

## INSTRUCTIONS FOR PROGRAMMING CODE INTO EEPROM MEMORY

By D. J. Mehrl, 10/20/99 [for EE3362\_002, Fall '99]

Setting up Hyperterminal (under Windows 95) to insert a 10msec delay between characters:

- (1) Get into your existing Hyperterminal session, configured for the usual 9600 Baud rate, 1 stop bit, 8 data bits, no parity, no handshaking etc..
- (2) Choose "Properties" under the "File" menu.
- (3) In the dialogue screen that appears, click on the "Settings" tab, then click on the "ASCII SETUP" button.
- (4) In the dialogue box that appears, there should be a "text box" labeled "Character Delay" that normally contains a delay of 0 (milliseconds). Click in this box and type in "10". Then click the "OK" button. This should bring you back to the "Properties" dialogue window, where you should also click "OK". At this point you are ready to download your code to EEPROM just as you normally download (using the "load" command). Note that it takes considerably longer to download code into EEPROM (due to the 10msec delay between characters).
- (5) To avoid having to repeat the above procedure every time you need to program EEPROM, you may want to "save" this session, creating an icon that you can click each time you want to program EEPROM. You can now save your session by choosing "Save As" under the File menu. If you want the icon to appear on the desktop, use the file dialogue controls to select the "Desktop" as your file destination. You probably should name your file (or icon name) something like "PrgEEPROM" to distinguish it from the normal Hyperterminal Session which has 0 msec delay.

Note: The procedure for setting up Hyperterminal under Windows 2000 could vary slightly - I've never tried Hyperterminal under Windows 2000. Also, under Windows 3.1, you have "Terminal" instead of "Hyperterminal", but the procedure is more or less similar.

---

Special considerations for code intended to reside in EEPROM:

- (1) You must "org" your code to \$0D00 (start of EEPROM), rather than the usual \$0800 (start of user RAM). Don't try to declare RAM variables in EEPROM though, because your program cannot write to EEPROM locations! Variable locations (e.g. "count rmb 1" reserves 1 byte of RAM named count that your program can read or write to) should be org'd to somewhere in the \$0800 to \$09FF range where user RAM resides.
- (2) If you want to later run your EEPROM code from debug-12, simply type "g 0d00" at the Debug prompt. Later you may want to also run your EEPROM code in the so called "EEPROM" mode, where your program will be off and running on its own (i.e. no need to load the program from within Debug - that's the beauty of EEPROM - it's NONVOLATILE memory). To do this, you must also insert the following statements in your initialization code before assembling and programming your code into EEPROM. Note that it doesn't hurt to ALWAYS install these statements as a matter of routine. That way your code can be run in EEPROM mode later, if you should so decide.

```
copctl equ    $16
movb    #0,copctl
```

Now you can switch to EEPROM mode by simply moving the "W3" jumper from the "0" (or "EVB") position to the "1" (or "EE") position. This is described in more detail in the small spiral bound EVB Users Manual. Then either hit the reset switch or unplug and then plug your power supply back in to effect a "power on reset". Your code should start running immediately --- TOTALLY AUTONOMOUS OF DEBUG-12 and Hyperterminal.

Finally, if you start up your 68HC12 in "EEPROM Mode", Debug-12 will not initialize the SCI (RS232 serial) hardware - you must do this your self. Also, Debug-12 will not initialize an important RAM location necessary for the Debug-12 "getchar","putchar", "out2hex", "out4hex", etc. routines to work properly. If you need to utilize the SCI port in your EEPROM code, and you want to be able to do a "cold start" - i.e. power up already in EEPROM mode, then you should also add the following code to your initialization routine.

```
sc0bdh      equ      $C0
sc0cr2      equ      $C3
            movw     #52,sc0bdh      ;Set baud rate to 9600
            movb     #$0C,sc0cr2     ;enable SCI transmitter and receiver
            movw     #$00C0,$0ADB    ;Must do this so that getchar/putchar work
```