 'ON Inactive' state to 'Active' when a NMI is triggered, by pressing the pushbutton for **less than 0.5 seconds.** Switching **back** from the 'Active' state is only possible to the 'OFF' state, by pressing the push-button for longer than 1.2 seconds.

There are two main reasons why the ZX Spectrum could **crash or freeze** after the interface switches to the 'Active' state:

- bad contact, the remedy of which is cleaning the edge connector and

So kurz wie
möglich!!!
Sonst nur Aufblitzen

- corrupted SRAM contents, which requires the reloading the interface's operating system, described at point 6.

← Oder: Pushbutton von 1. zu 2. zulange

In the 'Active' state the interface is fully operational and the ZX Spectrum is running the 'Extended BASIC' from the interface's on-board non-volatile RAM, while the 'B' and 'S' LEDs are indicating which of the 'BASIC ROM' or respectively 'Shadow ROM' is currently paged-in.

4. Using the SD card

4.1 Although the 'Interface 1bis' is equipped with a push-push type of micro SD card slot it may be perhaps advisable to use it in push-pull mode, just for the sake of exerting less pressure on the edge connector.

4.2 The SD card may be inserted or extracted at any time, **but it is nevertheless recommended to insert it only when the interface is not powered.**

After being inserted, the SD card is sensed by the interface's microcontroller, which then proceeds to query it, and turns the 'C' LED on, if the identification was successful. The microcontroller also turns the 'C' LED off when it senses that the SD card was removed from the slot.

4.3 Before the first use of a new card or when changing the SD card with another one of a different size, or just for checking the type and size of the current card, the:

```
FORMAT "m";0
```

command should be issued.

The command:

```
FORMAT #r;"m";0
```

reserves `r` (0 .. half the current volume's size) logical drives, creating a partition for large, sequential-access files, inaccessible to the interface's operating system.

4.4 The SD card doesn't need to be otherwise partitioned, as it is implicitly subdivided into 'volumes', comprising 255 logical drives each, except for the last one, which may have less.

All logical drives are of the same size: 32MB and need to be formatted, before they can be accessed, using the command:

```
FORMAT "m";d
```

where `d` is a number between 1 and the highest drive number for the corresponding 'volume', depending on its size.

For cards larger than 8 GB, the 'volume' may be specified as a letter (A to O) following the device literal "m":

```
FORMAT "me";d
```

The allocation unit (cluster) size can be individually set for each logical drive, when it is formatted using the syntax:

```
FORMAT #c;"m";d
```

where `c` is a number in the set (2, 4, 8, 16).



4.5 The command for producing the catalogue of a logical drive is:

```
CAT "m";d or  
CAT d , if the current device is "m", or just  
CAT , for the current drive.
```

The file names appear in the first column of the CAT command's output, while the second column shows the corresponding 'type literals':

- P BASIC program (.ZZP)
- N Number array (.ZZN)
- A String array (.ZZA)
- C 'CODE' block (.ZZC)
- F 'PRINT' file (.ZZF)
- X Text file (.TXT)
- B 'Binary' file (.ZZB)
- S Screen dump (.SCR)
- T Tape file (.TAP)
- Z Snapshot (.Z80)
- D Directory

```
Catalog of Drive 3 Dev MA  
Directory  
COPY P 439  
GENS 113992 25000  
HR 131 23298  
JOY 642  
MONS 66556 55000  
mouse  
TAP COP 298 1  
URAM C 18 23298  
URAM S 118  
BASIC  
DOCS  
SRC  
SYS  
COPYTREE P 1553  
SYSLOAD P 724  
11 Files 22231 Bytes  
Free Sectors 7550 * 8  
CAT 3L
```

The third column displays the file size, and the fourth:

- the start line number of BASIC programs,
- the array's name or respectively
- the loading address of 'CODE' blocks.

The free space on the logical drive is given as number of allocation units (clusters).

5. Connecting to a server

COM-Port-Nummer
muss einstellig sein!!

