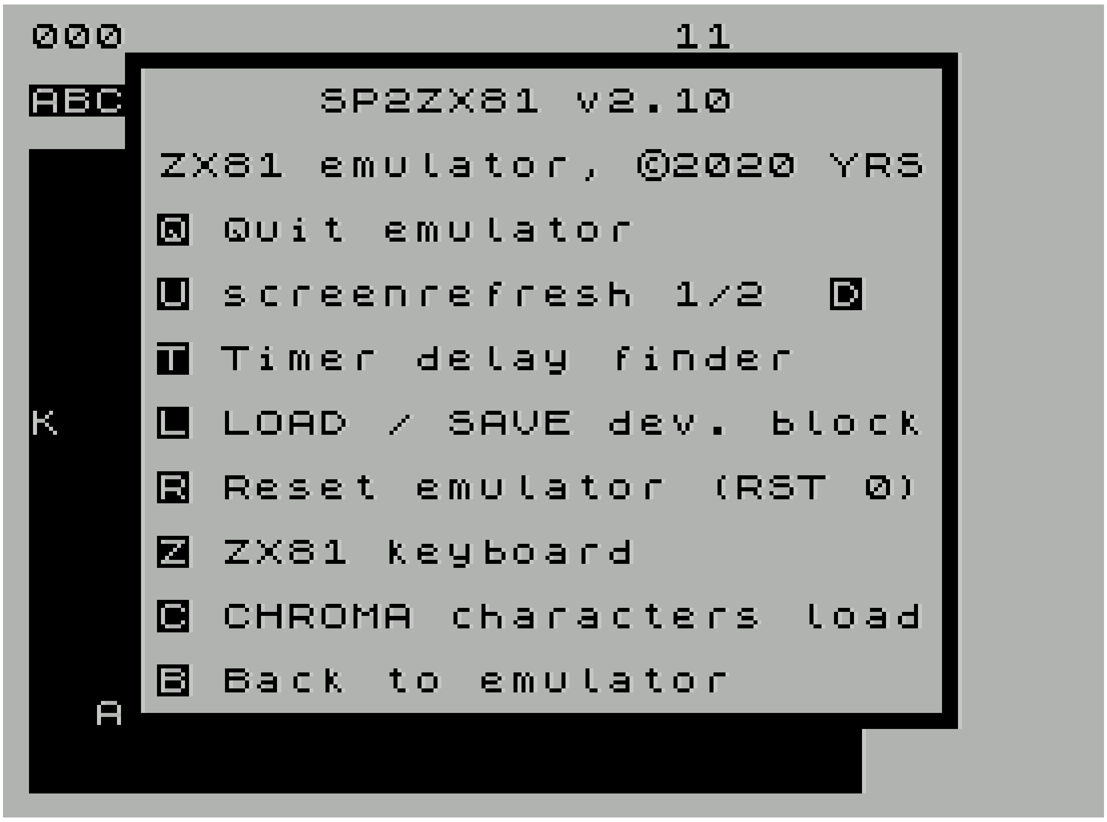
SP-2-ZX81



```
000                        11
ABC       SP2ZX81 v2.10
    ZX81 emulator, ©2020 YRS
    Q Quit emulator
    U screenrefresh 1/2  D
    T Timer delay finder
K   L LOAD / SAVE dev. block
    R Reset emulator (RST 0)
    Z ZX81 keyboard
    C CHROMA characters load
    B Back to emulator
  A
```

1 Introduction

When I wrote ZX81EMUL, the first ZX81 emulator for the ZX Spectrum, I was happy with the result. Some years later I thought I could code a better version of the emulator but there was never a good reason to start coding a new version. My latest lowresgames use modern codingtechnics that made it worthwhile to code this emulator.

1.1 History

In 1997 my first emulator was released. I still remember the reactions on it during the meeting in Mönchengladbach. I did code a renewed version with a new mainprogram but that version was lost when I changed computers or whatever diskette it was saved on. The mainprogram that was develooped then is later used in emulators that are written for the SAM Coupe. The Galaksija-emulator uses this fast mainprogram.

In 2019 I started coding the emulator again but the project was on hold. I missed the incentive to code the new emulator. Would it get the higher speed I thought was possible or would I waste a lot of time for just a few percentages of speed? After finishing my 10 lowresgames I thought it was time to give a new emulator a change.

2 The emulator

The ZX81 is still the same machine as in 1997 but my knowledge about the ZX81 has increased. The 1997 emulator was a full Z80 emulator placed in a ZX81. This new emulator does not emulate all Z80-opcodes but only the Z80-opcodes necessary for a LOWRES ZX81.

2.1 How to start the emulator

When you have the tape of the emulator you type LOAD “” and make the tape play.
The computer will load a BASIC-program and 1 single codeblock. After loading the ZX81-emulator will start and displays the K-cursor.

2.2 What does the emulator do?

The emulator will make the ZX Spectrum react like a 16K LOWRES ZX81.In theory every lowres ZX81 game should work. In practice games that uses hardware features like mirrormemory can crash since this is not emulated. Most lowresgames will run on this emulator. As said the Z80-emulator is coded to work most optimal on a ZX81. This means that instructions that are not needed for gameplay are not emulated. The IX-register is such an example. This is used to trigger the display of the ZX81. This done in a different way and IX is therefore not needed. The ROM is altered to bypass IX-commands. IY is used for accessing the systemvariables. This is also the only function IY may have.

