

# EPROM EMULATOR

## USER'S MANUAL

### MODEL EE256

Version 1.2

9-04-92

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 EE256 uses 32 KBytes of static RAM to emulate EPROMS up to 32 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 EE256 into the target socket.

<p><b>BE CAREFUL TO INSERT THE EE256 PROPERLY. INSERTING THE EE256 BACKWARDS CAN RESULT IN PERMANENT DAMAGE TO THE EE256 OR THE TARGET SYSTEM!</b></p>
--

Pin 1 of the EE256 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 one end of the supplied RJ-11/12 cable into the adaptor.
6. Plug the other end of the RJ-11/12 cable into the EE256 jack.
7. (OPTIONAL) Connect a jumper between one of the reset pins on the EE256 to the appropriate place on the target if automatic reset is desired after each download. RESET and /RESET are TRI-STATED TTL/CMOS compatible outputs, capable of sourcing 25 milliamps and sinking 25 milliamps.

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

The EE256 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.

9. Apply power to the target system.

10. 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 "LD256" or "BLD256" 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 EE256 for different EPROM sizes.

`/O:xxxxx` Absolute starting address of the EPROM in HEX. EXAMPLE: If you are placing a 27C256 at the top of memory in an 8086 ( 1M - 32K = F8000), you would enter "`/O:F8000`". 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. Therefor, 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 parrallel 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.

## 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. Therefor, 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: BLD256.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, and the overhead of buffering an image of the data in RAM. This results in a considerable decrease in download times.

The old loader (LD256.EXE) is still included in case your linker/locator will not generate BINARY directly, and you have no way of converting the file yourself.

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 are tri-stated outputs. This allows you to connect them across reset switches or RC reset circuits. If you observe them with a scope or DVM, it may appear that they are not toggling unless they attached to a pull up or pull down. This is normal.

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 therefor the Emulator).

5. The EPROM Emulator cannot fully emulate an EPROM on initial power-up. The Emulator (and therefor the target) must be powered before the object code can be downloaded 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. This not normally a problem. All hardware and software can still be debugged except possibly the exact power up sequence of the circuit. Once the software is developed, it can be burned into the final EPROM and the RESET circuit can be fully tested.

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

## EPROM EMULATOR TROUBLESHOOTING TIPS

1. Verify that the RESET and /RESET outputs go active during the download and return inactive after the download. This verifies that the software is talking to the correct printer port and that the cable is connected correctly.
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 it always contains a full 32KB 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 be high during all valid reads. 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:

<u>PIN</u>	<u>NAME</u>	<u>2764</u>	<u>27128</u>	<u>27256</u>
1	Vpp/A15	+5	+5	+5
27	PRG/A14	+5	+5	A14
26	NC/A13	FLOAT/+5	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 27128 and pin 27 is connected to A14 instead of +5 (the socket is actually wired for a 27256), you can tell the loader that you are emulating a 27256. 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. We designed the EE256 for as much noise immunity as possible, but the longer the download cable, the higher the probability you will exceed the noise margins. We have many customers successfully using 25' or more of cable, but we cannot guarantee their proper operation.
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.