Running a SoftConsole "Hello, World!" Project on the SmartFusion2 Starter Kit

This application note explains how to use the Libero SoC and SoftConsole software to develop a simple application on the SmartFusion2 Starter Kit. The application is a "Hello, world" demo that uses GPIO to blink the on-board LEDs and LPDDR memory for data storage.

Hardware Platform

The hardware platform is Emcraft Systems' SmartFusion2 system-on-module (SOM) plugged into the SOM-BSB-EXT baseboard (SmartFusion2 Starter Kit).

The demo described in this application note assumes that the mini-USB cable is plugged into the P1 mini-USB connector on the SOM-BSB-EXT board (to provide power and serial console) and the FlashPro4 (or FlashPro3) JTAG programmer is connected to the development host and to the P5 connector on the SOM-BSB-EXT board.

The serial connection may require the **FT232R USB drivers** to be installed to the development host.

Also, please open the serial console window on the USB-Serial console port using any terminal application (HyperTerm, PuTTY, etc.) with the 115200,8N1 parameters.

Installing the Software on the Development Host

For this application note we use SoftConsole and Libero software running on a Windows development host.

1. Download and install the following **components**:
    o Libero SoC v11.0 Beta SPA
    o SoftConsole 3.4 beta
    o Firmware catalog
2. Download the .zip version of the M2S-SOM Libero project corresponding to your **M2S-SOM revision** (1A or 2A) and open the M2S-SOM project in Libero, update IP cores if asked.
3. Download the respective prebuilt **M2S-SOM firmware design** .stp file.
4. Program the .stp design into M2S-SOM using the FlashPro v11.0_Beta (FlashPro is a part of the Libero installation):
    a. Start FlashPro on a Windows host.
    b. From the FlashPro IDE, create a new project with an arbitrary name.
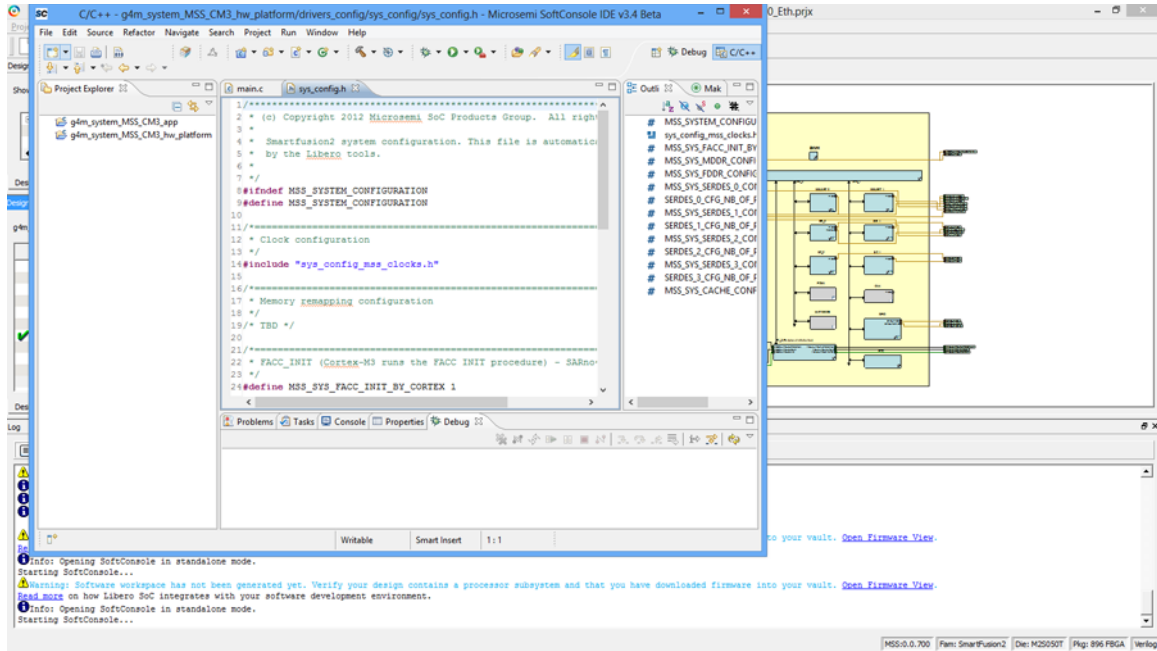   c. From the main FlashPro window, push **Configure Device**.

  d. Push **Browse next** to load existing programming file.

  e. Browse to the Linux SmartFusion2 project .stp file and choose it.

