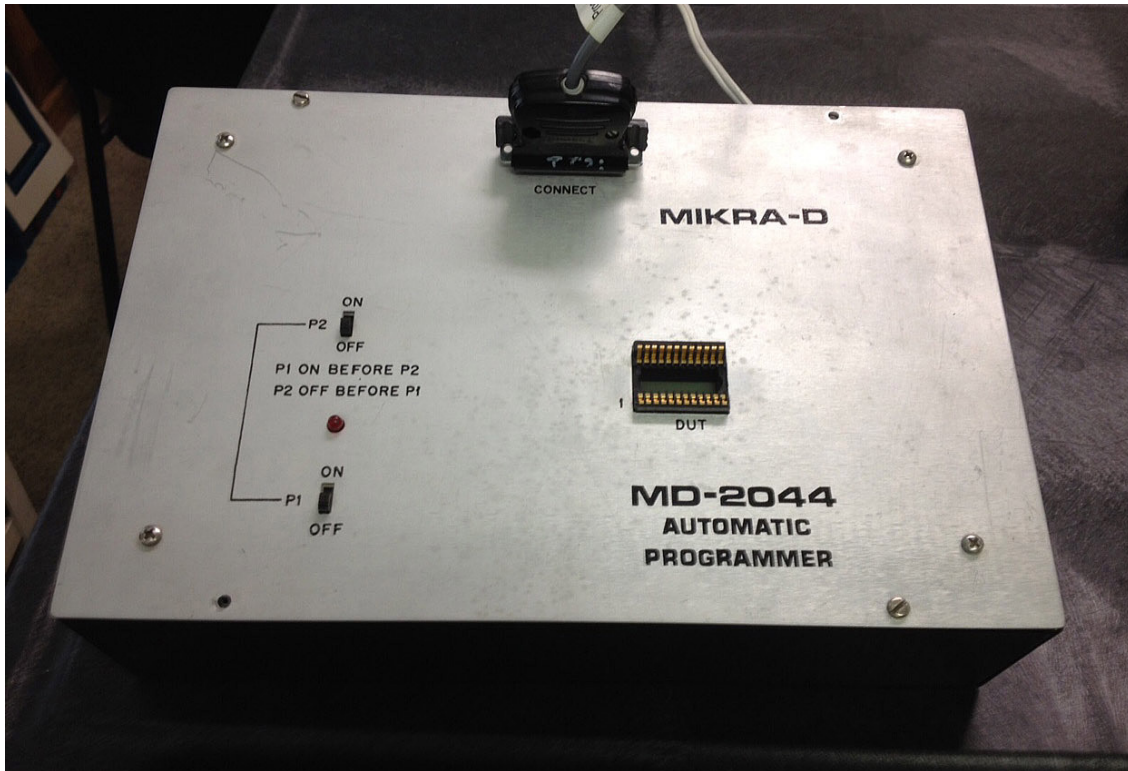


## Mikra-D MD-2044 1702A EPROM Programmer

Shown below is a picture of the MD-2044 1702A EPROM programmer from Mikra-D. This is a stand-alone programmer that requires only a serial port connection to a host computer to program or read a 1702A EPROM. All critical timing is performed by the programmer. This programmer is from around 1976.



### Initial Power-On

Before initial power-on, I opened the unit and had an oscilloscope ready to look at the condition of the various power supplies on the internal board. Unfortunately, smoke and odor ensued within the first few seconds of powering on the unit.

I looked first at the large electrolytic filter caps. None of them showed any external signs of leakage or failure, but I found the 1000uf cap for the +12v supply was a dead short. This short, in turn, drew enough current through two 1N4002 rectifier diodes to damage them. I replaced both of the 1000uf caps with new ones just to be safe. I also replaced the two damaged 1N4002 diodes. After this, power supplies look good (+12, +5, -12, +80).

### Serial Data Cable

The connector on the programmer is wired as DTE. When connecting to DCE computer (e.g., the typical Altair port wiring), then:

- Pin 2 is host data in and programmer data out.
- Pin 3 is host data out and programmer data in.
- Pin 7 is host and programmer signal ground.

The programmer requires shorting of the following DB-25 pins at the programmer's DB-25 connector: 12-13, 18-19, 21-24. The cable supplied with the programmer has these pins shorted.

This same cable swaps 2 & 3 for direct connection to a DTE host. A typical DCE Altair will require a null modem to un-swap 2 & 3. Note to self: The custom cable for my EP-1 programmer works well as a "null modem" between the Altair and the cable that came with the programmer.

