Linux SmartFusion2
Emcraft Systems SmartFusion2 SOM Starter Kit Guide

Release 1.9.1

Table of Contents

1. Overview

This document is the SmartFusion2 SOM Starter Kit Guide, Release 1.9.1.

The SmartFusion2 Starter Kit (SF2-STARTER-KIT-ES) provides designers with a prototyping platform for SmartFusion2 system-on-chip (SoC) FPGAs, which integrate inherently reliable Flash-based FPGA fabric, a 166 MHz ARM Cortex-M3 processor, advanced security processing accelerators, DSP blocks, SRAM, eNVM and industry-required high-performance communication interfaces all on a single chip.

The SmartFusion2 device resides on Emcraft Systems' miniature (34 mm x 59 mm) mezzanine form factor System-On-Module (M2S-SOM), which integrates the SmartFusion2 SoC (FG896 package), 64 MB LPDDR, 16 MB SPI Flash and an Ethernet PHY. The SOM-BSB-EXT baseboard provides easy-to-use benchtop access to the SmartFusion2 SoC and System-On-Module interfaces.

2. Product Contents

This product includes the following components.

2.1. Shippable Hardware Items

The following hardware items are shipped to customers of this product:

1. SmartFusion2 System-On-Module with M2S050-ES-FG896;

2. SOM-BSB-EXT SmartFusion2 System-On-Module baseboard;

3. USB 2.0 A Male to mini-B Y-cable for UART/power interface (up to 1A) to PC;

4. USB 2.0 A Male to mini-B cable for connection of SmartFusion2 to PC in USB device ("gadget") role, and also FlashPro4 connection;

5. Mini-B to USB 2.0 A Female cable for connection of USB devices to SmartFusion2;

6. Ethernet cable;

7. USB WiFi module;

8. FlashPro4 JTAG programmer for programming and debugging of SmartFusion2.

Note: *The SF2-STARTER-KIT-ES is RoHS compliant.*

Note: *Unless purchased as a product option bundled with a Microsemi's FlashPro4 device, this product does not include a FlashPro4 programmer tool or associated hardware items. That equipment needs to be purchased directly from Microsemi.*

2.2. Downloadable Hardware Materials

The following hardware materials are available for download from Emcraft's web site to customers of this product:

1. `SOM-BSB-EXT-1A-schem.pdf` - SOM-BSB-EXT schematics in PDF format;

2. `SOM-BSB-EXT-1A-bom.xls` - SOM-BSB-EXT Bill-Of-Materials (BOM) in Excel format;

3. `SOM-BSB-EXT-1A_Dimensions.pdf` – SOM-BSB-EXT mechanical drawing in PDF format;

4. `M2S-SOM.IntLib` - Altium Designer 9.4 integrated library for the M2S-SOM symbol and footprint.

2.3. Downloadable Software Materials

The following software materials are available for download from Emcraft's web site to customers of this product:

1. `m2s-som-<rev>.stp` – M2S-SOM Libero `.stp` file with the U-Boot image embedded, ready for installation onto the SmartFusion2 SOM using the Microsemi FlashPro tool;

    Note: It is critical that the `.stp` file match the kit version (revision of the SF2 SOM); otherwise, damage to the hardware may result.

2. `m2s-som-<rev>.zip` - M2S-SOM Libero and SoftConsole "Hello, world" demo project ready for the SmartFusion2 SOM;

    Note: It is critical that the `.zip` file of the demo SoftConsole project match the kit version; otherwise, damage to the hardware may result. Customers, who have developed custom Libero projects using SF2 SOM Rev 1A modules, must explicitly port their designs to SF2 SOM Rev 2A modules, using the provided demo project as a starting point.

    Note: The Libero project included with Release 1.9.1 of the SF2-STARTER-KIT-ES2 was developed using Libero v11 Beta SPA (version 10.9.2.10). That version of Libero doesn't support selection of `M2S050T_ES` as the SmartFusion2 device; only `M2S050T` is supported.

    The Libero project included with the upcoming Release 1.10.0 of the SF2-STARTER-KIT-ES2 will be developed using Libero v11 SP1. Select `M2S050T_ES` as the SmartFusion2 device when using that Libero version.

3. `networking.uImage` - prebuilt Linux image ready to be loaded to the SmartFusion2 SOM board;

4. `linux-M2S-1.9.1.tar.bz2` - Linux SmartFusion2 software development environment, including:

    o   U-Boot firmware;

    o   Linux kernel;

    o   `busybox` and other target components;

    o   Linux-hosted cross-development environment;

    o   Framework for developing multiple projects (embedded applications) from a single installation, including sample projects allowing to kick-start software development for Linux SmartFusion2.