5. In the **Project -> Tools profile** editor create a new profile for the Software IDE:

  o Name: an arbitrary name (**SoftConsole-M2S**)

  o Tool Integration: **SoftConsole**

  o Location: Location of the eclipse.exe executable of the SoftConsole 3.4 beta installation.
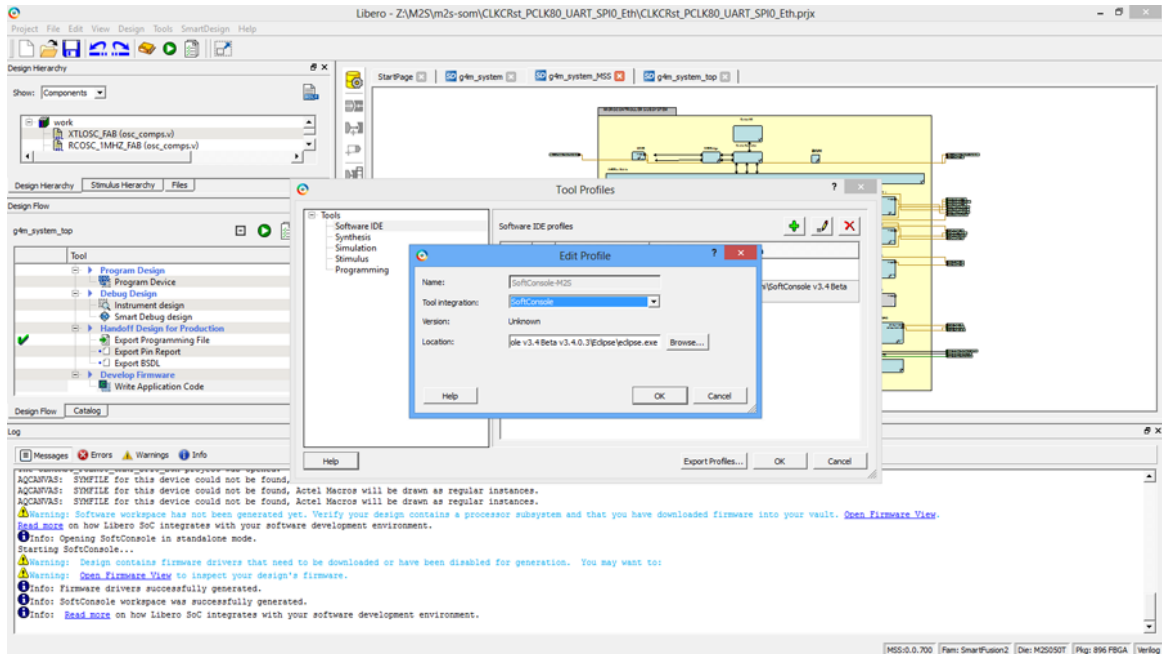
6. Save the profile, make sure that it is set as **Active**.

Running and Debugging the Project

1. In the Libero **Design Flow** view, launch the **Write Application Code** action. SoftConsole will start with the sample project workspace with two projects opened.

Microsemi Corporation
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100 · Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996 · **www.microsemi.com** 2

2. Start debugging the prebuilt application. Open the **Debug as** pulldown menu (to the right of the **bug** button) and select g4m_system_MSS_CM3_app Debug. Confirm the**Perspective Launch** dialog.
3. The program will be loaded to the target SRAM and stopped on the first breakpoint atmain(). Step over (F6) and into the functions (F5), examine variables (just point the mouse over the variable), and so on. Single-step until up to the sprintf() just after theconfig_mddr_lpddr(). Add a memory monitor to the RAM address0xa0000000 using the **Memory** tab (lower SoftConsole window). Now step over sprintf() and observe that the string appears at location 0xa0000000.
4. Resume the program run with F8 ("green triangle" in the **Debug** window). Observe theHello, SmartFusion2 on the console and two LEDs blinking.