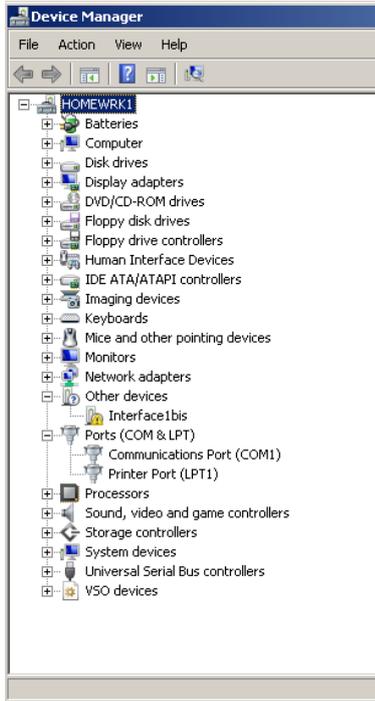
5.1 The interface's microcontroller USB firmware implements the 'Communication Device Class' 'Abstract Control Model' subclass (CDC ACM), emulating an asynchronous serial device, like an USB modem. Although the corresponding 'usbser.sys' driver is part of the MS Windows distribution, it still needs to be installed and associated with the 'Interface 1bis'. The '**IF1bis.inf**' file, required for this purpose, is available at the 'Interface 1bis' site.

5.2 When the 'Interface 1bis' is connected for the first time to a USB port of the server machine it appears in the 'Device Manager' under 'Other devices' with a yellow exclamation mark (5.2.1), indicating that it is not functional.

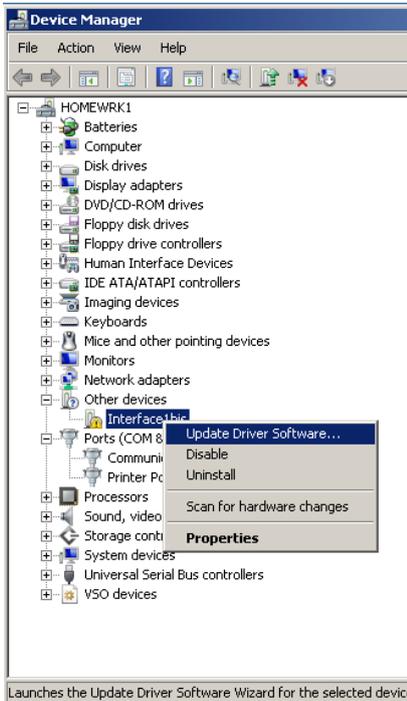
To **install the driver**:

- Select 'Update Driver Software' from the right-click menu (5.2.2)
- Select 'Browse my computer for driver software' (5.2.3)
- Browse to the location of the '**IF1bis.inf**' file (5.2.4)
- Select 'Install this driver software anyway' (5.2.5) (The publisher is Microsoft but Windows can't verify it!! So typical.)
- When done (5.2.6), right-click the 'Interface1bis' icon, now under 'Ports', select 'Properties' and 'Advanced' under the 'Port Settings' tab and change the allocated COM port number to the lowest possible one (like COM3 or COM4), not associated with a physical port or a permanently connected USB device, even if it's marked 'in use' (5.2.7).

