

SD81 Booster

User Manual v1.0

Expansion Interface for the Sinclair ZX81
SD - AY Sound - Speech - 512KB RAM - RTC

Open Hardware and Open Software
RancanuoTeam / SD81 Booster Project



SD81 Booster interface, English edition cover artwork

Contents

Quick Start Guide

1. Introduction
2. Hardware Description
3. Package Contents
4. Installation
5. Preparing the microSD Card
6. First Steps
7. Loading and Saving from SD
8. File and Directory Management
9. Additional Functions
10. Sound
11. Memory Management
12. Advanced BASIC Extensions
13. Example Programs
14. Error Codes
15. For Programmers
16. Troubleshooting
17. Firmware Updates
18. Glossary
19. Firmware Version History
20. References

Appendix A - Complete PLAY Command Reference

Appendix B - PEG Effects Generator Reference

Appendix C - Speech Synthesiser Dictionary

Appendix D - Memory Paging System

Appendix E - Chroma81 Interface Port (7FEFh)

Appendix F - Superfast and Spectrum Modes

EDITORIAL NOTE: In Microsoft Word or LibreOffice, right-click the table of contents field and choose Update Field if you replace this manual with an automatic index.

Quick Start Guide

If you have just opened the box and wish to begin at once, follow these five steps.

1. Prepare the microSD card: format it as FAT32, copy the SYS directory to the root of the card, and copy your .P files onto the card.
2. Power off the ZX81. Connect the SD81 Booster to the rear expansion connector and insert the microSD card into the interface.
3. Power up the ZX81. The machine should start normally and display the usual K cursor. The STAT LED should remain steady green.
4. Load your first program from SD with LOAD FAST "NAME", replacing NAME with the filename without the .P extension.
5. To explore the SD card, use LOAD *DIR.

```
LOAD FAST "NAME"  
LOAD *DIR
```

1. Introduction

The SD81 Booster is an open-hardware expansion interface for the Sinclair ZX81. It extends the original machine while retaining the character and operating style of the ZX81. The interface is intended for users, collectors, developers and hardware enthusiasts who wish to use modern storage and enhanced facilities without losing compatibility with classic software.

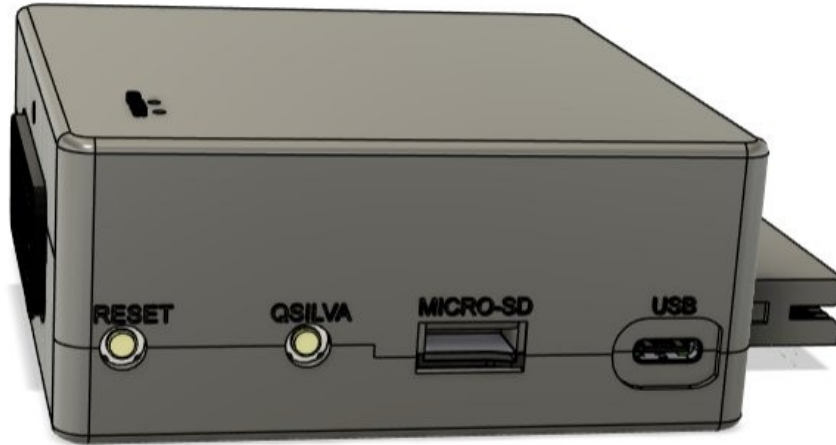
- Loading and saving from microSD card using the standard .P format used by popular emulators.
- Up to 512KB of RAM using an 8KB block memory mapper.
- AY-3-8910/12 sound emulation with support for a PLAY-style command.
- Background playback of VGM music files containing AY data.
- Speech synthesis using phoneme samples stored inside the microcontroller.
- Up to 128 user-definable characters, compatible with QuickSilva-style character definition.
- Compatibility with the Chroma81 interface for RGB colour video output.
- Full compatibility with the original tape routines and the ZX Printer.

2. Hardware Description

The SD81 Booster is housed in a compact enclosure which plugs into the rear expansion port of the ZX81. The external connectors and controls are arranged to give direct access to storage, reset, joystick, RGB output and programming/debugging facilities.

