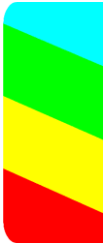




Spectranet logo design by Alistair Cree

Spectranet

User's manual



Ethernet interface for your ZX Spectrum
Connect your ZX Spectrum to the internet!



Developed by Dylan Smith
Manufactured by Ben Versteeg

BYTEDELIGHT

 YouTube



Subscribe!

Disclaimer

This manual and information contained in it is copyright material and may not be reproduced, transcribed, translated or transmitted in any form without the prior written consent of Ben Versteeg from ByteDelight. Do not copy or distribute (parts of) this manual.

Spectranet overview

Finally... The Spectranet will connect your ZX Spectrum to the internet!

With Spectranet and the server side TNFS software - both developed by Dylan Smith - a door to a whole new worldwide set of features for your ZX Spectrum opens up!

In the box

Your Spectranet interface is delivered with a right-angle stereo jack plug adapter and this manual.

The jack adapter is useful for ZX Spectrum models like the Harlequin kit that has a stereo audio output just underneath where the ethernet cable enters the Spectranet interface.

Requirements

You need an ethernet cable (not included) to connect your ZX Spectrum to your local network with internet connection.

You also need a free ethernet port on a network switch or router of your local network.

If there is no free port, obtain a simple network switch that 'splits' one port into multiple ports for more available ports.

Latest firmware

The firmware version when shipping your Spectranet, is R1002.

This release includes:

- C-like debugging facility contributed by desertkun.
- Build system update, so that the source code builds with the latest GNU Z80 binutils.
- Compatibility with 4K erase page flash memory (upcoming issue 3 Spectranet boards)

When there is a firmware update, you can update online by entering this from BASIC:

```
%umount @  
%mount @,"vexed4.alioth.net"  
%load ""
```

Select option 'A'.

Online resources

Make sure to visit to get the most out of your Spectranet! Visit: spectranet.bytedelight.com

Join the Spectranet Facebook community: search for 'Spectranet - ethernet for the ZX Spectrum!'.

What can Spectranet do?

Spectranet has a wide range of features.

The most used feature is visiting TNFS servers, and load games and programs from it.

By default, Spectranet is configured to connect to ByteDelight's TNFS server, and automatically load a welcoming program.

Some other features that are also available are:

- IRC (chat) client
- 2-person chat client
- SpecTank: a cool multiplayer 4-person game!
- And much more to come in the near future!

Find out more about these and other features at spectranet.bytedelight.com

What is TNFS?

TNFS is a communication protocol for Spectranet, developed by Dylan Smith.

TNFS serve a so-called 'filesystem' that can be used to load programs from, with your ZX Spectrum with Spectranet.

You can connect to any TNFS server around the world from your ZX Spectrum with Spectranet!

Some programs will make it feel like you're loading a webpage on your ZX Spectrum!

How do I find a TNFS server?

There is a list of TNFS servers at: spectranet.bytedelight.com

This webpage, launched in 2022, has an actively maintained list of TNFS servers: speccytools.org/index/

You can visit these websites from any webbrowser.

Starting to use your Spectranet

Prepare your ZX Spectrum

Make sure the ZX Spectrum's edge connector is clean!

It's important that your ZX Spectrum has a clean edge connector before connecting any interface. Advised is to use cotton swabs with >95% pure (e.g. isopropyl) alcohol.

Spectranet switch configuration

Make sure to read this paragraph before connecting your Spectranet!

The Spectranet by ByteDelight, has three configuration switches on the bottom of the interface:

- ZX48/128 (default) or ZX +2A/+3: select your ZX Spectrum model with this jumper
- Disable RST8: when set to this position, all BASIC extensions (%-commands) will be disabled
- Disable ALL: when the jumper is put on this position, the Spectranet ROM will be disabled and the computer will boot as if it's not connected - this position is required for firmware updates

Spectranet throughput edge connector

The Spectranet has a throughput edge connector for any device like ZX-AY, ZX-HD and more.

However, other interfaces with their own ROM and/or NMI circuit will collide with the Spectranet ROM. So you can't for example use a DivMMC interface and Spectranet at the same time (with the exception explained in the 'Installing firmware' paragraph).

Connecting the Spectranet to your ZX Spectrum

Make sure your ZX Spectrum is powered off!

Gently slide the Spectranet's edge connector to the ZX Spectrum making sure the 'key' pin on the edge connector is correctly positioned.

Connect your ethernet cable to the Spectranet interface.

