

Relay Trainer

[Introduction](#)[History](#)

Design

[Circuit Design](#)
[Architecture](#)
[Conditional](#)
[Logic](#)
[Semiconductors](#)
[Instruction Set](#)

Usage

[Keypad/Display](#)
[Serial Console](#)
[Example](#)
[Programs](#)[Software Tools](#)
[Build/Dev Log](#)

[Project page](#)
[Comments](#)**Single Board Relay Computer "Trainer"**

The Single Board Relay Computer (or Relay Trainer) is a computer whose CPU is made out of electromechanical relays (like [Konrad Zuse's Z3](#) from 1941), but with memory and front panel interface built from modern semiconductors (like the 1970s and 80s single board "trainer" computers, such as the [KIM-1](#)).

Yes, there are videos! Check out the [example programs page](#).

Want to buy one? I'm sold out of fully assembled computers, but I'm selling bare boards (including programmed microcontrollers) for you to source your own components and hand assemble. The component list (BOM), schematics and assembly instructions are in the files section of the project page. The bare boards are available here:

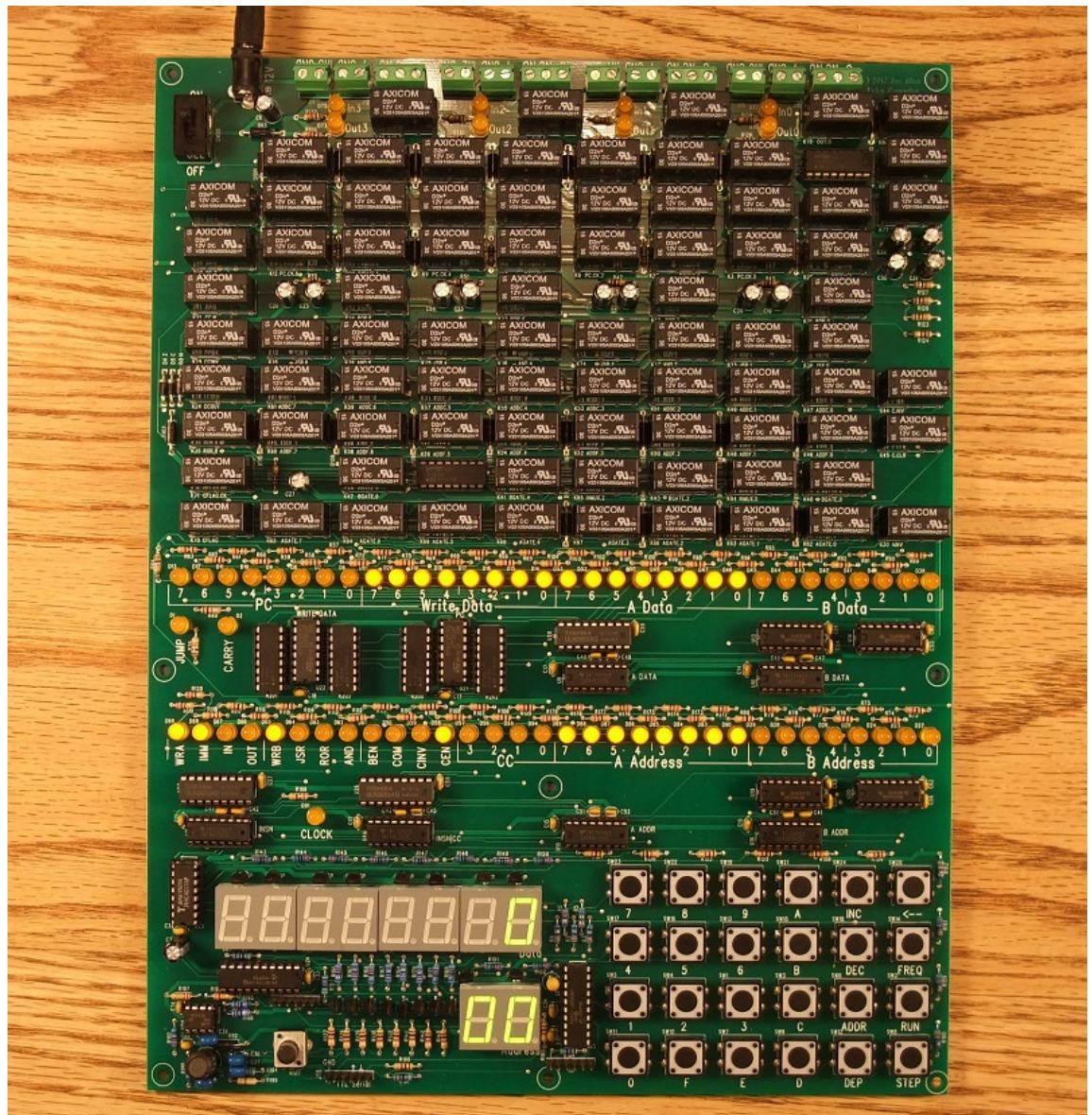
[eBay Bare Board Listing](#) or [Tindie Store](#).

