# ‘Interface 1bis’ for the Sinclair ZX Spectrum Getting started

04 October 2017

## 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.

## 2. Connecting the interface to the ZX Spectrum

2.1 The two jumpers of the ‘Spectrum model selector’ header 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 power socket, in order to prevent serious damage.

2.3 Unless the ZX Spectrum’s edge connector is in perfect condition, it is strongly recommended to apply some electrical contact cleaner/lubricant, like the ‘Electrolube EML’. The fact that other interfaces work well on that same machine is irrelevant. Actually it is exactly the very hard edge receptacle contacts of other devices that, by cutting grooves in the ZX Spectrum’s board edge, create difficulties for the ‘Interface 1bis’, which has much softer contacts. Applying a lubricant solves this problem and also reduces wear and tear.

2.4 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 exerted on the interface for it to fully slide into place, in one continuous move.

2.5 At power-up, the leftmost (‘U’) LED of the interface lights up for two 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 the edge connector needs to be thoroughly cleaned of any loose metallic particles, using a toothbrush dipped in alcohol.

2.6 After power-up the ‘Interface 1bis’ enters its ‘OFF’ state, in which it does not interact in any way with the ZX Spectrum, apart from drawing some 30mA from the 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 server machine or the SD card.

3.2 The six control LEDs of the interface indicate:

- 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’ serial SRAM chip selected, if installed, otherwise:
  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’ 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 while 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.
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. Preparing the SD card

4.1 An ‘Interface 1bis’ SD card cannot be directly accessed and therefore appears to be empty under Microsoft Windows®, because it uses a proprietary block device file system, optimized for the Z80 processor.

4.2 A ready-for-use SD card can be optionally purchased together with the ‘Interface 1bis’ or otherwise, an empty 4GB / 8GB card has to be prepared as described below.

4.3 Insert the SD card in the PC’s reader and change its automatically assigned drive letter to **S:**, by running **diskmgmt.msc** and selecting Change Drive Letter and Paths.. from the right-click menu of the corresponding volume.

4.4 Download **LDC_setup.exe** from the software section of the ‘Interface 1bis’ official website and run it, in order to install the ‘Logical Disc Copier’.

4.5 Download **ZXS_setup.exe** and run it. Un-tick the ‘Launch the Server Applet’ box on the last page of the installer.

4.6 Run ‘Prepare New Card’ from the ‘IF1bis\Logical Disk Copier' start menu.

4.7 Copy folders, not larger than an ‘Interface 1bis’ logical disk’s maximum capacity of 32MB, from the PC to the SD card, by selecting: Send to > Logical Disk Copier, from their right-click menu and then specifying the corresponding logical disk numbers at the prompt. Only the ZX Spectrum related files types enumerated at point 9.3 are copied.

# 5. Starting the interface

5.1 After power-up the interface needs to be activated, by pressing the button until the rightmost LED (‘O’) lights up and then again, very briefly, for less than 0.5 seconds, so that the ‘B’ LED also lights up. On a 128k ZX Spectrum, a manual reset may be necessary, to bring up the 48k copyright message.
Entering the BASIC command `RUN` starts the GBL ‘Game Browser / Launcher’, a mouse / joystick driven user interface, specifically designed for advanced game playing and file handling.

5.2 Whenever the SD card is changed, it is necessary to identify it, by executing the extended BASIC command: `FORMAT 0`.

# 6. Restoring the operating system

6.1 When using the ‘Interface 1bis’ for the first time, it is possible that the ZX Spectrum crashes or freezes after switching to ‘Active’ mode.
Apart from a bad edge connector contact, this can be caused by the contents of the onboard non-volatile memory being corrupted.
Pressing the button for 2.5 seconds normally solves this issue, but if it does not, than the operating system was totally erased, most probably due to X-Ray scanning during transport and needs to be reloaded, from a prepared SD card, by switching the interface to the ‘ON’ state, entering the command: `LOAD ""` in ‘48k BASIC’ mode, and playing the ‘OpSys Loader’, from the ‘IF1bis/IF1bis Server’ start menu, through a suitable audio signal amplifier, to the ZX Spectrums ‘EAR’ input.