2.3 The I/O ports

This emulator is mainly programmed to play games. I/O ports for the printer is skipped.
The only I/O possible is by the keyboard, loading and saving games. To save game you need to enter the name in ZX Spectrum basic to save with the correct ASCII-name.

2.4 The keyboard

The ZX81 keyboard will return the value of #7F when no key is pressed on a port.
The ZX-spectrum gives #BF or #FF. Only the IN A,(N) is corrected for this feature. Some games read #7F as no key pressed. Other registers are mostly not used to check the keyboard. This might cause games not to work properly.

2.5 Hires

Hires has a major impact on the speed of the emulator. For this reason only lowres is emulated.

Most hires games also use IX-register to trigger display. Since IX is not supported hiresgames will not work neither.

2.6 Modern interfaces

Only a 16K lowres ZX81 is emulated.
From the CHROMA-interface the modified characters is supported. More in chapter 4. No other modern interface is supported.

2.7 ROM 8192-16383

Like the 1997 version this part is used by the emulator.Games that use this memory will make the emulator crash.

2.8 Screen-emulation

Like the ZX81 during intrupt the screen is displayed. Only changed items are displayed.
Extra option in this emulator is that “characters” that are in fact hidden data are emulated in the right way. Games like 1K MINESWEEPER store data on the screen In the old emulator this was shown as characters, in this emulator it is tested and when found displayed as a space, like the ZX81 does too.

2.9 The emulator on a PC on a ZX Spectrum emulator

There is no difference for the emulator if it runs on real hardware or emulation on a PC.
The menu however provides with an extra option to load memoryblocks when the emulator runs on a ZX Spectrum emulator.

2.10 The menu

The first page of this manual shows the menu activated during a BASIC-program.
The menu is activated in the same way as the old emulator: PRESS 1 and 0.
The options in the menu are :

- Quit the emulator
- Up or Down the screenrefreshcounter
- T Timer delay finder
- L Select LOAD/SAVE device
- R Reset the emulator
- Z Show ZX81 keyboard
- B Back to the emulator

- Quit the emulator

With “Q” you quit the emulator. The program will stop emulating and returns to ZX Spectrum BASIC. The emulator can be restarted with RUN 9999

- Up or Down the screenrefreshcounter

With “U” and “D” you can increase or decrease the number of intrupts before a screen is redrawn. Games with littile changes will run faster when less screenupdates are done. Use this the way you like to play a game.

- T Timer delay finder

Some games have delays built in. On a real ZX81 without the delay the game would run too fast. On this emulator less delay is possible to speed up the game. This delay finder can do 3 things:

1) On compressed screens is searches for unused lines at the bottom and replaces on the first unused line the HALT by JP (HL). The emulator stops updating the screen when a JP (HL) is found.

The more lines are unused the quicker the screenrefreshment goes.

2) The framescounter is used for delayloops. 2 routines with framescounter are searched for
Loop 1:

```
        LD A,(HL)
        SUB nr
WAIT CP (HL)
        JR NZ,WAIT
```

This loop will replace the JR NZ,WAIT with a special routine>
When the emulator executes the code it will check if this loop uses FRAMES.
If so it will DECREASE the nr-value by 1 and repair the JR NZ,WAIT.
If not it will just repair the JR NZ,WAIT. The next time the waiting will be 1/50 less, time that is added to the emulator.

3) The second loop with frames is searched for:
Loop 2:

```
        ADD A,(HL)
WAIT    CP (HL)
        JR NZ,WAIT
```

The finding of the loop is the same, This is entered with the correctionvalue preset in A.
Games with this loop often use this routine wit multiple values in A.
When FRAMES is used it will replace this piece of code with a CALL to a special routine where the value of A is checked and INCREASED when possibile.

The first routine can be updated until nr reaches 1. Some delay will be kept. The second routine can be updated 3 times and no more.

When you search for the routines the border will go black indicating the search. Once back in the emulator the border will go black again when succesfully a delayloop in altered.
BEST USE THIS OPTION IN A STATIC MOMENT OF A GAME, THERE IS A SMALL CHANGE OF CRASHING WHILE PLAYING A GAME .
- L Select LOAD-device
By default LOADING is done from tape. You can change devices by pressing L
The second option is LOADING BINARY BLOCK. With this option you can load .P directly in the ZX Spectrum emulator you use to emulate a ZX81.
The third option is MICRODRIVE 1.

- R Reset the emulator
This is the RESET-button to restart the emulator.

- Z Show ZX81 keyboard
The keyboard of the ZX81 is different from the ZX Spectrum. This option will show the ZX81 keyboard.

- B Back to the emulator
This option returns to the emulator the moment you interrupted with calling the menu.

2.11 Change to a not defined LOAD-device

If you happen to have other storage systems with other LOAD-syntax you can alter the BASIC to your device. Restart the emulator by saving the emulator with RUN 9900.