5.2.1

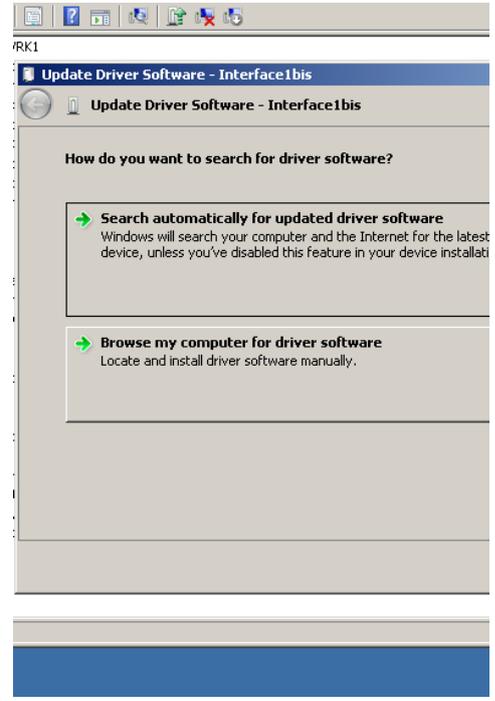


5.2.2

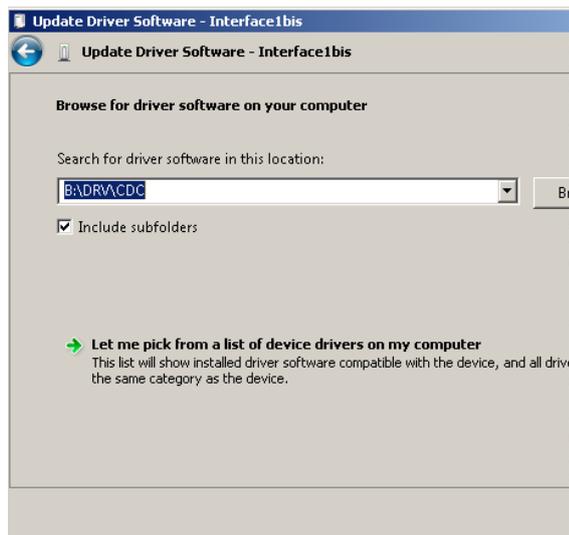


Launches the Update Driver Software Wizard for the selected device

5.2.3



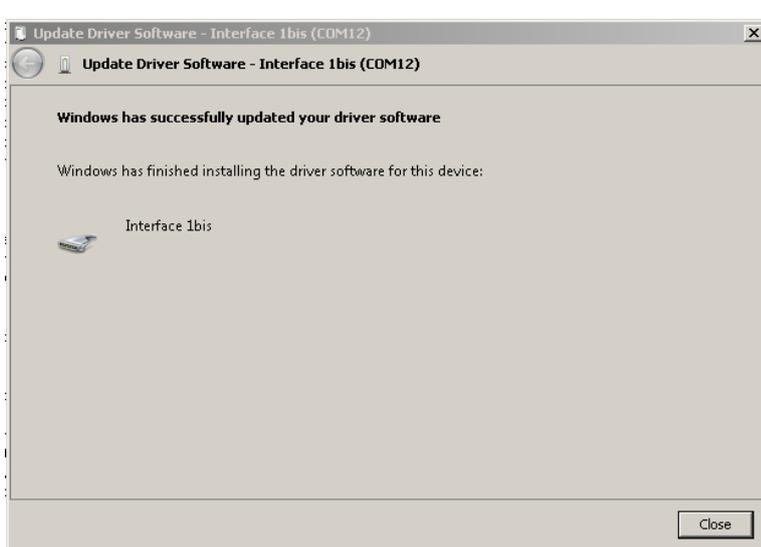
5.2.4



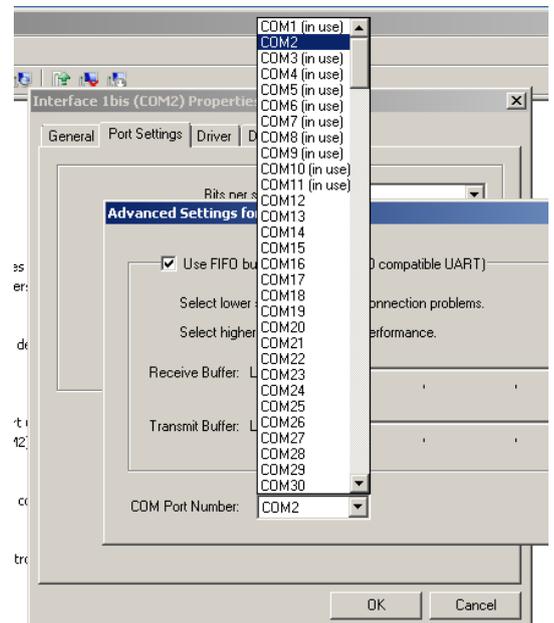
5.2.5



5.2.6

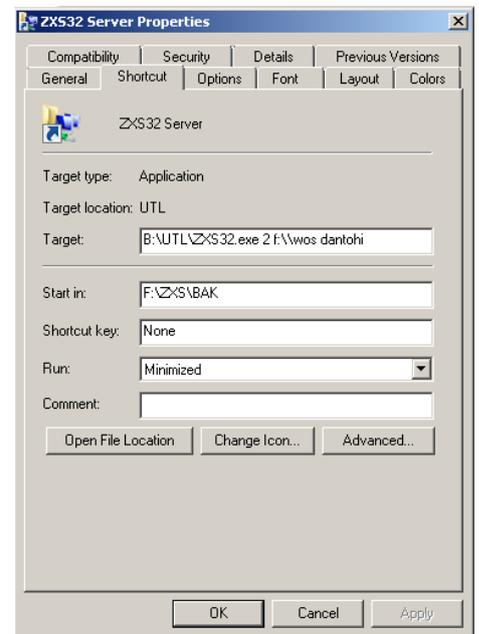


5.2.7

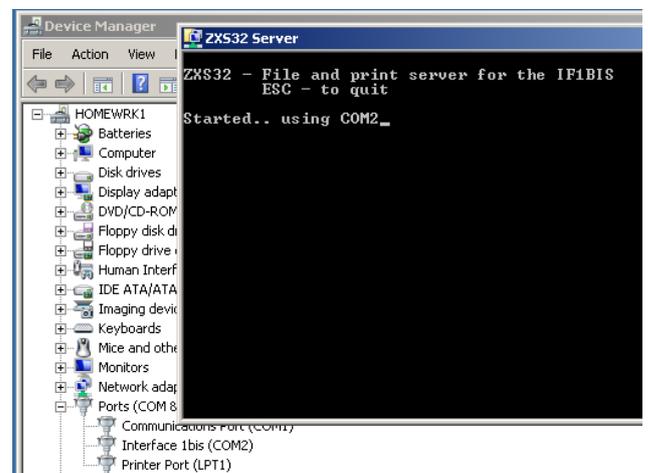


5.3 In order to communicate with the 'Interface 1bis' the server machine needs to be running the 'ZXS32.exe' Win32 console **server applet**, available at the 'Interface 1bis' site, in compressed form as 'ZXS32.ZIP'.

The 'Target' command line in the applet's shortcut **must** include, as first parameter, the COM port number, allocated to the 'Interface 1bis'. The second parameter may specify an **existing** local working directory for the WOS game launcher (default is c:\WOS) and the third, a WOS login username (default is: user_0001). The