Spectron Public

Notifications Fork 0 Star 4

main 3 Branches 6 Tags Go to file Code

voytas Update package versions and dependencies 15aeb68 · 3 weeks ago 670 Commits

.github/workflows	Exclude .github and docs directories from workflow trigger p...	5 months ago
docs	Update CommandLine.md to document full-screen mode opt...	6 months ago
src	Fix publish issue	last month
tests	Fix UIA+ border colour handling (#71)	last month
.gitignore	Add AI instructions and PR review workflow	5 months ago
AGENTS.md	Add AI instructions and PR review workflow	5 months ago
CLAUDE.md	Add AI instructions and PR review workflow	5 months ago
Directory.Packages.props	Update package versions and dependencies	3 weeks ago
GemFile	Configure Jekyll page generation	11 months ago
GemFile.lock	Bump concurrent-ruby from 1.3.6 to 1.3.7 (#58)	3 months ago
LICENSE	Update LICENSE	5 months ago
README.md	CPU overclocking (#63)	5 months ago
Spectron.sln	Move Spectron.Files project directly to the solution instead of...	5 months ago
Spectron.sln.DotSettings	Migrate to new solution file type 'slnx'	9 months ago
_config.yml	Configure Jekyll page generation	11 months ago
docs_gen.sh	Add script to serve Jekyll locally	11 months ago

README MIT license

CI Build and Test passing

Spectron — ZX Spectrum 16/48/128/TC2048 Emulator

Here is Spectron, my personal ZX Spectrum emulator written in C# and Avalonia UI. It emulates the classic ZX Spectrum 16K, 48K, 128K and Timex Computer 2048.

It is accurate and stable, capable of running most games and demos without issues, including support for loading protected tapes.

Spectron is a cross-platform emulator that runs on Windows, Linux, and macOS. Developed primarily on macOS, it is well-tested on that platform, while also showing solid performance on Windows. Further testing on Linux is ongoing.

This project is a hobby I have always wanted to pursue. It has been both a lot of fun and a rewarding challenge. While many other emulators exist, this is my own personal take.

The ZX Spectrum was my first computer, and I still have a deep affection for it. I plan to continue maintaining and evolving this project, as I use it personally to enjoy classic games and demos.



Features

- Emulation of classic machines: ZX Spectrum 16K, 48K, 128K and Timex Computer 2048
- Time Machine: Rewind and continue from any point in the past
- State persistence: Emulator state is automatically saved and restored on restart
- Wide format support: SNA, SZX, Z80, TAP, TZX, RZX, MDR, TRD, and SCL (including ZIP archives)
- Support for POK trainer files
- RZX playback (recordings of ZX Spectrum games)
- High accuracy: Precise timings, including memory and I/O contention
- Floating bus support
- Multicolor screen effects
- ULA+ support
- AY-3-8912 sound chip emulation
- DivMMC emulation
- Beta 128 Disk emulation
- Microdrive emulation
- Kempston mouse emulation
- ZX Printer emulation
- Integrated tape and disk viewers
- Adjustable emulation speed (frames/second)
- Adjustable clock speed (3.5, 7, 14 and 28 MHz)
- Keyboard-based joystick emulation: Kempston, Sinclair, Cursor, and Fuller
- Audio and video recording
- Support for alternative and custom ROMs
- Built-in debugger
- Favorites manager
- Screen effects: Blur and CRT/Scanlines
- TC2048 extended graphics support (hi-res and hi-color)
- And more to come...

Spectron relies on several custom libraries developed specifically for this project:

Library	Description
Z80	Generic Z80 CPU emulator
Files	Handles TZX, Z80, SNA and other file formats
Beeq	Basic audio player, cross-platform, native
Inpeed	Gamepad handler, cross-platform, native

The solution consists of several projects:

Project	Description
Spectron	Avalonia-based UI and main application logic
Spectron.Debugger	Fully featured code debugger, including UI and controls
Spectron.Emulation	Core emulation engine
Spectron.Basic	Basic interpreter and related utilities
Spectron.Disassembly	Z80 disassembler, used by the debugger
Spectron.Recorder	Audio and video recording functionality

Quick demo



Releases

There are some test releases that can be found in the project Releases if you don't want to build the project. These are self-contained and do not need .NET Framework to be installed separetely.

Running the emulator using source code

Requires .NET 10 to build and run the emulator. Grab the latest code from the repository, build and run the emulator:

```
git clone https://github.com/oldbit-com/Spectron.git

cd Spectron

dotnet build -c Release
dotnet run --project ./src/Spectron
```

Command line options

Command lines allow overriding most of the default options and loading a specified file. Full list of available commands is here.

Testing and compatibility

- Passes Rustepy v0.33 (RAMSOFT) Rusting bus test in both 48k and 128k mode
- Passes HALT2INT v3 (Mark Woodmas) test in both 48k and 128k mode
- Passes EIHALT (Mark Woodmas) test in both 48k and 128k mode
- Passes Super HALT (Invaders (Mark Woodmas) test
- Passes btime (Jan Bobrowski) test
- Passes prime (Patrik Rak) test

Test results can be found in the Results directory. Test programs can be found in the Tests directory.

CPU Emulation

I have developed a custom Z80 CPU emulator library for this project. The emulation is highly accurate and supports many undocumented instructions, as well as memory and I/O contention.

