

SIF-TRD utility for ZX Spectrum

About origins.

Originally SIFTRD was utility for backuping TRDOS disks through SIF interface, which is very capable serial port interface for ZX Spectrum with two hardware UARTs 16C650. These UARTs allows transfer data through RS232 with hardware data flow control independently in Z80 CPU. Thanks to capabilities of UART 16C650 data can be transferred not only faster up to 115200bps, but very reliable too. Internal 32 byte buffer in UARTs really helps.

More about SIF on my website in article about utility SIF-TERM. (Maybe in future I will wrote special article about SIF too. I already have almost finished project with replica of this interface.)

I published first version of SIFTRD in 2010. Program back then was very simplified and worked only with RAW data without any transfer protocol and worked only with 640kB disk. But it still allowed me backup several boxes of TRDOS disks. Unfortunately SIF was and still is very rare between ZX Spectrum users, so I had no feedback from anyone. Because of that I didn't had any reason do it better.

Until one day...

(It was rainy, stormy and very dark outside... it should be like that, right?)

Capabilities of 2021 version

SIFTRD can

- backup RAW data through SIF with speed 115200bps
- restore RAW data through SIF with speed 115200bps
- backup RAW data through software RS232 on AY-3-8912 with speed 57600bps
- restore data through software RS232 on AY-3-8912 with speed 38400bps with sercp utility

ZX Spectrum unfortunately is not fast enough to be able receive data with higher speed, control data flow and read keyboard at a same time. So RAW restore through software RS232 is implemented, but is almost never reliable enough.

Big problem are buffers in much faster PC's. At time when ZX Spectrum is setting RTS inactive in these buffers are already several bytes and these bytes will be send in time when ZX Spectrum is not expecting any more data. It is impossible predict how many bytes will be sent. Without FIFO buffer in hardware UART will be these bytes lost. It can be solved with much lower baudrates or data can be sent in small predictable blocks which can ZX Spectrum receive whole. Next block will wait until ZX Spectrum will be ready again. This method, combination of lower baudrate and predictable data blocks, is using utility sercp originally written by Pavel Vymetalek for ESXDOS.

Problem with buffers can be even worse with USB RS232 converters instead a real RS232 port.

RAW transfer can be optionally verified with CRC32 checksum utility which is available in any Linux distribution. It is not usually in default installation, but almost always is available in repositories. I know, computing of CRC32 take significant time, but it is still the best compromise between availability, reliability and compatibility on all platforms (especially ZX Spectrum and Linux).

What you mean with software RS232?

Original ZX Spectrum 128k+ toastrack was first ZX Spectrum with this internal serial port. But many years it was almost forgotten and very rarely used.

As was tradition in Sinclair devices, serial port was built very cheaply, there was no UART, there are only voltage level converters directly connected to the IO pins of sound chip AY-3-8912. Whole data serialization in both directions must do Z80 itself. Therefore it is very time consuming, especially on higher baudrates is every CPU tick important and timing must be very precise. That is why it is called software RS232, because software must control everything.

Which was maybe worse, in original manuals is almost nothing about RS232. Newer manual from Amstrad for grey ZX Spectrum 128k +2 have at least pinout in chapter about RS232, but nothing more, not even anything about BASIC commands which supports RS232. So everything must have been analysed later.

Unusuall connector BT631W didn't help either. These connectors are almost unobtainum today. Fortunately sometimes can be bought cable with connectors BT631A (all 6 wires must be connected) which was used for telephones by British Telecom. Connector BT631A can be modified to be usable in ZX Spectrum. Not perfect, but it works.

But it is almost only one standard interfaces which is available in original ZX Spectrum 128k.

When I saw RS232 ROM from Paul Farrow and later sercp utility written by Pavel Vymetalek it was almost mandatory I must modify at least some of my utilities to be able use this RS232 port too.

RAW backup through SIF

When using SIF, the procedure and settings are the same for both the old and new versions of SIFTRD utility, except for the CRC32 calculation.

- On the PC, set the serial port (in Linux with the stty command).
- On the PC, start receiving and saving data to a file (on Linux you can use the cat command)
- In SIFTRD, select the serial port "**SIF 115200bps UART 16C650**" (default option)
- In SIFTRD select the drive with the source floppy disk
- In SIFTRD select BACKUP TRDOS drive
- wait
- When ZXS reports completion on PC stop receiving data (on Linux CTRL+C)
- The file should be exactly 655360 bytes (or 327680 or 163840 for smaller capacities)
- Compare the CRC32 displayed by SIFTRD against the CRC32 calculated on the PC on the received file

Example:

```
stty -F /dev/ttyS0 115200 cs8 clocal cread cstopb -parenb -parodd crtscts raw
cat /dev/ttyS0 > target_disk_image.trd
```

RAW restore through SIF

This method is only possible due to the small buffer directly in the 16C650 chip (32 bytes FIFO buffer). Without it and without the data flow control directly by the 16C650 chip, data could be lost during transmission and the transmission would not be usable.