NOTE: The illustrations in this section are provisional 3D renders and should be replaced by photographs of the production unit before release.

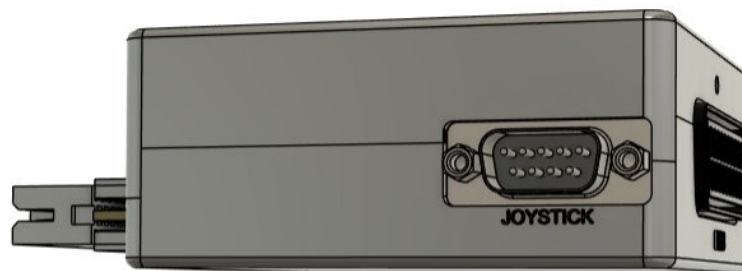
2.1 Right-hand Side Panel



Right-hand side: RESET, QSILVA, MICRO-SD and USB-C.

- RESET button: resets the whole system, including the ZX81 and the interface.
- QSILVA button: toggles between the QuickSilva character set and the standard ZX81 ROM character set. Each press switches to the other set.
- MICRO-SD slot: accepts the microSD card containing programs, ROM images and system files.
- USB-C port: used for microcontroller programming and serial debugging. It is not required for normal operation.

2.2 Left-hand Side and Rear Panels



Left-hand side: DB9 joystick connector.

- JOYSTICK connector (DB9): compatible with standard 9-pin Atari/Commodore-style joysticks. Button-to-key mapping is fully programmable using LOAD *JOY.
- RGB connector (SCART): RGB colour video output compatible with Chroma81, allowing connection to monitors and televisions with SCART input.



Rear panel: RGB SCART connector.

2.3 Top Panel - Status LEDs



Top panel: STAT and SD LEDs.

- STAT LED: indicates the general state of the interface by colour and flashing pattern.
- SD LED: flashes during read or write access to the microSD card.

2.4 STAT LED Status Table

Colour / Pattern	Meaning
Steady green	Interface ready. System operating normally.
Steady yellow	Loading a file from the SD card.
Steady blue	Saving a file to the SD card.

Steady red	SD card access error.
Flashing purple	Audio playback, WAV or speech synthesis.
Fast flashing red	Critical system error.

3. Package Contents

The SD81 Booster package contains the following items:

- 1 x SD81 Booster interface
- 1 x microSD card, pre-formatted
- 1 x CR2032 button cell, pre-installed in the interface
- This user manual

WARNING: If any item is missing or damaged, contact the seller before connecting the interface.

4. Installation

4.1 Before Connecting the Interface

- Ensure that the ZX81 is powered off before connecting or disconnecting the interface.
- The SD81 Booster connects to the rear expansion port of the ZX81.
- The interface is not compatible with devices which replace the internal ZX81 ROM.

4.2 Connection

6. Power off the ZX81.
7. Align the SD81 Booster connector with the rear expansion port. Check that all pins are correctly aligned.
8. Push gently until the connector is fully seated. Do not force the connection.
9. Insert the microSD card into the interface slot.
10. Power up the ZX81.

WARNING: Connecting or disconnecting the interface while the ZX81 is powered may damage both the interface and the computer.

4.3 Checking the Installation

When the ZX81 starts with the SD81 Booster fitted correctly, it should boot normally and display the usual K cursor. The interface does not alter the normal start-up screen.

To verify that the interface is running, enter the following BASIC command and press ENTER:

```
LOAD *VER
```

NOTE: The asterisk * is obtained with SHIFT + B. The interface will display the installed firmware version.

5. Preparing the microSD Card

5.1 Card Format

The SD81 Booster requires a microSD card formatted as FAT32. The card supplied with the interface is already prepared. If you use your own card, format it as FAT32 and copy the SYS directory and all its contents to the root of the card.

WARNING: exFAT and NTFS are not supported. The SYS directory is mandatory because it contains ROM and system files required by the interface.

5.2 Permitted Characters in File Names

- Letters A to Z. Saved filenames are converted to upper case.
- Digits 0 to 9.
- Symbols: . , ; \$ () = + -
- The / character is used as a directory separator and cannot be used inside a filename.