2.4. Downloadable Documentation Materials

The following documentation materials are available for download from Emcraft's web site to customers of this product:

1. `m2s-som-ha.pdf` - Emcraft Systems SmartFusion2 SOM (System-On-Module) Hardware Architecture specification;

2. `m2s-som-ext-bsb-ha.pdf` - Emcraft Systems SOM-BSB-EXT Baseboard Hardware Architecture specification;

3. `linux-cortexm-um-1.9.1.pdf` - Linux Cortex-M User's Manual;

4. `M2S-SOM-skg-1.9.1.pdf` - Emcraft Systems SmartFusion2 SOM Starter Kit Guide (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux SmartFusion2, Release 1.9.1:

- U-Boot firmware:
    - U-Boot v2010.03;
    - Target initialization from power-on / reset;
    - Runs from the internal eNVM and internal SRAM (no external memory required for standalone operation);
    - Serial console;
    - Ethernet driver for loading images to the target;
    - Serial driver for loading images to the target;
    - Device driver for built-in Flash (eNVM) and self-upgrade capability;
    - Device driver for storing environment and Linux images in external Flash;
    - Autoboot feature, allowing boot of OS images from Flash or other storage with no operator intervention;
    - Persistent environment in Flash for customization of target operation;
    - Sophisticated command interface for maintenance and development of the target.
- Linux:
    - uClinux kernel v2.6.33;
    - Boot from compressed and uncompressed images;
    - Ability to run critical kernel code from integrated Flash of SmartFusion2;
    - Serial device driver and Linux console;
    - Ethernet device driver and networking (`ping`, NFS, Telnet, FTP, `ntpd`, etc.);
    - `busybox` v1.17;
    - POSIX `pthreads`;
    - Process-to-kernel and process-to-process protection using the Memory Protection Unit (MPU) of the SmartFusion2 core;
    - Hardened exception handling; an exception triggered by a process affects only the offending process;
    - Loadable kernel modules;
    - Secure shell (`ssh`) daemon;
    - Web server;
    - MTD-based Flash partitioning and persistent JFFS2 Flash file system for external Flash;
    - Device driver for the USB OTG interface;
    - I^2C device driver;
    - SPI controller master-mode device driver;
    - Device driver for the embedded NVM;
- Development tools:
    - ARMv7-optimized GNU toolchain from CodeSourcery (`2010q1`) is used for development of U-Boot, Linux and user-space applications (toolchain must be downloaded separately from the CodeSourcery web site);

- o Cross GDB for debugging user-space applications;
- o `mkimage` tool used by the Linux kernel build process to create a Linux image bootable by U-Boot.

- Development environment:

  - o Linux-hosted cross-development environment;
  - o Development of multiple projects (embedded applications) from a single installation;
  - o `hello` sample project ("Hello, world!" single-process configuration);
  - o `networking` sample project (basic shell, networking and Flash management tools demonstration);
  - o `developer` sample project (template project that can be used to jump-start development of custom user-space applications and loadable kernel modules).

3.2. New and Changed Features

This section lists new and changed features of this release:

1. Resolve the following limitation: UART is limited to 57.6 Kps on SmartFusion2 (should be 115.2 Kps).
   ID: RT 82000.

2. Resolve the following limitation: cache is disabled in Linux on SmartFusion2.
   ID: RT 82001.

3. Develop Linux USB device driver for SmartFusion2.
   ID: RT 82002.

4. Resolve the following limitation: SF2 uClinux kernel incorrectly assumes that Sysreg clock is 160MHz (should be 166MHz).
   ID: RT 82189.

5. SmartFusion2 Linux SPI driver must be interrupt-driven.
   ID: RT 82211.

6. Resolve the following limitation: USB WiFi unreliable on SmartFusion2 under uClinux.
   ID: RT82414.

7. Improve DDR settings on SmartFusion2.
   ID: RT 82416.

8. Update the Libero project not to swap the MSS SPI0 signals in FPGA.
   ID: RT 82539.

3.3. Known Problems & Limitations

This section lists known problems and limitations of this release:

1. `CONFIG_KERNEL_IN_ENVM` requires disabling `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
   ID: RT 74683.
   Workaround: When enabling `CONFIG_KERNEL_IN_ENVM` in the kernel, disable `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.

