Auto-start a user application program from on-chip flash memory of the MiniDRAGON+ Rev. D board during power up

By Wayne Chu, 12/28/05

After your application program runs perfectly in RAM, you can program it into on-chip flash memory for auto starting during power up. At first you replace the D-Bug12 monitor with your application program in bootloader mode, and then set J9 and S7 for EVB mode. During power up, it would have run the D-Bug12 monitor, but now it will run your program. The flash memory is non-volatile memory like EEPROM. Your code will run every time the board is powered up or reset.

The most common mistake made by a user is to download an s1 file into flash memory.

In order to program the MC9S12DP256 flash memory, you must program an even number of bytes and begin on an even address boundary for each s-record. If any one s-record in the file contains an odd number of bytes or begins with an odd address, the flash memory cannot be programmed. Let's use the test_flash.asm as a sample program and assume that the assembler successfully generated the test_flash.s19. Next, you must use the Freescale's s-record conversion utility, sreccvt.exe, to make an s2 file from the test_flash.s19 by using the following command line:

```
Srccvt -m c0000 fffff  32 -of f0000 -o test_flash.s29 test_flash.s19
```

It will create a new s2 file named test_flash.s29 that has an even number of bytes and can be programmed into the flash memory. For your convenience, I made a batch file, make_s2_test_flash.bat that can be placed along with the srccvt.exe in your working directory. After assembling your test_flash.asm without an error, you can double click the make_s2_test_flash.bat and it will create test_flash.s29 from test_flash.s19. You can use this batch file on your other application programs and edit the make_s2_test_flash.bat by substituting the filename 'test_flash' with your new file name. For instance, if your new filename is demo.asm the dos command in the make_s2_test_flash.bat will be:

```
Srccvt -m c0000 fffff  32 -of f0000 -o demo.s29 demo.s19.
```

To edit the make_s2_test_flash.bat, right click it and a small pull down menu will appear on screen. Left click the edit option to modify the dos command.

Here are the steps to program the flash memory with test_flash.asm for auto-starting:
(For your convenience, I have made steps 1 - 4 for you and you can use the test_flash.asm as a template for modifying your assembly source program. By comparing our original test.asm and the new test_flash.asm you can see what have to be done for modifying your source program)

1. Our original test.asm was made for running in RAM and it must be modified by changing the starting address from RAM to flash memory.

2. The interrupt vectors table addresses must be moved to the secondary vector table addresses at $EF80-$EFFF. The addresses $EFFE and $EFFF must contain the starting address of your program. (see the secondary vector table addresses info on page 38 of AN2153)

3. Edit and Assemble test_flash.asm until it's assembled without error messages.

4. Double click on make_s2_test_flash.bat and it will create test_flash.s29 and a duplicated file named test_flash.s2, which will only be recognized by some IDEs as an s2 file.

5. Change the jumper on J9 from right side to left side and the set the slide switch S7 at the left side for setting up bootloader mode.

6. Press the reset button and it will enter bootloader mode. The 7-segment display will show b-4-3 sequentially and the PC screen will show the bootloader menu:

   ```
   HCS912DP256 bootloader menu:

   a) Erase Flash
   b) Program Flash
   c) Set Baud Rate
   d) Erase EEPROM
   ?
   ```

   Option a) will erase the D-Bug12 portion of the flash memory, but not the bootloader itself.
   Option b) will program the D-Bug12 portion of the flash memory, but not the bootloader itself.
   Option c) will set a new baud rate.
   Option d) will erase all on-chip EEPROM.

   To program flash memory with test_flash.s29:

   1. Enter the option a) to erase the D-Bug12 portion of flash memory. Wait until the bootloader menu comes back after the flash memory is erased.
      Note: Always erase the flash memory before programming it.
   2. Enter the option b), the bootloader will wait for your file. **Do not type anything.**
   3. Click the Build button in AsmIDE, then click the Download option and select the file, test_flash.s29. If you don't see test_flash.s29, you have to select 'all files' for type of file. Once the download starts you should see the following marching stars on the screen:
      ```
      ****************
      ```
   4. The bootloader menu re-appears after the flash memory is programmed.

7. Change the jumper on J9 from left side to right side and the set the slide switch S7 at the right side for setting up EVB mode.

8. Press the reset button. The 7-segment display will show E-4-3 sequentially and test_flash.asm will run from flash memory just like test.asm runs from RAM.

You can program flash memory with the ex2_flash.asm. For your convenience, I generated the ex2_flash.s29 for you to download. The ex2_flash is a much simpler sample program. By comparing our original ex2.asm and the ex2_flash.asm you can easily see what have to be done for modifying your source program.

WARNING: If your program code size is smaller than 3K, use the EEPROM to store your code instead of the flash memory because the EEPROM write is rated at 10,000 and it's much higher than on the flash memory.