Modifying the Libero and SoftConsole Projects

To modify the Libero project, rebuild and rerun the application in SoftConsole:

1. Terminate the program with Ctrl-F2 ("red square") and switch back to the **C/C++ Editor** perspective. Don't leave SoftConsole.
2. In Libero, open the **Design -> Configure Firmware** tab. Let's add a new software driver to our SoftConsole project. Add a checkbox to theSmartFusion2_MSS_NVM_Driver_0. Re-generate SoftConsole workspace (yellow barrel with gear in the middle of the Libero window).
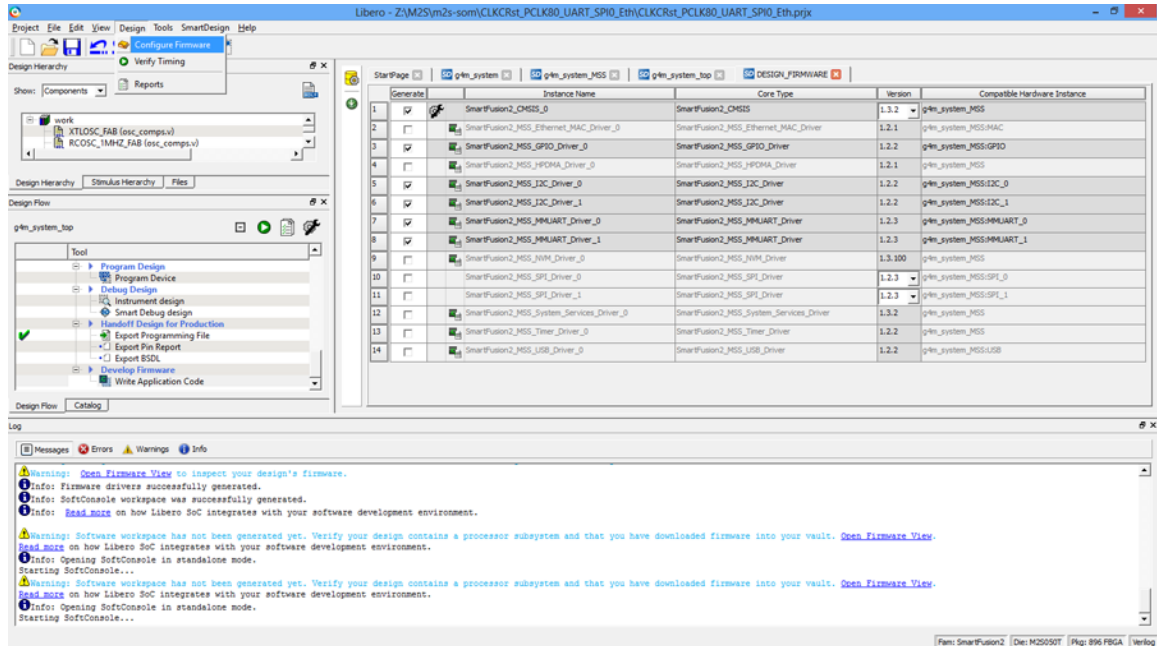
3. Now switch to SoftConsole and observe that the mss_nvm firmware appears in the tree of the g4m_system_MSS_CM3_HW_platform project (in the drivers/directory).
4. Before rebuilding the project, open the sys_config.h (present in the editor tab) and change

#define MSS_SYS_MDDR_CONFIG_BY_CORTEX   1

to

#define MSS_SYS_MDDR_CONFIG_BY_CORTEX   0

This is needed because Libero re-generated the SoftConsole project workspace and unconditionally enabled initialization of the LPDDR memory by the DDR module firmware added to the SoftConsole project. This initialization is incompatible with the memory used on the M2S-SOM (the demo application uses the correct LPDDR initialization code inconfig_mddr_lpddr()).

5. Rebuild the project: **Project -> Clean**, select **Clean all projects** and **Start the build immediately**.
6. Re-run the debug session of the project.