

EPROM EMULATOR

USER'S MANUAL

MODEL EE512A

Version 1.2

9-25-92

Copyright (c) Technical *Solutions* 1991,92

INTRODUCTION:

Thank you for selecting a Technical *Solutions* product. We have made every attempt to provide a quality product at an affordable price. Our goal is to provide tools for the Engineer, Technician and Hobbyist that are inexpensive, but fully functional. If you have any problems or comments, please don't hesitate to call or FAX us and let us know.

The EE512 uses 64 KBytes of static RAM to emulate EPROMS up to 64 KBytes in size. An IBM compatible computer (PC) is used to down-load object code into the RAM. The RAM is dual-ported, allowing both the PC interface circuitry and the target to have access to it. When the PC starts to download object code, arbitration circuitry disables the target port and enables the PC interface port. While the download is occurring, the RESET pins are activated, holding the target in reset until its port is re-enabled.

INSTALLATION:

1. Turn off power to the target system.
2. Inset the EE512 into the target socket.

<p>BE CAREFUL TO INSERT THE EE512 PROPERLY. INSERTING THE EE512 BACKWARDS CAN RESULT IN PERMANENT DAMAGE TO THE EE512 OR THE TARGET SYSTEM!</p>
--

Pin 1 of the EE512 is identified by the small notch in the DIP plug, similar to an IC. Insert the EPROM Emulator so that this notch points the same direction as the notch on the target socket.

3. Select an un-used printer port on your IBM compatible PC/XT/AT.
4. Plug the supplied DB-25 to RJ-11/12 converter into the selected port.
5. Plug the other end of the RJ-11/12 cable into the EE512 jack labeled "IN".
7. For target systems with wider data paths or multiple banks of EPROMS, multiple EE512s can be daisy-chained. Up to four units can be connected for data paths up to 32 bits wide or for up to 4 banks of 8 bit EPROMs. The "OUT" jack of the first EE512 is connected to the "IN" jack of the next EE512. This is repeated up to three times. The Most-Significant-Byte of the Most-Significant bank is loaded into the first EE512 connected to the PC. The Least-Significant-Byte of the Least-Significant bank is loaded into the last EE512 in the chain. Therefor, the first byte in the object file will end up in the last EE512 in the chain. If four EE512s are daisy-chained in a 32 bit system, the second byte will be placed in the next-to-the-last EE512. The fourth byte will be placed in the first EE512 in the chain (the one connected directly to the PC).
8. (OPTIONAL) Connect a jumper between one of the reset pins on one of the EE512s to the appropriate place on the target if automatic reset is desired after each download.

RESET and /RESET are TTL/CMOS compatible outputs, capable of sourcing 25 milliamps and sinking 25 milliamps.

9. IF YOUR TARGET HAS IN-CIRCUIT EPROM PROGRAMMING CAPABILITY, DISABLE IT!

The EE512 is a 5 Volt ONLY device. Programming voltages will DAMAGE the device and will void the warranty. Even short surges during power-up or reset can be damaging.

10. Apply power to the target system.

11. Run the loader program. We wrote the loader as a command line driven routine rather than an interactive one. This allows you to run the program from a batch file without intervention. It can be added directly to your Compile-Link-Locate batch file. Enter "LD512" or "BLD512" without parameters to see the parameter syntax.

Most of the parameters should be self-explanatory, but a few clarifications are in order.

`/T:64/128` Pin 26 is defined as a No Connect on a 2764. Some systems will ground it or connect it to A13. This causes a mirror-image of the EPROM to appear immediately following it. To fully emulate this behavior, the emulator must either be strappable to ignore this line while emulating a 2764, or it must mirror the contents of the EPROM into the next 8KB block. We chose to do the latter. This has two advantages. It keeps the cost down, and it eliminates any need to manually configure the EE512 for different EPROM sizes.

`/O:xxxxx` Absolute starting address of the EPROM(s) in HEX. EXAMPLE: If you are placing a 27C512 at the top of memory in an 8086 (1M - 64K = F0000), you would enter " /O:F0000". This parameter is used for Intel Hex or Motorola 'S' files. It has no effect for binary files. Binary files are assumed to be absolute binary images of the EPROM. Therefore, the address of the EPROM is irrelevant.

All parameters are optional and can be listed in any order. The Printer Port is specified by base address rather than LPTx. This allows the use of multi-port parallel cards that do not map directly into LPT1-LPT3. It also avoids some of the early XT clone BIOS incompatibility issues. Most BIOS/DOS compatible printer ports are located at one of the following I/O addresses: 378, 3BC, or 278.