2017 Update

After a long delay, a third version and run of boards has been produced. This new version improves the manufacturability and serviceability of the computer. With this new design, 8 out of 10 boards immediately worked, and the remaining two were easily diagnosed and repaired.

- All relays and integrated circuits are now socketed.
- All relay pins now have top-side test points. The relays are spread further apart to provide space to allow easy access to the test points.
- Relays are labeled by function as well as by reference designator. This makes finding a particular bit easier than having to refer to the schematic.
- The previous design required a 12V / 5V hard drive power supply. This one uses a single 12V supply provided through a conventional 5.5mm barrel power connector (the 5V DC-DC converter is on the board).
- Mounting holes are now included in the PCB design for standoffs or an eventual enclosure- perhaps a wooden box.

Here is the new version:



[Click for high resolution](#)

Relays are expensive and large, so primary importance is given to keeping the relay count low. The basic strategy is implement just the CPU and some basic I/O using relays, but implement the memory and front panel user interface with semiconductors. The CPU is designed using low cost DPDT DIP relays. The front panel and memory are implemented using Microchip PIC microcontrollers.

Architectural and circuit design tricks keep the relay count for the CPU low. The Relay Trainer uses edge-triggered D flip-flops implemented with just 1.5 DPDT relays / bit. Multi-port memory reduces the number of required register bits. Self-modified code (borrowed from very early computers) keep the number of registers and logic gates needed to an absolute minimum while still maintaining an interesting instruction set.

Here are some of the basic characteristics of the Relay Trainer:

- Relay count: 83
- Datapath width: 8 bits
- Instruction width: 32 bits
- Instruction type: two-address (each instruction specifies two memory locations and the result is written back to one of them)
- Memory size: 256 32-bit words (1 KB)
- Registers: 13 bits total including 8-bit Program Counter, 1-bit carry flag, 4-bit output register
- Performance: one cycle per instruction, up to ~12 Hz for ~12 instructions per second
- I/O: 4 switch closure inputs and 4 relay contact outputs brought to screw terminals. Programs can also read from keypad and serial console and write to serial console.
- Supported concepts:
 - Subroutines (supported with self-modified code: the return address is inserted into a jump instruction)
 - Indexing (supported with self-modified code: the index is inserted into a memory reference instruction)
 - Multi-precision arithmetic
 - Multiply and Divide implemented as subroutines

- Complete set of conditional jumps (although there are no Zero or Negative flags, a memory word can be tested for zero and negative)
- Single-instruction loops with "incjne" (increment and jump if not equal to zero) instruction

Other basic features:

- Keypad and LED display allows you to:
 - Display and modify memory
 - Halt / run and single step
 - Vary clock speed (15 pre-programmed speeds or use knob)
- Discrete LEDs
 - Show current 32-bit instruction given to CPU
 - Show two 8-bit operands fetched from memory
 - Show current program counter value
 - Show 8-bit data being written back to memory
 - Show carry flag and condition logic result
 - Show input and output port bits
 - Show the clock
- Powered by 12V through 5.5mm barrel connector.
- 512 bytes of Non-volatile memory is provided to save and restore the RAM
- Serial port provided for 9600 serial console
 - Built-in assembler, disassembler and instruction trace display
 - Load or save programs by cutting and pasting into terminal emulator window
 - Provides 6-pin Arduino logic-level RS-232 header for use of widely available USB to TTL RS-232 converter cables
- The relay clock is stopped when the CPU halts: this saves the life of the relays. The clock resumes if you press the step or run buttons.

Homebuilt CPUs WebRing

[Ring-Home](#)
[Previous](#)
[Next](#)
[Random](#)

JavaScript by [Orien Dhacela](#)

Join the ring?

To join the Homebuilt CPUs ring, drop me [a line](#), mentioning your page's URL. I'll then add it to the list. You will need to copy this code fragment into your page.

Note: The ring is chartered for projects that include a home-built CPU. It can emulate a commercial part, that's OK. But actually using that commercial CPU doesn't rate. Likewise, the project must have been at least partially built: pure paper designs don't rate either. It can be built using any technology you like, from relays to FPGAs.