TIP: Avoid ending a filename with a space or full stop, as some host operating systems may have difficulty reading such files.

5.3 Recommended Directory Structure

```
/
| - AUTOEXEC.P          Program loaded when starting from RUN
| - GAMES/
|   | - MANIC.P
|   | - PACMAN.P
| - DEMOS/
| - SYS/                Mandatory system directory - do not modify
```

5.4 The AUTOEXEC Program

If a file named AUTOEXEC.P is present in the root of the SD card, it is loaded and executed automatically when RUN is entered on a ZX81 which has no program already loaded. This is useful for custom start-up menus.

6. First Steps

6.1 Loading Modes: SD or Tape

By default, the BASIC LOAD and SAVE commands use tape exactly as on an unexpanded ZX81. To use the SD card, you may either enable SD mode permanently or add FAST to individual commands.

```
LOAD FAST
```

From that point, LOAD and SAVE use SD by default. To return to tape mode:

```
LOAD SLOW
```

NOTE: FAST is obtained with SHIFT + F and SLOW with SHIFT + D. These are ZX81 BASIC tokens and should not be typed letter by letter.

6.2 Loading Your First Program from SD

11. Ensure that at least one .P file is present on the SD card.
12. Power up the ZX81 with the interface and card fitted.
13. At the K cursor, enter LOAD FAST "NAME" using the filename without its extension.
14. Press ENTER. The program will load and will auto-start if it contains an auto-start line.

7. Loading and Saving from SD

7.1 Loading a Program

```
LOAD FAST "NAME"  
LOAD FAST "GAMES/PACMAN"  
LOAD FAST "NAME" THEN GOTO 100  
LOAD FAST "NAME" THEN STOP
```

The interface first searches for the exact filename. If it is not found, it attempts to add the .P extension automatically. THEN, GOTO and STOP are ZX81 BASIC tokens.

7.2 Saving a Program

```
SAVE FAST "NAME"
```

WARNING: The interface automatically adds the .P extension. If a file with the same name already exists, it is overwritten without further warning.

7.3 Loading and Saving Memory Blocks

```
LOAD FAST "NAME" CODE 30000  
SAVE FAST "NAME" CODE 30000,2048
```

The address after CODE is mandatory when loading raw data because SD files do not contain a ZX Spectrum-style load address header. For saving, the first number is the start address and the second number is the length in bytes.

7.4 Game Compatibility Notes

- LOAD *128C before loading games which use 128 user-definable characters.
- LOAD FAST before loading multi-file games so that subsequent LOAD and SAVE operations are also directed to SD.

7.5 Recognised File Formats

Extension	Behaviour
.P	Standard ZX81 BASIC program. The interface calculates the real program size from system variables and discards surplus bytes at the end.
.81	Handled in the same way as .P.
.P81	Multi-program format. The embedded filename is skipped before the program data is read.
.ROM	ROM image. Loaded at address 0 and the system is reset. Control does not return to BASIC.
.WAV	Uncompressed PCM audio file. Played directly instead of being loaded into memory.
Other	Loaded into memory as a raw file without processing.

WARNING: Loading a .ROM file with LOAD FAST causes an immediate reset. Ensure that the file contains a valid ROM image.

8. File and Directory Management

8.1 Listing SD Contents

```
LOAD *DIR
LOAD *DIR "GAMES"
LOAD *DIR "*.P"
```

Wildcards are supported: * matches any number of characters and ? matches exactly one character. If the listing does not fit on the screen, an ellipsis is displayed; press any key to continue or SPACE to abort.

8.2 Navigating Directories

```
LOAD *CD "GAMES"
LOAD *CD "/"
LOAD *PWD
```

8.3 Creating and Removing Directories

```
LOAD *MD "NEWDIR"
LOAD *RD "EMPTYDIR"
```

8.4 Deleting, Renaming and Copying Files

```
LOAD *DEL "FILE.P"
LOAD *MV "OLD.P" TO "NEW.P"
LOAD *CP "SOURCE.P" TO "DEST.P"
```

NOTE: TO is a ZX81 BASIC token, obtained with SHIFT + 4.