Powering up with the Spectranet connected

When you power up your ZX Spectrum with Spectranet connected, after a second it will show a blue screen with information about getting linked to your network and obtaining an IP address – like this example:

This should only take a couple of seconds.

If there is an error, it will show it on screen for a longer time.

By default (from factory), the Spectranet will autoboot to tnfs.bytedelight.com.

The introduction program that will be loaded, will also show how to disable it.

```
Alioth Spectranet winston@adun R1002
Link.>OK
Press BREAK to interrupt.
DHCPDISCOVER
DHCPOFFER
DHCPREQUEST
DHCPACK
Allocated IP address 192.168.001.019
INFS 1.01 initialized
BASIC extensions installed
BASIC streams support initialized
Config extensions OK
Snapshot manager initialized
Mounting 0:tnfs.bytedelight.com
```

USR 0 Mode

All ZX Spectrum models (even 128K models) will enter 48 BASIC in so-called 'USR 0 mode', after the ZX Spectrum with Spectranet has finished the boot sequence.

128 BASIC or +3 BASIC will not be available with Spectranet, because of firmware limitations. However, all 128K games will load just fine.

Autoboot

When a TNFS server has a 'boot.zx' main file, you can use %load "" to easily load that main program. If you have enabled Autoboot (read the %fsconfig section in this manual), the Spectranet will automatically load 'boot.zx' when it powers up, which will make connecting to many TNFS servers, feel like loading a webpage.

Basic use of Spectranet BASIC extensions

Connecting to and disconnecting from a TNFS server

Use this example to connect to a TNFS server:

```
%mount @,"tnfs.bytedelight.com"
```

This example will connect filesystem 0 on your Spectranet to the TNFS server from ByteDelight.com. You have up to 4 filesystems that you can connect to at the same time, which can be pre-configured by using the %fsconfig tool, described in the %fsconfig section in this document.

Use this example to unmount (disconnect) filesystem 0 from the TNFS server:

```
%umount @
```

Listing the files in a folder on the filesystem

Use this command to show the files in the current folder on a filesystem:

```
%cat
```

Changing to another folder

Use this example to change to another folder on the filesystem:

```
%cd "foldername"
```

Use this command to change the current folder to the parent folder on the TNFS server:

```
%cd ".."
```

Loading a program from a filesystem

Use this example to load a file from a connected filesystem:

```
%load "filename"
```

Saving files

If you have write-access on the TNFS server you are using, like when you are hosting your own TNFS server and setting it up, you can use this command to save a file to the connected filesystem:

```
%save "filename"
```

This example will save the file with the name 'filename' to the TNFS server.

Your own TNFS Server

You can quite easily setup your own TNFS server, with the software that Dylan Smith has developed, and which is freely available.

More info can be found at: spectranet.bytedelight.com

Make sure that if you publish your TNFS server to other users, that you:

- Know how to keep your own network with your other computers and network services safe
- Configure the filesystem that is served as read-only access

Setting up your Spectranet

Spectranet network configuration menu

The integrated network configuration menu is launched by pressing the 'NMI' button on your Spectranet interface.

You can also access this network configuration menu with this BASIC command:

```
%i f c o n f i g
```

The configuration menu is used to setup and configure the network settings on your Spectranet.

By default there is not much need to enter this menu yourself, since it is pre-configured when shipped. But there are a lot of advanced features in the NMI configuration menu.

Setting up configuration data wrongly, can brick your Spectranet!

However, if something goes wrong and the Spectranet won't boot correctly anymore, you can reinstall the Spectranet firmware from a storage solution such as a DivMMC interface, as described in the 'Installing firmware' paragraph.

IP address settings

By default, your Spectranet interface is configured to obtain an IP address automatically by the DHCP server in your local network.

The MAC address of a Spectranet interface when shipped, is: AA:00:04:00:00:00.

The hostname of the Spectranet is: Spectranet.

These settings can be changed from within the network configuration menu, as described in the 'Spectranet network configuration menu' paragraph.

Static IP address

The default configuration is to use DHCP to get IP settings.

If you don't have a DHCP server or want to use a static address, change the setting from within the network configuration menu, as described in the 'Spectranet network configuration menu' paragraph.

Installing firmware

In case of a bricked Spectranet ROM, or if you want to update firmware locally, you can use a storage interface such as a DivMMC interface, to load the installer from, for example by connecting the storage interface to the back of the Spectranet interface.

For this, the Spectranet configuration switch must be set to 'Disable ALL', as described in the 'Switch configuration' paragraph.