If you need any additional technical assistance, we can be reached at:

Technical *Solutions*

P.O. BOX 462101

Garland TX. 75046-2101

Voice: (214) 272-9392 FAX: (214) 494-5814

ADDITIONAL NOTES

1. Most linker/locators will generate Binary files quicker than Intel Hex or Motorola 'S' files. In addition, the EPROM Emulator downloads Binary files quicker than the other two because it does not require conversions. Therefore, for the quickest download times, we recommend the use of Binary files. It will speed up the linker/locator and the download process.

With that in mind, we created a new program: BLD512.EXE. This program works with BINARY files only. Removing the Intel Hex and Motorola 'S' file conversion routines from the downloader eliminates the conversion time, the time for buffering an image of the data in RAM and the overhead of HUGE pointers. This results in a considerable decrease in download times. BLD512.EXE does NOT support multiple banks of EPROMs. It DOES support daisy-chaining for wider data paths. If you are emulating more than one bank, you must use the original loader; LD512.EXE.

2. The Eprom Emulator uses high quality machined pin headers for the target socket pins. However, they are not very forgiving to being bent. We recommend the use of a "sacrificial" socket between the Eprom Emulator and the target socket. This socket protects the Eprom Emulator from mechanical damage. If one of the socket pins becomes bent or damaged, the socket can be replaced a lot easier than the Eprom Emulator can be repaired. This is common practice for most test equipment that plugs into target sockets.
3. The RESET and /RESET outputs are driven by TRI-STATE devices. They are active during the download only.
4. The loader programs have no feedback from the Eprom Emulator. They can not tell if the Eprom Emulator is plugged in or if power is applied to the target (and therefore the Emulator).
4. The Emulator (and therefore the target) must be powered before the object code can be down-loaded into the Emulator. While the download is occurring, RESET and /RESET are active to reset the target. After completing the down-load, the reset lines are returned to their in-active states. If the target attempts to access the Emulator during down-loading (the reset lines were not connected) it will read whatever data is being down-loaded at that instant. Once object code is downloaded, it will stay valid until power is removed or new code is downloaded.

EPROM EMULATOR TROUBLESHOOTING TIPS

1. Verify that the EE512 is communicating with the host. The LED should turn off during the download, and back on after the download. If the LED stays on or off:
 1. Verify that the proper printer port address was specified
 2. Verify that the cable is connected properly
 3. Verify that the target is powered-up
 4. Try a different printer port
 5. Verify that the EPROM Emulator is plugged in properly

2. Try using a BINARY file if possible. Binary files are simpler and leave less room for mistakes in locating or specifying offsets.
3. Verify that your target socket is wired (or jumpered) correctly for the EPROM you are emulating. I know this seems obvious BUT ... Some board manufactures take some liberties in wiring "universal" EPROM sites. Many EPROMs are tolerant to specification violations on their VPP pins or PRG pins. These EPROMS will function even though the socket is not wired to specs. The EPROM Emulator is a little more sensitive to these violations because is always contains a full 64KB of memory that is driven by these pins. If the spec calls for a logic high on these pins during normal read operations, the EPROM Emulator expects it to stay high. Some boards will tie PRG or Vpp pins to address lines to save some configuration jumpers. Other times, the socket is jumpered for a larger part, but the smaller EPROM still functions. In particular, verify the following pins:

PIN	NAME	2764	27128	27256	27512
1	Vpp/A15	+5	+5	+5	A15
27	PRG/A14	+5	+5	A14	A14
26	NC/A13	FLOAT/+5	A13	A13	A13

SIGNALS FOR PROPER EPROM Emulator OPERATION

If any of these pins are wired differently than indicated above, the EPROM Emulator will not function properly.

In some cases, you can specify a larger EPROM type on the command line and get it to work. For instance, if you are emulating a 27256 and pin 1 is connected to A15 instead of +5 (the socket is actually wired for a 27512), you can tell the loader that you are emulating a 27512. This will load the file into the lower half of the emulator memory instead of the upper half. A better solution would be to fix the target wiring.

4. Remove any extension cables you may have added to the download cable or the target socket. If this makes the EE512 start to work, the problem may be an error in the wiring of the extension cable or noise induced by the cable. We have many customers succesfully using up to 25' of cable between the computer and the EE512. However, the longer the cable, the higher the probability of induced noise or cross-talk. Extension cables added to the target socket add a "STUB" to the address , data and control signals on the target. This can degrade the quality of these signals and cause malfunctions. These transmission line effects are a function of the physical extension of those signals and is not related to the design of the EE512 itself.
5. If you do buy a longer modular cable, be sure to purchase one that is wired the same as the one provided (6 conductor, pin1 to pin1). Note that modular cables can be purchased in straight-through or cross-over configurations.