8.5 Free Space on the SD Card

```
LOAD *FREE
```

This operation may take several seconds to complete.

8.6 T81 Directories - Alpha Feature

The SD81 Booster supports a special .T81 container format which behaves like a directory containing several ZX81 programs. This is comparable in purpose to a ZIP file, but oriented towards ZX81 program collections.

```
LOAD *CD "COLLECTION.T81"
LOAD *CD "/"
```

WARNING: This feature is currently alpha quality. Write operations such as DELETE, MD, RD, MV, CP and SAVE are not supported inside a T81 directory.

9. Additional Functions

9.1 WAV File Playback

Uncompressed PCM WAV files can be played directly from the SD card by loading them normally. When a .WAV extension is detected, the interface plays the file instead of loading it as a program.

```
LOAD FAST "SOUND.WAV"
```

9.2 Real-time Clock - RTC Command

The SD81 Booster includes a battery-backed real-time clock. The LOAD *RTC command displays, reads or sets the date and time from BASIC.

```
LOAD *RTC
LOAD *RTC TO R$
LOAD *RTC="2025-04-30 18:30:00"
LOAD *RTC="2025-04-30"
LOAD *RTC="18:30:00"
```

Format	Example	Description
YYYY-MM-DD HH:MM:SS.CC	2025-04-30 18:30:00.00	Full date and time with centiseconds
YYYY-MM-DD HH:MM:SS	2025-04-30 18:30:00	Full date and time
YYYY-MM-DD	2025-04-30	Date only
HH:MM:SS.CC	18:30:00.00	Time with centiseconds
HH:MM:SS	18:30:00	Time only
HH:MM	18:30	Hours and minutes; seconds are set to zero

9.3 RTC Battery Status - BAT Command

The CR2032 button cell keeps the RTC running while the ZX81 is switched off. Its state may be checked from BASIC.

```
LOAD *BAT
LOAD *BAT TO B$
```

9.4 Extended RAM Mode - RAM48 Command

```
LOAD *RAM48
LOAD *RAM48 STOP
```

LOAD *RAM48 enables 48KB extended RAM mode for BASIC programs and data. Use LOAD *RAM48 STOP to return to the standard compatibility mode if a program does not operate correctly.

9.5 Displaying Text Files - THEN PRINT

LOAD THEN PRINT is similar to the MS-DOS TYPE command or the Unix cat command. It displays a text file directly on the ZX81 screen. LOAD THEN LPRINT sends the same output to the ZX Printer.

```
LOAD THEN PRINT "FILE"
LOAD THEN LPRINT "FILE"
LOAD THEN PRINT "*PLAY"
```

When an asterisk is placed before the name, the interface looks in /MAN/ and appends .TXT, allowing a simple manual-page help system.

9.6 Programmable Joystick - JOY Command

The DB9 joystick port is fully programmable. The configuration string contains exactly five characters in the order left, right, up, down, fire.

```
LOAD *JOY "OPQA "
```

- Left -> key O
- Right -> key P
- Up -> key Q
- Down -> key A
- Fire -> SPACE

10. Sound

The SD81 Booster includes AY-3-8910/12 sound support, giving the ZX81 three-channel music, programmable effects and background playback facilities.

10.1 PLAY Command - AY Music

```
LOAD *PLAY "string1"  
LOAD *PLAY "string1", "string2", "string3"  
LOAD *PLAY "T120 04 5C 5E 5G 9C"
```

Each string corresponds to one AY voice. The notation describes notes, duration, tempo, octave and effects. Notes are written using C D E F G A B. A preceding = raises a semitone; a preceding pound sign lowers a semitone. A pound sign without a following note is a rest.

Value	Name	Duration at 60 bpm
1	Semiquaver	0.25 s
3	Quaver	0.5 s
5	Crotchet	1 s
7	Minim	2 s
9	Semibreve	4 s

- T<number> sets tempo from 60 to 240 bpm.
- O<number> selects octave 0 to 8.
- Parentheses repeat a section once; an unmatched) repeats the whole string indefinitely.
- H stops PLAY even if looping voices are active.
- W selects one of eight envelope effects, U enables the effect on the channel, and X controls envelope speed.