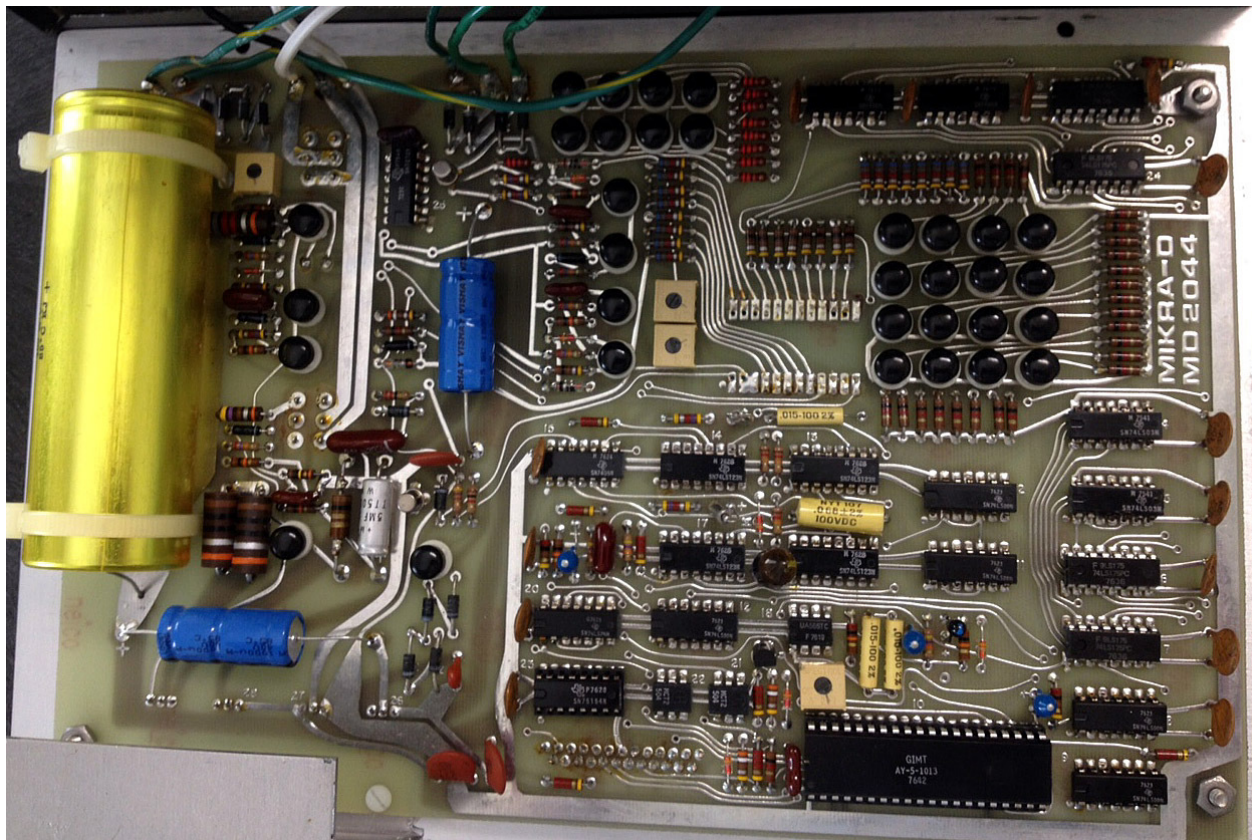
The internal PCB flexes a lot when inserting the DB-25 into the programmer. I pushed it on with the unit open so I could support the PCB with my hand. Probably best just to leave the cable plugged in all the time.

### Serial Data Timing

The programmer communicates at 110 baud, 8N2. The baud rate is generated with a 555 timer. The UART requires a clock at 16x the bit rate, or 1760hz. A pot near the 555 and the UART adjusts the frequency. The frequency can be measured on pin 3 of the 555 or pin 40 of the UART.

### Other Adjustments

Three additional pots are present on the board that allow adjustment of the 48v pulses to the EPROM when programming, and two one-shot timing adjustments. Hand written notes on the schematics describe these adjustments.



## Programming Algorithm

The programming technique recommended in the MD-2044 manual is as follows: Two bytes are sent for each location to program. The first byte is the EPROM address and the second byte is the value to write. Once given a new location/data pair to program, the programmer self-times 32 write pulses of the data to the specified location. The byte is then read from the EPROM and transmitted back to the host. Software should check if the byte received matches the value sent, and if so, programming of the current byte is complete. Otherwise, the host should resend the same data to attempt to program the same location again for up to ten total tries per location.

A similar process is used to read the EPROM except the data value sent is always zero and the programming voltage (switch P2) should be left “off.” Unfortunately, a read takes the same amount of time per byte as a write (about one second per byte) since the programmer doesn’t actually differentiate between a read and write. This means that typically quick operations, like reading an entire EPROM, verifying blank, computing checksum, etc., take several minutes.

## Programming Software

I have ported Martin Eberhard’s Bytesaver programmer software for CP/M to support the MD-2044. The program is MD2044.COM. This full-featured program allows programming from Intel hex files or binary files, reading and saving EPROMs to files, blank checking EPROMs, modifying the program buffer for patches, etc.

**IMPORTANT:** Insert and remove the EPROM only while both P1 and P2 power switches are OFF.

The programming algorithm specified in the MD-2044 manual does not match the algorithm recommended by the part manufacturers. Instead of fully programming each location before moving to the next, Intel and other manufacturers recommend that each consecutive location in the EPROM should be written once, cycling through all 256 locations over and over, until all locations match the data written. If this first step took N cycles through the EPROM, the EPROM should then be over-programmed  $4*N$  more times.

Unfortunately, the MD-2044 hardware does not lend itself to reasonably duplicating this algorithm. As a compromise, the MD2044 program follows the algorithm recommended in the MD-2044 manual, and once the part has “passed,” the entire chip is then over-programmed three more times.

## Resources

Schematics, operation notes, programming software (MD2044.COM) and this file can be found at [deramp.com](http://deramp.com) in the downloads directory (eprom\_programmers->mikra-d->md-2044).