



## Summary

This app note tells how to generate a PROM that contains a C application program. Use the Fabius C compiler to compile the application.

## Description

1. First compile the application using either the "hammer" icon (if the application code fits in one 32K page), or the "bricks" icon (for larger applications).
2. All DIP switches should be OFF (with the exception of the "2COM" Switch #4, which may be ON or OFF depending on whether you use the secondary serial port). A DIP switch is OFF if the side towards the board edge is down.
3. Use the terminal program to send the <program\_name>.txt file to the QED-3 Board.
4. From the terminal window, type the command:  
`CFA.FOR MAIN PRIORITY.AUTOSTART`  
to install the autostart vector at the top of page 4; this vector will end up on PROM.
5. BEFORE resetting the board or typing main (which would start the application program), we'll make an image of the pages that hold the object code and autostart vector. We'll use Motorola S-record format for the dump; all prom burners accept this format.

Set your terminal to capture incoming text to a disk file using the "Receive Text File" menu item. A suggested file extension is \*.S2 to remind you that it is a motorola S2 record file.

IF YOU USED THE "HAMMER" ICON to compile your code (code is on page 4 only), then interactively type from the terminal:

```
HEX 0 4 DIN 0 8000 DUMP.S2
```

This dumps 32K of data from page 4 to the file.

IF YOU USED THE "BRICKS" ICON to compile your code, AND if you know that your code fits on page 4 and 5 only

(check the compiler's \*.map output file to see where code is located), then interactively type from the terminal:

```
HEX 0 4 DIN 0 FFFF DUMP.S2
```

This dumps 64K of data from pages 4 and 5 to the file.

(If your application requires more than two 32K pages, contact Mosaic for help.)

6. Close the file when the dump terminates, and use WinEdit to remove any extraneous file contents such as the DUMP.S2 command and Forth's "ok" prompts at the very beginning and the very end of the file. Save the modified file.

7. (OPTIONAL, but recommended.)

At this point, you can test whether the file you created will work properly on the QED Board. Carefully type the following commands interactively at the terminal:

```
NO.AUTOSTART
```

```
COLD
```

```
HEX
```

```
0 4 FFFF ERASE
```

```
0 4 RECEIVE.HEX
```

(If you misspelled the first item by mistake and your application started, you can "factory clean" the board using switches 5 and 6 to get rid of the autostart vector). These commands erase the prior code, and get ready to receive the newly created PROM file.

Now use the terminal program to send the newly created \*.S2 file to the QED-3 Board. When the file transfer is complete, reset the board or type

```
MAIN
```

and your application should start and run properly. If it does not, then either there is a problem with the application source code, or the .S2 file was not properly created.

8. Then send the \*.S2 file to a PROM burner and burn a 32K PROM (part No. 27C256) if you have a 1-page application, or a 64K PROM (part No. 27C512) if you have a 2-page application. PROMs must be 120 nsec or faster. Of course, you can always use a larger PROM than you need, up to a 128K size. (Beware of some new PROMs with different part numbers that are

designed to be compatible with either 3.3 or 5 volt systems; these parts put out a "high" voltage of only 3 volts, and this does not meet the processor's specifications.)

9. Turn off your QED Board and remove the RAM from socket S3. Turn DIP switch #3 ON to configure socket S3 to accept a PROM, and plug the programmed PROM into socket S3. Power up the QED Board and it will automatically run the application!

**NOTES:**

If you find that the QED Board on which you originally compiled your code works with a PROM, but other QED Boards don't work using the same PROM,

AND if you have battery-backed RAM in the center socket, there are probably initialization errors in the source code. That is, some variables are not being initialized at runtime, and the BBRAM is "remembering" the compile-time initialization. Remember that all variables and arrays must be explicitly initialized at runtime; there is no automatic runtime initialization performed.

If there are still problems getting the PROM to work, call Mosaic at 510-790-1255, or email us at [info@mosaic-industries.com](mailto:info@mosaic-industries.com).

The information provided herein is believed to be reliable; however, Mosaic Industries assumes no responsibility for inaccuracies or omissions. Mosaic Industries assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk.

## Mosaic Industries

5437 Central Ave Suite 1, Newark, CA 94560

Telephone: (510) 790-8222

Fax: (510) 790-0925