6.2 The ‘Interface 1bis’ can also act as an audio signal source, if equipped with the ‘EAR cable’, which can be purchased as an optional accessory. Issuing the commands: `OUT 159,243: LOAD ""` in ‘48k BASIC’ mode, while the interface is in the ‘ON’ state, restores the operating system, by rendering the ‘LD5’ program, stored on the SD card, as audio signal, to the ZX Spectrum’s ‘EAR’ input.

6.3 For ZX Spectrum models that do not have an ‘EAR’ socket the following operating system loader must be entered in ‘48k BASIC’ mode and run, with the ‘Interface 1bis’ in ‘ON’ state:

```
100 DATA 243,211,127,211
105 DATA 191,211,255,33
110 DATA 0,0,17,64
115 DATA 91,1,159,8
120 DATA 26,19,237,121
125 DATA 16,250,175,219
130 DATA 95,203,119,40
135 DATA 249,6,4,27
140 DATA 175,219,159,16
145 DATA 250,62,16,6
150 DATA 32,237,178,61
155 DATA 32,249,26,60
160 DATA 18,254,224,40
170 DATA 206,254,248,32
175 DATA 209,211,127,211
180 DATA 63,211,255,199
185 DATA 248,0,4,0
190 DATA 192,255,0,0
200 FOR i=0 TO 71: READ b
210 POKE 23296+i,b: NEXT i
220 PRINT USR 23296
```

6.4 After restoring the interface’s operating system it is necessary to issue the extended BASIC commands: `FORMAT 0` and `FORMAT !0`.

6.5 Restoring the operating can fail only because of bad electrical contact at the edge connector, which has to be addressed before taking any further action.

## 7. Connecting to a server

7.1 A distinctive feature of the ‘Interface 1bis’ is its full-speed USB port, dedicated for client-server networking, over a standard mobile phone cable.
The server machine can be any PC, running the **ZXS32.exe** ‘IF1bis Server’ applet, under Microsoft Windows® XP, or later.

7.2 The applet is installed by running **ZXS_setup.exe**, available at the ‘Interface 1bis’ official website. When started for the first time, the ‘IF1bis Server’ applet automatically installs the required USB port driver.
When the USB connection is established, the interface’s ‘U’ LED lights up and the ZX Spectrum’s ‘Extended BASIC’ operating system gains access to the server’s disks, as subdirectories of a virtual ‘Microdrive’ of unlimited size, as well as to its internet connection.

7.3 If equipped with a suitable Ethernet module, the ‘Interface 1bis’ can also connect to one or several servers, running ZXS32.exe, over the internet. The required ‘SPI cable’ is available as an optional accessory.

## 8. Software updates

The server applet, operating system, system utilities and microcontroller firmware can be simultaneously updated by downloading the latest release of **ZXS_setup.exe** and running it on the server machine, followed by loading the BASIC program ‘**UPD**’, from the server’s ‘c:\tmp’ folder, on the ZX Spectrum, using the GBL ‘Game Browser / Launcher’ or issuing the ‘Extended BASIC’ command: `LOAD $;"/c/tmp/upd"`.

# 9. 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 is available at the ‘Interface 1bis’ site.

## 9.1 Devices and channels

The ‘Interface 1bis’ can access three 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
- ‘RAM drive’ device: `"r"`, if the optional serial SRAM chip is installed

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"`, ‘Server’: `"v"` or ‘RAM drive’: `"r"` devices.

- Streams can be opened to a ‘B’ channel, when a suitable cable is attached to the joystick and mouse connectors. Only `FORMAT` and `OPEN#`, followed by: `PRINT#`, `INKEY$#` and `INPUT#` statements are implemented, while `LOAD`, `SAVE`, `VERIFY` and `MERGE` are not.