10.2 VGM Player - Background Music

```
LOAD *VGM "MUSIC"  
LOAD *VGM THEN RUN  
LOAD *VGM THEN PAUSE  
LOAD *VGM THEN CONT  
LOAD *VGM THEN STOP  
LOAD *VGMLOOP  
LOAD *VGMLOOP STOP
```

Only VGM files containing AY-3-8910 or AY-3-8912 data are supported. Other sound chips are not compatible.

10.3 PEG - Programmable Effects Generator

PEG is a small virtual machine inside the interface. It executes sound-effect programs independently of the Z80 and can drive the AY registers using up to three parallel threads.

```

LOAD *PEG <address>,"<hexadecimal>"
LOAD *PEG THEN RUN <thread>,<address>
LOAD *PEG THEN STOP <thread>
LOAD *PEG THEN PAUSE <thread>
LOAD *PEG THEN CONT <thread>

```

10.4 Speech Synthesis - SAY Command

The speech synthesiser produces English speech from BASIC. It is based on SP0256-style phonemes; the audio samples are stored in the microcontroller, so no SD files are required.

```

LOAD *SAY "HELLO WORLD"
LOAD *SAY "ZX81 COMPUTER"
LOAD *SAY "ERROR 5"
LOAD *SAY "*HELLO WORLD"

```

By default, BASIC waits until speech playback has finished. If the string begins with *, speech continues in the background and the ZX81 carries on executing the program. The input text should be in upper-case English. Numbers are spoken in English automatically.

TIP: For better pronunciation of unrecognised words, write them phonetically in English; for example, SINCLAIR may sound better as SINCLER.

11. Memory Management

The SD81 Booster provides up to 512KB of RAM. This memory is divided into 64 pages of 8KB and mapped into the Z80 64KB address space in 8KB blocks.

11.1 Blocks and Pages

Block	Address range	Typical use
0	0000-1FFF	ZX81 ROM, read only
1	2000-3FFF	Interface expansion ROM
2	4000-5FFF	Main RAM / display
3	6000-7FFF	Main RAM
4	8000-9FFF	Extended RAM
5	A000-BFFF	Extended RAM
6	C000-DFFF	Mirror of block 2, required for video
7	E000-FFFF	Mirror of block 3, required for video

WARNING: Blocks 6 and 7 must remain mirrors of blocks 2 and 3 for the ZX81 video system to work correctly.

11.2 MAP Command - Assigning Pages to Blocks

```

LOAD *MAP <block>,<page>
LOAD *MAP <block> TO <variable>
LOAD *MAP 4,10

```

The block number is 0 to 7 and the page number is 0 to 63. Once block 4 is mapped to page 10, reads and writes at addresses 32768-40959 access page 10.

11.3 MC45 Mode - Machine Code in Blocks 4 and 5

```

LOAD *MC45
LOAD *MC45 STOP

```

On a standard ZX81, machine-code instructions placed in blocks 4 and 5 are not executed correctly. MC45 disables this limitation and allows ordinary Z80 code to run in that area.

WARNING: MC45 works by briefly forcing the Z80 M1 pin low. The interface includes the necessary protection resistor, but use of this feature remains at the user's own risk. When MC45 is active, BASIC programs larger than 16KB cannot be loaded or edited.

11.4 Booting with an Alternative ROM

Alternative ROM files named 0.ROM to 9.ROM may be placed in /SYS/. To boot one of them, hold the corresponding number key while powering up the ZX81, then release it after start-up. If no key is held, the standard ZX81 ROM is used.

11.5 User-definable Characters - 128C / 64C

```
LOAD *128C
LOAD *64C
LOAD *LDIR 7680,15360,512
LOAD *LDIR 7680,15872,512
```

128-character mode allows all characters to be redefined by writing to memory addresses 15360 to 16383. The upper 64 characters appear in inverse video automatically. To display them normally, store the inverted glyph data in those positions.