- Set up the serial port on the PC (in Linux with the stty command).
- In SIFTRD select the serial port "**SIF 115200bps UART 16C650**".
- In SIFTRD select the drive with the destination floppy disk.
- Make sure that the correct floppy disk is indeed inserted, the original data on it will be lost.
- In SIFTRD select RESTORE TRDOS drive
- SIFTRD will first attempt to empty the buffers by briefly receiving the data from PC
- Now send the data from the PC using e.g. the cat command on Linux
- SIFTRD receives the first 4kB and verifies that the TRDOS mark and the floppy format are correct.
- If everything is ok, SIFTRD continues receiving data and writes to the floppy disk.
- If the TRDOS tag and format are missing, the user can accept the continuation.
- Wait
- At the point when the cat command completes sending data, SIFTRD should report the completion of the transfer.
- Compare the CRC32 displayed by SIFTRD against the CRC32 computed on the PC from the sent file

Example:

```
stty -F /dev/ttyS0 115200 cs8 clocal cread cstopb -parenb -parodd crtscts raw
cat target_disk_image.trd > /dev/ttyS0
```

RAW backup through software RS232 on AY-3-8912

This procedure is no different from the previous backup via SIF, except for the transfer speed which is limited by speed of Z80.

- Set up a serial port on the PC (in Linux with the stty command).
- On the PC, start receiving and saving data to a file (in Linux with the cat command)
- In SIFTRD select the serial port "**RAW 57600bps (AY-3-8912)**"
- In SIFTRD select the drive with the source floppy disk
- In SIFTRD select BACKUP TRDOS drive
- Wait
- When ZXS reports completion on PC stop receiving data (on Linux CTRL+C)
- The file should be exactly 655360 bytes (or 327680 or 163840 for smaller capacities)
- Compare the CRC32 displayed by SIFTRD against the CRC32 calculated on the PC on the received file
-

Example:

```
stty -F /dev/ttyS0 57600 cs8 clocal cread cstopb -parenb -parodd crtscts raw
cat /dev/ttyS0 > target_disk_image.trd
```

SERCP restore through software RS232 on AY-3-8912

This is probably the most interesting part of new version of SIFTRD. The utility sercp has become popular in latest versions of ESXDOS for various modern storage devices such are DivIDE, DivMMC or MB03+. Pavel Vymetalek originally wrote it to be able transfer TAP files between computers, but sercp is generally usable for any files.

Sercp solves problems with data flow control and pausing while ZX Spectrum is writing data to disk. To prevent data sent from the PC to the ZX Spectrum from being lost, sercp sends data to the ZX Spectrum (or any other computer) in blocks whose size is known in advance. After it has finished sending a block of data, it waits for a short time before testing whether the receiving computer is able to receive again.

- In SIFTRD select the serial port "**SERCP 34800bps (AY-3-8912)**".
- In SIFTRD select the drive with the destination floppy disk.
- Make sure that the correct floppy disk is indeed inserted, the original data on it will be lost.
- In SIFTRD select RESTORE TRDOS drive

- SIFTRD will first attempt to empty the buffers by briefly receiving the data
- After the buffers are emptied, run sercp with the appropriate parameters on the PC.
- SIFTRD receives fileinfo and the first data block, verifies that the TRDOS tag and floppy format are correct.
- If everything is ok, SIFTRD continues receiving data and writes to the floppy disk.
- If the TRDOS tag and format are missing and the checksums still match, the user can agree to continue.
- If the checksums do not match, SIFTRD aborts data reception and stops saving.
- When the data transfer is complete on both computers, sercp will terminate successfully.
-

The serial port speed is set by sercp itself. The usage should be the same on all operating systems supported by Pavel Vymetalek.

Connection between PC & ZX Spectrum with SIF

Use standard Null Modem cable with crossed signals RxD – TxD and CTS – RTS.

Connection between PC & ZX Spectrum with software RS232

BT631 pin	RS232/MIDI by manual	RS232 better names	AY-3-8912 signal	direction	DSUB-9 pin
1	GND	GND	GND	GND	5
2	TxD	RxD	I/O A7	vstup	2
3	RxD	TxD	I/O A3	výstup	3
4	DTR	CTS	I/O A6	vstup	8
5	CTS	RTS	I/O A2	výstup	7
6	+12V	+12V	-	+12V	-

This is wiring of cable which have BT631W on one side and 9pin male DSUB connector on other side. It will allow use RS232 on ZX Spectrum as standard RS232, between this DSUB and other computer you will need same nullmodem cable as for SIF.

You can connect two computers directly without second nullmodem cable, but I consider this solution more practical, because it allows me use standard RS232 cables and connect ZX Spectrum not only to PC, but connect RS232 devices, such are plotters to the ZX Spectrum too. For both I have same BT631W-DSUB cable, but different DTE-DTE (nullmodem) or DTE-DCE (1:1) cables connected to it, and on DCE I am not limited to only one type of DSUB connector.

Use SIFTRD carefully, it still can have bugs.