## 9.2 Abbreviated syntax

As the ‘drive number’ is irrelevant for the devices `"v"` and `"r"`, these can be accessed using the characters:
`!` in place of: `*"r";`<drive> ,(compatible with 128k BASIC syntax) and
`$` in place of: `*"v";`<drive> , like in: `CAT $;"*.t"` , to list all ‘tape files’ from the current server directory.

When a ‘device’ or ‘drive number’ are specified in a command, they become ‘current’ and may be omitted in a subsequent command. Thus, after:

```
      CAT "m";7
```

the `screen1` screen-dump can be loaded from drive `7` of device `"m"` using the abbreviated syntax:

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

Specifying a ‘drive number’ or using an initial `*`, not followed by a device type literal, changes the ‘current device’ to `"m"`. After: `CAT $`,

```
      LOAD ;"Test"
```

will load the program `Test` from the server, but:

```
      LOAD *;"Test"
```

will load it from the current SD card drive, or

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

will load it from drive 3 of the SD card

## 9.3 File types

Following file types are implemented

| Literal | Description | Extension |
|---|---|---|
| P | BASIC program | (.ZZP) |
| N | Number array | (.ZZN) |
| A | String array | (.ZZA) |
| C | ‘CODE’ block | (.ZZC) |
| F | ‘PRINT’ file | (.ZZF) |
| E | Text file | (.ZZE) |
| K | ‘Backup’ file | (.ZZK) |
| B | ‘Binary’ file | (.ZZB) |
| S | Screen dump | (.SCR) |
| X | ZX Tape file | (.TZX) |
| T | Tape file | (.TAP) |
| Z | Snapshot | (.Z80) |

## 9.4 File names

File names are case-insensitive and may be preceded by a path. For devices `"m"` and `"r"`, only the first 10 characters of a name are significant. The following examples specify that the file: `Program1` is located in:

| | |
|---|---|
| `"Program1"` | the current directory |
| `"../Program1"` | the parent directory of the current directory |
| `"/Program1"` | the root directory |
| `"subdir/program1"` | the `subdir` subdirectory of the current directory |
| `"../dir/Program1"` | the `dir` subdirectory of the parent directory |
| `"/folder/program1"` | 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 9.3, appended as an extension:

`"Game1.t"` would be a ‘tape’ file
`"list.f"` 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.

## 9.5 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 2;"newdir/"
```

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

- Changing a directory, using the LOAD command:

```
      LOAD $;"f:/games/"
```

makes the games folder of the server’s f: drive current.

- Deleting a directory, using the ERASE command:

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

deletes the empty subdirectory from the temp directory of drive 1.
Only empty directories that are not current can be deleted.

## 9.6 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 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 1;"test.p"
```

Likewise, the command:

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

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

## 9.7 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.

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

```
      LOAD 12;"/games/manicmin.T
```

The ‘input tape’ and ‘output tape’ assignments remain valid until the ‘input 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’.

Besides regular tape ( .TAP) files, ‘ZX Tape’ files (.TZX) can also be assigned to the ‘input tape’, when using the filename extension `.x`

In order to create a version 1.45 48k snapshot, a new file must be first assigned, using the `SAVE` statement while explicitly specifying the file type using the filename extension `.z`. The command:

```
      SAVE 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, unless the ‘Caps Shift’ key is held until the border stops flashing.
A version 3.05 128k snapshot can be created by specifying the filename extension `.Z` (capital).
Snapshots can be loaded using the `LOAD` statement while explicitly specifying the file type using the filename extension :

```
      LOAD 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`:

```
      LOAD 1;"scrn.s"
```

# 10. Using the mouse

## 10.1 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.

## 10.2 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.

## 10.3 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;"ml"`
- windowed mode: `CAT 0;"mw"`