shortcut may also set the 'Start in' directory, which will be the default directory of the 'Server' device "v" from the ZX Spectrum's perspective.



5.4 After the applet has identified the interface, it displays the message: 'Started.. using COMx', where 'x' is the number of the COM port allocated to the 'Interface 1bis' in the 'Device Manager'. If this does not happen, the USB cable needs to be disconnected and reconnected after a couple of seconds.



5.5 Once the 'Interface 1bis' can communicate with the server applet the 'U' led is turned on and the 'Server' device "v", comprising all logical drives of the server machine, mapped as directories, becomes available. Thus, the command to produce the catalogue of the 'system32' folder would be:

```
CAT "v";d;"/c/windows/system32/"
```

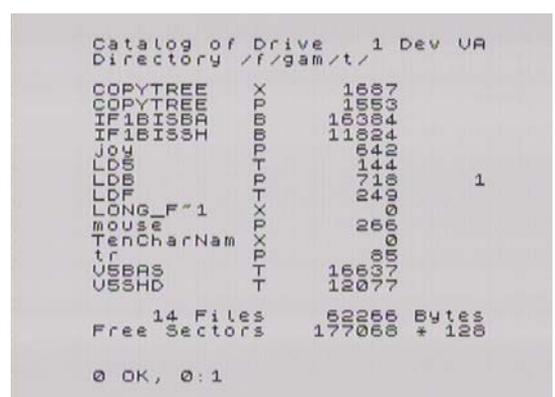
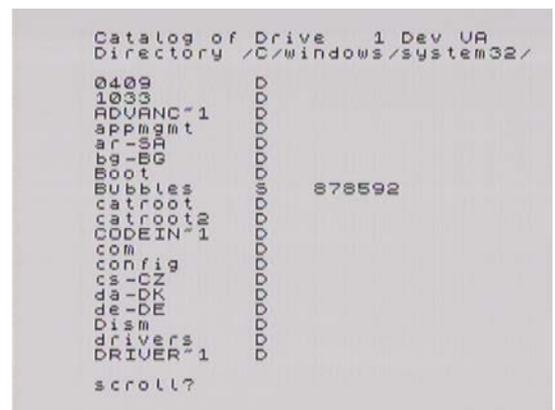
or also:

```
CAT "v";d;"c:/windows/system32/"
```