Configure filesystems with %fsconfig

To enter the filesystem configuration utility, enter:

```
%fsconfig
```

With this utility, you can setup up to 4 so-called 'filesystems', that will be connected by default when you power on your ZX Spectrum with Spectranet:

You can also enable autoboot from this utility. When autoboot is enabled, the Spectranet will try to connect to the first filesystem that is configured when the system is powered on, and load the 'boot.zx' file from there.

By default, your Spectranet interface is shipped with filesystem 0 set to tnfs.bytedelight.com and autoboot is enabled.

In this utility you can press "A" to set a filesystem, and then:

- Press "0", and Enter
- Enter the filesystem address, e.g.: tnfs://vexed4.alioth.net
- Press D to save the changes

When the Spectrum is reset, the server "vexed4.alioth.net" will be mounted.

If there are problems mounting a fileserver (for example, the fileserver isn't running) you can press BREAK while the Spectrum is trying to mount it to abort the operation.

Advanced Spectranet BASIC extensions

Navigating a filesystem

Using the %mount command

%mount <fsnum>,"<url>" - Mount a filesystem. Examples:

```
%mount 0,"vexed4.aliath.net"  
%mount 0,"192.168.0.1"  
%mount 1,"tnfs://192.168.0.1"  
%mount 3,"tnfs://example.com/games"
```

The URL works just like you would expect. If there were a filesystem for a DivIDE, the command may be something like **%mount 0,"ide://0** (or something very similar).

Using the %umount command

%umount <fsnum> - Disconnect from a filesystem. Example:

```
%umount 0
```

Using the %cat command

%cat "<fsp>" - Show the contents of a directory. Examples:

```
%cat  
%cat "foobar"  
%cat "0:/foo/bar/baz"
```

The path functions like a Unix path, the separator between the directories that make up the path is a "/". As expected, "." means the current directory, ".." means the directory above.

Using the %cd command

%cd "<fsp>" - Change directory. Examples:

```
%cd "/"  
%cd "games"  
%cd "/programs/basic"  
%cd ".."
```

Using the %fs command

%fs <fsnum> - Change filesystem. Example:

```
%fs 1
```

The default filesystem is 0.

Loading and saving files

Using the %load command

%load "<fsp>" [CODE address] - Load a file. Examples:

```
%load "manic"  
%load "image" CODE 16384  
%load "/foo/bar/baz.zx"  
%load ""
```

The command %load "" will load the BASIC program "boot.zx". (If you've selected the Autoboot option in the %fsconfig menu, the Spectrum will try to load "0:/boot.zx" when it is reset).

Using the %save command

%save "<fsp>" [CODE address,size] - Save a file. Examples:

```
%save "program"  
%save "image" CODE 16384,6912  
%save "program" LINE 1
```

Using the %aload command

%aload "<fsp>" CODE address - Load arbitrary data. Example:

```
%aload "machinecode" CODE 32768
```

The difference with %load and %aload is that %aload effectively loads a "headerless" file. The "normal files" are formatted as a TAP archive, that's to say they have the same headers that are found in a file saved to tape, they contain the address where the data should be loaded and the kind of file - for example, "Program" or "Bytes". The %aload command can load any file. Because of this you always have to put the load address where the data must be loaded because there's no header to specify this information.

Therefore %aload is essential when you want to load a file that you've created on a PC (for example, with a cross assembler) that wasn't saved formatted as a TAP file.

Using the %tapein command

%tapein "<fsp>" - Load a .TAP archive. Example:

```
%tapein "jsw.tap"  
LOAD ""
```

Effectively, %tapein is to emulate a tape. You always need to LOAD "" (or similar) after %tapein to actually run the "virtual tape".

Using the %loadsnap command

%loadsnap "<fsp>" - Load a .SNA snapshot. Example:

```
%loadsnap "matchday.sna"
```

This command works with both 48K and 128K snapshots.

Opening and closing files

There is a set of commands that are used with ZX BASIC streams. They work with files, directories and network sockets. Here is an example and a short explanation.

```
10 %fopen #4,"file","r"  
20 %eof 100  
30 INPUT #4;a$  
40 PRINT a$  
50 GO TO 30  
100 %close #4
```

This program opens the file named "file" and reads the contents, putting them on the screen. The first line:

```
10 %fopen #4,"file","r"
```

opens the file called "file" for reading. Line 20 is to catch the end-of-file condition. What it says is that when an EOF is encountered, jump to line 100. Lines 30, 40 and 50 are standard ZX BASIC commands to read the stream and put the results on the screen. Finally we arrive at line 100, where we close the file. (It is very important that this is done!)