NOTE: 128-character mode is incompatible with the internal HRG character generator. The two modes cannot be active at the same time.

```
5 LOAD *128C
20 LOAD FAST "SPEC-81-128.BIN" CODE 15360
30 CLS
35 POKE 16418,0
40 PRINT AT 23,1;CHR$ 8;" 1982 S[INCLAIR] R[ESEARCH] L[TD]. "
45 POKE 16418,2
50 IF INKEY$="" THEN GOTO 50
```

12. Advanced BASIC Extensions

12.1 String Manipulation

```
LOAD *INV A$
LOAD *INV A$(2 TO 7)
LOAD *BOLD A$
```

*INV toggles bit 7 of characters in a string, changing normal characters to inverse video and inverse characters to normal. *BOLD forces bit 7 to 1, producing inverse video.

12.2 Copying and Filling Memory Blocks

```
LOAD *LDIR <source>,<destination>,<length>
LOAD *LDDR <source>,<destination>,<length>
LOAD *HEX <address>,"<hexadecimal>"
```

*LDIR copies memory upwards and is equivalent to Z80 LDIR. *LDDR copies downwards and is equivalent to Z80 LDDR. Use the correct direction when source and destination overlap. *HEX loads hexadecimal byte values directly into memory.

12.3 Executing Machine Code

```
LOAD USR <address>
```

LOAD USR calls a machine-code routine like RAND USR, but does not disturb the random number generator, enters from the top level of BASIC, leaves the alternate BC, DE and HL registers available, and allows the routine to parse the rest of the line itself. Register BC contains the called address on entry.

12.4 Input/Output Ports

```
LOAD *OUT <port>,<value>
LOAD *IN <port> TO <variable>
```

12.5 16-bit Memory Access

```
LOAD PEEK <address> TO <variable>
LOAD THEN POKE <address>,<value16>
LOAD THEN CLEAR <address>
```

LOAD PEEK reads two consecutive bytes as a 16-bit value. LOAD THEN POKE writes a 16-bit value. LOAD THEN CLEAR sets the BASIC upper memory limit without clearing variables; it only clears the GOSUB stack.

12.6 Directory Access from a Program

```
LOAD *OPENDIR <string>
LOAD *ROW <number> TO <variable$>
```

*OPENDIR prepares a directory listing, including wildcard searches. *ROW reads one entry from that listing into a string variable. This allows BASIC menus to be built from SD card contents.

```
10 LOAD *OPENDIR "GAMES/*.P"
20 FOR I=1 TO 10
30 LOAD *ROW I TO A$
40 IF A$="" THEN GOTO 70
50 PRINT I;" ";A$
60 NEXT I
70 INPUT "Select: ";N
80 LOAD *ROW N TO A$
90 LOAD FAST A$
```

13. Example Programs

The following examples demonstrate the main functions of the SD81 Booster. They are written in standard ZX81 BASIC with the additional interface commands.

13.1 Real-time Clock

```
10 LOAD *RTC TO R$
20 PRINT "DATE AND TIME: ";R$
```

13.2 RTC Battery Status

```
10 LOAD *BAT TO B$
20 PRINT "BATTERY: ";B$
```

13.3 Checking MC45 Mode

```
10 LOAD *MC45
20 PRINT "MC45 MODE ENABLED"
30 LOAD *MC45 STOP
```

13.4 Superfast Mode Demonstration

```
10 POKE 16418,0
20 FOR I=1 TO 1000
30 PRINT I
40 NEXT I
50 POKE 16418,2
```

13.5 Loading a Spectrum-style Screen

```
10 LOAD *128C
20 LOAD FAST "SCREEN.BIN" CODE 15360
30 POKE 16418,0
40 PAUSE 0
50 POKE 16418,2
```

14. Error Codes

Code	Meaning
0	OK - no error
1	File not found
2	Directory not found
3	Invalid filename
4	Access denied or unsupported operation
5	SD card I/O error
6	Invalid command syntax
7	Out of memory or buffer overflow
8	Unsupported file format
9	Internal or critical error

TECHNICAL REVIEW: Exact firmware return codes may change. For production release, verify this table against the firmware source and compatibility test suite.

15. For Programmers

15.1 Expansion ROM Map

The expansion ROM occupies the 2000h-3FFFh address range. It provides command parsing, filesystem entry points, MCU communications and support routines. Applications should use documented commands wherever possible instead of relying on private ROM entry points.