where the 'drive number' **d** is irrelevant and can be any number in the range 1-255.

Only the file types that the ZX Spectrum can handle, as enumerated at point 4.5, are accessible on the 'Server drive'.

Using '**long filenames**' is possible but not meaningful, as they are abbreviated in '8.3' format, ending up containing less information than a normal 10-character name.



6. Loading the operating system

6.1 Restoring the interface's operating system is necessary after an accidental corruption. In such a case the ZX Spectrum crashes or freezes when the interface switches to the 'Active' state and therefore the required data has to be loaded from an audio source, via the 'EAR' port.

For this purpose, the two parts of the operating system: the 'BASIC ROM' and the 'Shadow ROM') are available at the 'Interface 1bis' site as auto-loading tape files, already converted to sound, in compressed .WAV format: '**V5BAS.ZIP**' and '**V5SHD.ZIP**'.

6.2 The interface must be first switched to the '**ON Inactive**' state, by pressing the pushbutton for longer than 1.2 seconds, after which each of the two operating system parts can be loaded by issuing a:

`LOAD ""`

command and playing the corresponding .WAV files on any device that can drive the Spectrum's 'EAR' port.

After the loading is completed without errors, the 'B' or the 'S' LED will blink shortly, indicating that the 'BASIC ROM' or respectively the 'Shadow ROM' has been transferred to the on-board non-volatile static RAM.

Auch nach Batteriewechsel!

6.3 A much faster method implies loading the operating system images: '**IF1BISBA.BIN**' and '**IF1BISSH.BIN**', available at the 'Interface 1bis' site, directly from a file server, over the USB port, using a very short loader, already converted to sound, in compressed .WAV format: '**LD5.ZIP**'.

First, the two operating system images need to be downloaded to the '**c:\tmp**' folder of the server machine, which must be created if it does not already exist.

The 'Interface 1bis' should be brought in the '**ON Inactive**' state, by pressing the pushbutton for longer than 1.2 seconds and then **connected to a USB port** of the server machine, as described at point 5.

Once the 'U' LED is on, the operating system is instantaneously loaded by issuing a:

`LOAD ""`

command and playing the '**LD5.WAV**' file on a device that can drive the ZX Spectrum's 'EAR' port.

6.4 The method described above (at point 6.3) can be used even in the absence of a suitable audio device for playing the loader .WAV file, by typing-in and running the BASIC loader shown in the listing or the text file: '**IF1bis4bSL.txt**' available at the 'Interface 1bis' site.

6.5 If the operating system just needs to be updated, without being corrupted, the procedure as described at point 6.3 should be followed, with the difference that the loader doesn't need to

```
100 DATA 243,22,2,211,127,211
110 DATA 191,211,2,255,33,0,0,1
120 DATA 159,4,6,2,199,211,159
130 DATA 175,16,6,2,251,219,95,203
140 DATA 119,40,2,50,6,4,175,219
150 DATA 159,16,6,2,51,254,2,32,11
160 DATA 62,16,6,2,32,237,17,6,11
170 DATA 32,249,6,2,4,218,211,191
180 DATA 21,32,2,210,211,127,211
190 DATA 63,251,201
200 FOR i=0 TO 60: READ b
210 POKE 23295+i,b: NEXT i
220 PRINT USR 23296

@ OK, @:1
```

be played over an audio device. Instead, the '**LD5.ZZP**' BASIC program can be run from the file server or the SD card.