Opening and closing sockets

It's very easy to do similar things with a network connection. For example:

```
10 %connect #4,"spectrum.alioth.net",80  
20 %eof 100  
30 PRINT #4;"GET HTTP/1.0"  
40 PRINT #4  
50 INPUT #4;a$  
60 PRINT a$  
70 GO TO 50  
100 %close #4
```

With code very similar to the first example, we can connect to another machine over the network using a TCP socket. The syntax of %connect is:

```
%connect #<stream>,"address",port
```

Here is a very simple example of a server written in BASIC:

```
10%listen #4,2000  
20%accept #5,4  
30 INPUT #5;a$  
40 PRINT #5;"You sent me: ";a$  
60%close #5  
70%close #4
```

You can try this out from a PC using "telnet":

```
telnet <ip-address-of-spectrum> 2000
```

For example:

```
serendipity:~ winston$ telnet 172.16.0.41 2000
Trying 172.16.0.41...
Connected to 172.16.0.41.
Escape character is '^]'.
hello
You sent me: hello
Connection closed by foreign host.
```

A little more complex example: a server that handles up to 3 connections at the same time on port 2000:

```
1 DIM c(4)
10%listen #4,2000
20%control #5
30 PRINT #5;"p"
40 PRINT "Waiting..."
45 INPUT #5;a;a$
50 IF a<>0 THEN GO TO 200
60 LET a$=INKEY$
70 IF a$="x" THEN GO TO 700
80 GO TO 45
200 IF a=4 THEN GO TO 400
210 IF a$="disconn" THEN GO TO 600
220 INPUT #a;c$
230 PRINT "Strm ";a;": ";c$
240 IF c$="rnd" THEN PRINT #a;RND: GO TO 40
250 IF c$="foo" THEN PRINT a;"Foo bar baz": GO TO 40
260 IF c$="quit" THEN PRINT #a;"Adios":GO TO 600
300 PRINT #a;"I didn't understand that"
310 GO TO 40
400 FOR i=1 TO 4
410 IF c(i)=0 THEN LET chan=i+5: GO TO 500
420 NEXT i
430 PRINT "Out of streams"
440 STOP
500 PRINT "Accepted connection on stream ";chan
510%accept #chan,4
520 LET c(i)=1
530 GO TO 40
600 PRINT "Closing #";a
605%close #a
610 LET c(a-5)=0
620 GO TO 40
700 PRINT "The program has finished"
710%close #5
715 PRINT "Closing socket"
720%close #4
730 PRINT "Closing connections"
740 FOR i=1 TO 4
```

```
750 IF c(i)=1 THEN %close #i+5
760 NEXT i
```

To try it out from a PC, using "telnet <ip-of-spectrum> 2000". You can type "rnd", "quit" or "foo".

Notable parts of the program:

```
10 %listen #4,"2000"
```

When this line executes, the Spectrum begins to listen on port 2000 (tcp). We need to do something else, too: when a connection arrives we must accept it. We use the %accept command for this. However, %accept will block waiting for a connection (in other words, execution of the program stops while waiting). To be able to continue without waiting, there is a control channel that allows us to look at the state of a socket. It's opened like this:

```
20 %control #5
```

Now stream #5 is the control channel. We can write and read this stream like this:

```
30 PRINT #5;"p"
50 INPUT #5;a;a$
```

Line 30 says to the control channel "tell me if there's data waiting on any socket" (this includes a socket that's in the listening state). In line 50, two things arrive: the stream number on which data is ready, and some information about the data. (Data being ready may mean a connection to be accepted, the remote end closing, or just some characters to read). If the stream is #4, we know that it must be a new connection that needs to be accepted, because stream 4 is the listening socket. If the variable a contains 0, no sockets have data waiting.

```
510 %accept #chan,4
```

In this line we finally accept the new connection. The %accept command accepts the connection and creates a new stream for data to be read/written to this connection. The syntax of %accept is:

```
%accept #stream,listening-stream
```

Stream 4 is the listening stream in this case. The variable #chan is the number of the new stream we want to open.

Machine code

If you want to write programs in C, there is a guide with examples. Sockets are used in the same way as they are in Unix or Windows - if you already know how to write networked programs for modern operating systems you'll have no problems. If you've never used the BSD socket library, there are tutorials:

<http://spectrum.alioth.net/doc/index.php/Software>

Spectranet and TNFS software have been developed by Dylan Smith

sinclair.wiki.zxnet.co.uk/wiki/Spectranet

Document revision: March 2023

Document by Ben Versteeg - ByteDelight

ByteDelight.com

BYTEDelight

ZXspectrum.Shop