15.2 I/O Ports and MCU Protocol

Communication between the Z80 and the microcontroller is performed through interface I/O ports. The protocol is command based: the Z80 sends a command identifier and parameters, and the MCU returns status, data or an error code. Timing-sensitive operations may synchronise with VSYNC where required.

NOTE: Programmers should avoid hard-coded delays and should poll the documented status flags instead.

15.3 Memory Mapper - Port E7h

The memory mapper is controlled through port E7h. It selects which 8KB RAM page is visible in each 8KB block of the Z80 address space. Correct mirroring of blocks 6 and 7 must be preserved during video operation.

15.4 Complete MCU Command Table

Group	Purpose
System commands	Version, reset, status and general control.
Filesystem commands	Open, read, write, delete, rename, copy and directory access.
Hardware control	RAM mode, mapper control, joystick mapping and video-related control.
Speech synthesis	Queue and play speech phonemes or parsed English text.
AY / sound	PLAY command, direct AY access and envelope control.
VGM	Load, start, pause, continue, stop and loop control.
PEG	Load effect programs and control PEG threads.
RTC and battery	Read and set date/time and query CR2032 battery status.

16. Troubleshooting

16.1 Interface Does Not Boot or ZX81 Locks Up

- Power off immediately and check the expansion connector alignment.
- Check that no ROM-replacement device is fitted.
- Remove other expansion hardware and test the SD81 Booster alone.
- Verify that the SYS directory is present on the SD card.

16.2 microSD Card Problems

- Use FAT32 only. exFAT and NTFS are not supported.
- Avoid long or unsupported filenames.
- Check that the card can be read on a modern computer.
- Try a different card if SD access errors continue.

16.3 RTC Loses Time

- Run LOAD *BAT to check the CR2032 battery state.
- Replace the battery with a standard CR2032 cell if necessary.
- Set the clock again using LOAD *RTC="YYYY-MM-DD HH:MM:SS".

16.4 Joystick Does Not Respond

- Check that the joystick is a standard 9-pin Atari/Commodore-compatible type.
- Reconfigure the key mapping with LOAD *JOY.
- Check the controls required by the loaded game.

16.5 No Sound

- Check the audio path and display connection.
- Confirm that the command being used is supported by the installed firmware.

- Test with a simple LOAD *PLAY command or a known compatible WAV/VGM file.

17. Firmware Updates

17.1 Updating the Microcontroller Firmware

The USB-C connector is used for programming the microcontroller and for serial debugging. Firmware update procedure depends on the firmware release package. Use only files provided by the SD81 Booster project or by RancanuoTeam.

WARNING: Do not disconnect power or USB during a firmware update.

17.2 Updating the FPGA/CPLD

The programmable logic device may also be updated when new hardware features or bug fixes are released. Follow the update instructions supplied with the release package and verify the installed firmware version afterwards with LOAD *VER.

18. Glossary

Term	Meaning
AY	AY-3-8910/12 sound chip family.
BASIC token	A keyword entered as a single ZX81 token rather than typed letter by letter.
Chroma81	RGB colour video interface for the ZX81.
FAT32	Filesystem required for SD cards used with the interface.
MC45	Mode allowing machine code to execute correctly in memory blocks 4 and 5.
PEG	Programmable Effects Generator, an internal sound-effect virtual machine.
RTC	Real-time clock.
VGM	Video Game Music file format, supported when it contains AY data.
ZX Printer	Original Sinclair thermal printer for the ZX81.

19. Firmware Version History

Version	Notes
v1.0	Initial public manual edition. SD loading/saving, AY sound, speech, RTC, RAM paging, joystick and Chroma81-compatible features documented.

EDITORIAL NOTE: Add dated firmware release notes here before publication.

20. References

- SD81 Booster Project documentation and firmware source.
- Sinclair ZX81 BASIC and hardware documentation.
- AY-3-8910 / AY-3-8912 sound chip references.
- SP0256 speech synthesis references.
- Chroma81 interface documentation.
- Zilog Z80 CPU documentation.

Appendix A - Complete PLAY Command Reference

PLAY strings contain note, timing, octave, tempo, repeat and envelope commands. Each PLAY string controls one AY channel; up to three strings may be supplied.