6.6 Starting with lot 2 boards, marked IF1BIS4B-02, the operating system can be **re-loaded** from the file server by pushing the button for longer than 2.5 seconds.

6.7 The operating system can also be **re-loaded from the SD card**, when the interface is **not connected** to a server, either by the method described above, or by running the 'LD5' utility, assuming the data was previously copied to the last 64 sectors of logical drive 1, using the '**SYS.ZZP**' BASIC program.

7. The 'Extended BASIC'

This section only contains limited information, strictly necessary to start using the 'Interface 1bis', assuming some familiarity with the original 'Extended BASIC' of the 'ZX Interface 1'.

A comprehensive reference manual: '**IF1bisOpSys.pdf**' is available at the 'Interface 1bis' site.

7.1 Devices and channels

The 'Interface 1bis' can access two different **storage devices**:

- 'Microdrive' device "**m**", using the SD card as storage media and
- 'Server' device "**v**", using disk space on a server as storage media

The 'ZX Network' and 'RS232 port' devices of the original 'ZX Interface 1' have no underlying hardware support.

A '**channel**' is a logical data source/destination, like a specific file on a storage device rather than the device itself. The command:

```
LOAD * "m";4;"screen" SCREEN$
```

specifies the file "**screen**" from the logical drive **4** of the 'Microdrive' device "**m**" as the input 'channel' for loading a screen dump, while the command:

```
OPEN# 5; "v";1;"list"
```

associates stream number **5** with an 'M'-type 'channel', defined as the file:

```
"list" from the current directory of the 'Server' device "v".
```

- 'M' channels, providing buffered sequential file access can use any of the 'Microdrive' "**m**" or 'Server' "**v**" devices.

- '**N**' channels are not supported.

- 'B' and 'T' channels are implemented as **output-only**. The 'B' channel sends binary data directly to the printer, while the 'T' channel behaves identically to the 'Printer' channel 'P'.

7.2 File names

File names can be 1 – 10 characters long, are **case-insensitive** and may be preceded by a **path**. The following examples specify that the file: `Program1` is located in:

<code>"Program1"</code>	the current directory
<code>"../Program1"</code>	the parent directory of the current directory
<code>"/Program1"</code>	the root directory
<code>"subdir/program1"</code>	the subdir subdirectory of the current directory
<code>"../dir/Program1"</code>	the dir subdirectory of the parent directory
<code>"/folder/program1"</code>	the folder subdirectory of the root directory

File names may contain the wildcards: `?` and `*`, standing for 'any character' and respectively 'any number of characters'.

File names may also have their one-character 'type literal', enumerated at point 4.5, appended as an extension:

<code>"Game1.t"</code>	would be a 'tape' file
<code>"list.f"</code>	would be a 'PRINT' file

Specifying file extensions is meaningful only in a few special situations, mainly for 'emulator' files or when the target directory contains files with the same name but of different types.

7.3 Directories

A name ending in a 'slash' character `/` designates a directory. The operations that can be performed on directories are:

- Creating a directory, using the **SAVE** command:

```
SAVE * "m";2;"newdir/"
```

creates the **newdir** subdirectory in the current directory of drive **2**.

- Changing a directory, using the **LOAD** command:

```
LOAD * "v";1;"f:/games/"
```

makes the **games** folder of the server's **f** drive current.

- Deleting a directory, using the **ERASE** command:

```
ERASE "m";1;"temp/empty/"
```

deletes the **empty** subdirectory from the **temp** directory of drive **1**.

Only empty directories that are not current can be deleted.

7.4 Repetitive commands

The commands **ERASE** and **MOVE** act repeatedly on all files that match the specified name, as opposed to all other commands, which operate only on the first matching file.

If the current directory contains both a BASIC program file called **test** and a 'tape' file with the same name, then the command:

```
ERASE "m";1;"test"
```

will delete both of them. If only the BASIC program needs to be deleted then the file type has to be specified, by means of a name extension:

```
ERASE "m";1;"test.p"
```

Likewise, the command:

```
MOVE "m";1;"t*" AT "m";1;"s*"
```

will change the initial letter **t** to **s**, in the name of any file present in the directory.