3 Technical Manual

This chapter will show how the emulator was coded. Each part is chosen carefully to optimize the emulator in all posible ways.

3.1 The memory layout

A standard ZX81 consist roughly of the following memory

#0000 - #1FFF	ROM
#2000 - #3FFF	unused
#4000 - #43FF	1K RAM
#4400 - #7FFF	extra 15K RAM

Above 32K is used by the hardware to do a screendisplay. On a 16K ZX81 there is physically nothing here.

The ZX Spectrum memory layout is simplified:

#0000 - #3FFF	ROM
#4000 - #5AFF	screen
#5B00 - #FFFF	RAM

The RAM below 32K is also contended, which simply means it is slower than RAM above 32K. Above 32K we have 32K in 1 block available for the 32K memory needed for the ZX81.
If we use this memory we need to correct certain Z80-opcodes to work properly.
To do a LD A,(HL) we need to add 32K to the value of HL, which is nothing more then just setting bit 7 of the H-register. The ZX81 doesn't use 8K-16K, so we have 8K to code a working emulator. The memory below 32K will be used for less time critical routines like the menu, the keyboard and loading.

For this emulator the ZX Spectrum memory is organized as:

#0000 - #3FFF	ROM
#4000 - #5AFF	screen
#5B00 - #7FFF	BASIC, menu and keyboardpicture
#8000 - #9FFF	ZX81 ROM (with alterations for the emulator)
#A000 - #BFFF	The actual emulator
#C000 - #FFFF	ZX81 RAM memory

3.2 The mainprogram

Due to the dislocated addressed we need to emulate Z80-opcodes although the ZX Spectrum has a Z80 processor itself. Each opcode must be analyzed and executed correctly.

This is done by a very small mainprogram where the DE-register acts like the program counter. The next opcode to execute is pointed by DE. Then this mainprogram starts:

```
        LD      A,(DE)          ; ( 7) fetch opcode
        LD      L,A             ; (11) set lowbyte of routine
        LD      H,B             ; (15) point to highbytetable, B is constant value
        LD      H.(HL)          ; (22) get highbyte of routine from table
        JP      (HL)            ; (26) go to routine of opcode
```

After years of experimenting this is the quickest way to jump to the needed routine within a few K

of available memory. You still need 26 tstates only to get to the routine of the opcode that needs emulation. For a simple NOP (opcode 00,4 tstates) you now need to increase DE and go back to mainprogram. To do this an INC DE is added before the mainprogram which is pointed to as opcode 00. To emulate NOP (4 tstates) you need 26 (main) + 6 (INC DE) 32 tstates ;
an emulation speed of 12,5%. Other opcodes can be faster or slower.

3.2 Userdefined instruction

To add speed to the emulator parts of the ROM are translated to work with the altered locations of the memory. These routines can be called by using empty parts in the ED and IX table. ED00 has no effect on a normal Z80. In the ROM on several locations these unused opcodes are placed and when met the emulator will execute a translated part of the ROM. This gives the emulator its speed in BASIC. In machinecodegames the speed can only be gained with the delayfinder or less screenupdates.

3.4 The source

The emulatorsource is freely available as textdocuments to this manual.
The ROM is not part of the source, except for the altered ROM which is part of the emulator.

4 Conversion

In the 1997 emulator a tool to convert physical ZX81 tapes was added. For conversion you will need to use that tool. For the CHROMA-interface you need an extra tooling under a PC-emulator.

4.1 81 TO SPECCIE

81 TO SPECCIE is the tool to convert ZX81 tapes on a real ZX Spectrum to 1500-baud loadfiles which both emulators can load. You may also convert these files to other LOADING-devices on a ZX Spectrum

4.2 CHROMA characterfiles

The CHROMA-interface supports both user graphics and colours. To emulate colours we would need to add the attribute file which would make the display slower. The user grpahics however could be emulated by extending the actual characterset and read from the extended characterset.
The CHROMA interface uses normal and inverted characters, 128 characters can be altered.
For my emulator I also use the 64 characters from 64 to 127. On a ZX81 these characters will display as a space, so the charaters are space only.
The CHROMA-character files are defined as XML-messages. For the ZX Spectrum these files are much too long to use. To use CHROMA-characters within SP-2-ZX81 you need to convert the XML-files to TAP-files for the emulator. The ConvCHR.Z80 can do this job. LOAD the snapshot and type RUN. Insert a (blanc) tap-file to save to. Keep the saved file near the original game.
The characterset is ready to load with the game.

Loading CHROMA-characters into the emulator
To load the characters you start with a NEW ZX81.
Due to the used memory by the emulator the ZX81 memory is temporarely used to load the characters. This means that the characters must be loaded before the game is loaded.
Type LOAD “”, but not NEWLINE. Activated the menu and press C. Play the TAP-file with the characters. When loaded return to the menu. Select the medium where you want to load and go back to the ZX81. Press newline and load the game.

4.3 .P to TAP-files

With the CHROMA-characters in TAP-file it is easy when the game itself is inside the TAP too.
With the tool P-TO-TAP.z80 you can add the P-file to the TAP-file. LOAD the Z80 and RUN it.