Emulation Speed

There are two options for adjusting the speed:

- **At n% of normal speed** - this option allows running the emulator at a higher or slower speed, e.g. more or less frames per second. Everything will run at selected speed, e.g. like using fast-forward or slow-motion playback.
- **At higher CPU clock** - this option runs the CPU at a higher frequency. Standard clock is 3.5 MHz, but it can be increased to 7 MHz, 14 MHz or 28 MHz. This can be useful for games that are normally sluggish. Some software and games may not work properly with higher clock speeds.

Session Persistence

When the emulator is closed, it automatically saves its current state, which is then restored upon restart. This behavior can be toggled in the settings.

Tape Loading

Tape loading is supported for **TAP** and **TZX** files (including ZIP archives). Three loading speeds are available:

- **Normal**: Loads the tape at the original speed, including border and audio effects.
- **Accelerated**: Loads the tape at the maximum possible emulator speed.
- **Instant**: Loads the tape instantly into memory, bypassing the loading sequence. This mode works for files using standard ROM loaders.

Tape Saving

Tape saving is supported for **TAP** and **TZX** formats at two speeds:

- **Normal**: Saves at the standard speed with border and audio effects.
- **Instant**: Saves instantly by reading the current memory state.

Tape Browser

The Tape Browser allows you to inspect the contents of the currently loaded or saved tape. You can view blocks, their types, and basic information. Selecting a block allows you to load it using the standard `LOAD ""` command.

Snapshots

The emulator supports saving and loading snapshots in **SNA**, **SZX**, and **Z80** formats. It also features a custom **.spectron** format, which is recommended as it captures most emulator settings.

Screen

Multicolor screen effects and border effects are fully supported. The border size can be adjusted in the settings. Additionally, screen effects like **Blur** and **CRT/Scanlines** can be applied to simulate an old TV feel.

Floating Bus

Floating bus emulation is supported for both 48K and 128K modes, which is required for a small number of specific games.

Alternative ROMs

Spectron allows you to select from various built-in alternative ROMs, such as TR DOS and BBC Basic (which require 128K mode). Custom ROM files can also be loaded.

Audio

Standard beeper audio and AY-3-8912 sound (mono or stereo ABC/ACB modes) are supported. AY is enabled by default in 48K mode but can be disabled.

Audio playback is powered by Beeq, a custom cross-platform audio library developed for this project.

Joystick and Gamepad

Emulation is supported for Kempston, Sinclair, Cursor, and Fuller joysticks. External gamepads and joysticks are also supported (tested with a variety of controllers). Buttons can be mapped to joystick or keyboard keys. The standard keyboard can also serve as a joystick, using the arrow keys for directions and Space for Fire.

> **Note**
> Controller compatibility may vary by platform. This is an experimental feature.

ULA+ Support

ULA+ mode is supported and can be enabled in the settings.

DivMMC

DivMMC emulation is supported and can be enabled in the settings. It is based on esxDOS 0.8.9 and requires a disk image containing the esxDOS system files. Sample disk images are available here.

Up to two SD card images are supported, and changes can be persisted to the images. RTC support is available via the RTC.SYS file, allowing DivMMC to use the current system date and time.

Microdrives

Microdrive emulation supports up to eight drives and can be enabled in the settings. MDR image files can be loaded and saved. Note that other Interface 1 features are not currently emulated.

Beta 128 (TR-DOS)

Beta 128 Disk Interface emulation supports up to four drives (TRD and SCL formats). It uses ROM version 5.03. To invoke TR-DOS, use `RANDOMIZE USR 15616` from BASIC, or use the Pentagon 128 ROM for easier access.

Time Machine

The Time Machine feature allows you to rewind the emulation and resume from a previous point in time. The recording interval and history depth are adjustable in the settings.

Video and Audio Recording

Audio and video recording are supported. This is an experimental feature and performance may vary by platform. Changing emulator settings during recording may lead to unexpected results.

Audio

Audio is recorded in **WAV** format (PCM 16-bit, 44,100 Hz, mono or stereo). No external dependencies are required.

Video

Video recording requires FFmpeg. Videos are encoded as **MP4** using the **H.264** codec at **50 FPS**. Rendering options like scaling and border size are adjustable.

Recording is processed in the background after it stops, leveraging FFmpeg to combine the captured frames and audio.

Debugger

The built-in debugger allows for inspection of CPU registers, memory, and disassembly. It supports code stepping and breakpoints. For more details, see the Debugger documentation.

Favorites Manager

The Favorites Manager helps organize your favorite games, demos, and files, accessible directly from the main menu. For more details, see the Favorites documentation.

RZX Playback

Playback of RZX files (recordings of ZX Spectrum games) is supported. You can find many RZX files in the RZX Archive.

Resources

- Avalonia UI
- FFmpeg wrapper
- SkiaSharp
- Material Icons
- ZX Spectrum Font by Patrick H. Lauke
- Hack Font
- VT320 Font

About

Cross platform ZX Spectrum emulator written in C#. Accurately emulates classic ZX Spectrum computers: 16K, 48K and 128K models. Now also emulates Timex Computer 2048.

spectron.oldbit.com

beta-128 · divmmc · dotnet · emulator · microdrive · sinclair · sinclair-zx-spectrum · tc2048 · timex-computer-2048 · tr-dos · zx-spectrum · zx-spectrum-emulator

Readme · MIT license · Activity · Custom properties · 4 stars · 1 watching · 0 forks

Report repository

Releases 3

v1.0.1 Latest — last month

+ 2 releases

Contributors 2

voytas Wojciech Sciborek

dependabot[bot]

Languages

C# 99.6% · Other 0.4%