7.5 Emulator files

The 'input tape' and the 'output tape' are assigned by using the `LOAD` or respectively the `SAVE` statements while explicitly specifying the file type using the filename extension `.t` or the command option `t`.

```
LOAD *"m";12;"/games/manicmin" t or  
LOAD *"m";12;"/games/manicmin.t"
```

will assign the `manicmin` file in the `games` directory of drive `12` as the 'input tape'. The game can then be loaded by issuing a

```
LOAD ""
```

command. Alternatively, the type `T` (capital) may be used to assign the 'input tape' and perform the equivalent of `NEW` followed by `LOAD ""`, in one command:

```
LOAD *"m";12;"/games/manicmin" T
```

The 'input tape' and 'output tape' assignments remain valid until the 'output tape' has been read to its end, the 'output tape's' size exceeds 16 MB or the tape files are explicitly unassigned using the `LOAD` or respectively the `SAVE` command with the option: `STOP`. Thus, the command:

```
SAVE STOP
```

'closes' the 'output tape'.

In order to create a snapshot, a **new** file must be first assigned, using the `SAVE` statement while explicitly specifying the file type using the filename extension `.z` or the command option `z`. The command:

```
SAVE *"m";1;"/temp/snap" z or  
SAVE *"m";1;"/temp/snap.z"
```

creates the file `snap` in the `temp` directory, to which the snapshot will be saved when a NMI is triggered, by pressing the button for less than 0.5 seconds, after which the corresponding file is unassigned as a snapshot destination.

Snapshots can be loaded using the `LOAD` statement while explicitly specifying the file type using the filename extension `.z` or the command option `z`:

```
LOAD *"m";12;"/games/monty" z or  
LOAD *"m";12;"/games/monty.z"
```

will load the `monty` snapshot from the `games` directory on drive `12`.

Using the type `Z` (capital) will switch the 'Interface 1bis' to the 'ON Inactive' state after the snapshot is loaded.

Loading, saving or verifying 'screen dumps' also requires that the file type is explicitly specified, using the filename file name extension `.s` or the command option `s`:

```
LOAD *"m";1;"scrn" s or  
LOAD *"m";1;"scrn.s"
```

7.6 Abbreviated syntax

When a 'device' or 'drive number' are specified in a command, they become 'current' and may be omitted in a subsequent command.

After the command:

```
CAT "m";1
```

the `screen1` screen-dump can be loaded from drive `1` of device `"m"` using the abbreviated syntax:

```
LOAD ;"screen1" s
```

and the program `Test` from drive `3` of the same device, using:

```
LOAD 3;"Test"
```

8. Using the mouse

8.1 Connecting the mouse

The mouse must be connected only while the interface is **not powered**. This is a feature of the PS/2 mouse rather than of the interface.

At power-up, and only then, the interface's microcontroller checks for the presence of a mouse and switches the 'M' led on, if one is found.

8.2 Mouse states

The state of the mouse is indicated by the 'M' LED:

- Off The mouse is not present or is disabled
- On The mouse is enabled as Kempston mouse
- Blinking The mouse is enabled as Kempston joystick. The LED blinks at half the sampling rate.

While the mouse is enabled, the **joystick port is disabled** and vice-versa.

The mouse can be disabled by pressing both buttons simultaneously.

While disabled, the mouse can be enabled as:

- Kempston mouse, by pressing the right button, or as
- Kempston joystick, by pressing the left button.

8.3 Mouse modes

The mouse is, by default enabled in its 'windowed' mode, in which the reported coordinates are contained within a configurable-sized window, but can also function in Kempston mouse 'legacy' mode, with the coordinates wrapping around from 0 to 255 and vice-versa.

The command `CAT 0` can be used to change the mouse mode to:

- legacy mode: `CAT 0; "mle"`
- windowed mode: `CAT 0; "mwi"`

9. Ethernet module

9.1 Connection

The pin headers J7 and J8 can be used to attach an Ethernet controller module, via a special SPI cable. While this connection is in place, the function of LEDs 'C' and 'M' changes to:

- LED 'C' Ethernet module communication
- LED 'M' Ethernet module interrupt