Parameter	Meaning
C D E F G A B	Musical notes.
= before note	Sharp, raises one semitone.
Pound sign before note	Flat, lowers one semitone.
Pound sign alone	Rest or silence.
1-12 before a note	Duration value, retained until changed.
T<number>	Tempo in beats per minute, 60-240.
O<number>	Octave, 0-8.
(...)	Repeat section once.
) without matching (	Repeat the whole string indefinitely.
H	Halt PLAY.
W0-W7	Select AY envelope effect.
U	Use envelope effect on this channel.
X<number>	Envelope speed. Higher values are slower.

Appendix B - PEG Effects Generator Reference

PEG programs are written as hexadecimal instruction words and loaded into PEG memory. Each instruction occupies two bytes, represented as four hexadecimal characters. PEG is intended for developers who require sound effects independent of the ZX81 CPU.

```
LOAD *PEG 0, "001200340056"  
LOAD *PEG THEN RUN 0,0  
LOAD *PEG THEN STOP 0
```

Concept	Description
PEG memory	256 byte instruction/data area used by the effect engine.
Thread	One of three parallel execution contexts, numbered 0, 1 and 2.
AY register access	PEG instructions can alter AY registers directly.
CPU load	Effects run independently of the Z80.

Appendix C - Speech Synthesiser Dictionary

The synthesiser scans the input text from left to right and always tries to match the longest possible entry in its internal dictionary. Recognised words sound better than words built syllable by syllable.

Input	Effect
Space	Short pause
,	Medium pause
; or :	Long pause
Upper-case English words	Parsed using the internal dictionary and phoneme rules
Numbers	Spoken automatically in English
Phonetic spelling	May improve pronunciation of names and non-dictionary words

NOTE: The phoneme set is fixed inside the microcontroller. User-defined phonemes or custom samples cannot be added.

Appendix D - Memory Paging System

The 512KB RAM is divided into 64 pages of 8KB. The Z80 address space is divided into 8 blocks of 8KB. The mapper assigns a RAM page to each writable block.

Rule	Reason
Blocks 6 and 7 mirror blocks 2 and 3	Required by the ZX81 video system.
Page numbers are 0-63	64 pages x 8KB = 512KB.
Block numbers are 0-7	8 blocks x 8KB = 64KB Z80 address space.
Use LOAD *MAP for BASIC access	Avoids direct hardware programming for normal use.

ROM modification and full 512KB paging should be attempted only by experienced Z80 programmers who understand the ZX81 display system and memory contention behaviour.

Appendix E - Chroma81 Interface Port (7FEFh)

The SD81 Booster is compatible with the Chroma81 RGB video interface. The Chroma81 control port is 7FEFh. Software may use this port to control colour-related functions where supported.

Operation	Description
OUT 7FEFh	Writes border and colour control information.
IN 7FEFh	Reads Chroma81 state where supported.
Bits 2-0	Border colour in GRB format.

TECHNICAL REVIEW: Verify exact bit assignments against the Chroma81 technical documentation before writing production software.

Appendix F - Superfast and Spectrum Modes

On the original ZX81, the CPU normally participates heavily in generating the video display. This reduces available processing time. Superfast mode frees the CPU from much of this work, increasing execution speed when normal display generation is not required.

Mode	Description
Standard ZX81 video	Normal display timing and compatibility.
Superfast mode	Frees CPU time for faster execution.
Spectrum mode	Provides a Spectrum-style display environment for compatible effects and demonstrations.
Border control	Controlled through documented POKes and Chroma81-compatible functions.
VSYNC synchronisation	Used for timing-sensitive display updates.

```
POKE 16418,0   REM Superfast mode
POKE 16418,2   REM Standard video mode
```

NOTE: Programs using video timing should synchronise with VSYNC and restore the standard display mode before returning to BASIC.

Release Readiness Notes

This English edition is a complete chapter-by-chapter technical draft prepared in British technical English with a Sinclair-era documentation tone. Before public release, verify firmware error codes, MCU command numbers, final hardware photographs and any late changes to firmware behaviour.