2. USB hot plug doesn't work on SmarFusion2 under uClinux.
   ID: RT 82415.
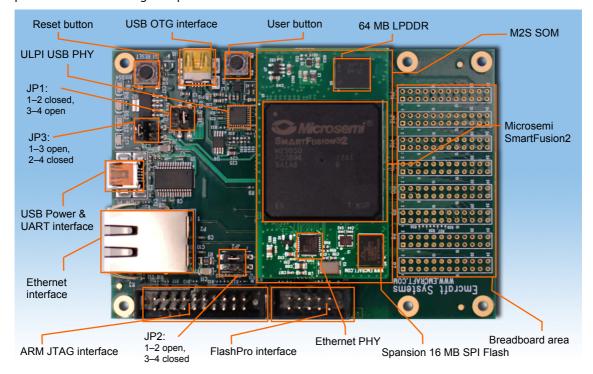   *Workaround*: Limitation will be lifted in next release.

4. Hardware Setup

This section explains how to set up the Emcraft Systems SmartFusion2 SOM board in harness with the Emcraft Systems SOM-BSB-EXT baseboard.

4.1. Hardware Interfaces

The SmartFusion2 SOM board in harness with the Emcraft Systems SOM-BSB-EXT baseboard provides the following components and interfaces:

4.2. Jumpers

Prior to powering up the SF2-STARTER-KIT-ES for the first time, make sure the jumpers are in the following positions on the SOM-BSB-EXT board:

Jumper	Location	Purpose	Settings
JP1	Below the USB OTG connector	Enable power on the SmartFusion2 SOM (VCC3)	1–2 closed, 3–4 open
JP2	To the right of the RJ-45 Ethernet jack	Select appropriate JTAG mode and enable power to the SmartFusion2 JTAG controller	1–2 open, 3–4 closed
JP3	Above the USB Power and UART I/F connector	Use the mini-USB port as the power source	1–3 open, 2–4 closed

4.3. Board Connections

To power the SOM-BSB-EXT baseboard with the SmartFusion2 SOM up, simply connect it to a PC / notebook by plugging a mini-USB cable into the P1 mini-USB connector on the SOM-BSB-EXT board. As soon as the connection to the PC has been made, the LED DS2 should lit up, indicating that the board is up and running.

A single USB connection provides a 500 mA power to the SmartFusion2 SOM, which is sufficient for basic functionality. Note however that some advanced operations, such as WiFi connectivity using the USB WiFi module, may require more than 500 mA for reliable operation. Use the second link of the mini-USB Y-cable to connect to the PC for such configurations.

On the PC side, the USB link provides a serial console device to the SmartFusion2 SOM. The software installed on the board is configured for a 115.2 K terminal. On the Linux host, the serial console is available using a /dev/ttyUSBn device.

Note: *Early version of the SmartFusion2 SOM (Rev 1A) is configured for a 57.6 K baud terminal.*

To provide network connectivity to the board, connect it into your LAN by plugging a standard Ethernet cable into the J1 connector. The module is pre-configured with an IP address of 192.168.0.2.

The SmartFusion2 SOM comes with the U-Boot firmware and an appropriate Libero project pre-installed into SmartFusion. U-Boot provides sufficient interfaces for uploading and installing new firmware images onto the board so you may never need to re-install firmware over the JTAG interface. If however at some point you require re-programming U-Boot onto your board, connect it to a Microsemi FlashPro programmer tool by plugging a standard JTAG cable into the P3 connector.

4.4. Extension Interfaces

For description of the extension interfaces provided by the Emcraft Systems SmartFusion2 SOM board refer to
[Emcraft Systems SmartFusion2 SOM (System-On-Module) Hardware Architecture](#).

For description of the extension interfaces provided by the Emcraft Systems SOM-BSB-EXT baseboard refer to [Emcraft Systems SOM-BSB-EXT Hardware Architecture](#).

The above mentioned documents can be downloaded from the following page:

> http://www.emcraft.com/som/m2s

5. SmartFusion2 SOM Board Linux Software Set-up

5.1. Linux Demo and Development Environment

The SF2-STARTER-KIT-ES comes with a preloaded Linux (uClinux) demo configuration. To initiate the Linux bootstrap, simply power-up the SF2-STARTER-KIT-ES or reset it by pushing the Reset Push Button (S1 on the SOM-BSB-EXT baseboard).

The Linux development environment for the SF2-STARTER-KIT-ES, including the configuration installed into the SF2-STARTER-KIT-ES, can be downloaded from this page, Section "Software Materials": http://www.emcraft.com/som/m2s

5.2. U-Boot Environment

When the SmartFusion2 SOM board is reset, the Linux bootstrap will proceed to boot the U-Boot firmware from the on-chip eNVM. Unless interrupted, U-Boot proceeds to load the Linux bootable image from the SPI Flash into the LPDDR and passes control to the Linux kernel entry point in LPDDR. Linux bootstraps the kernel, mounts a root filesystem in RAM and enters the interactive shell command interface printing the following output to the serial console:

```
U-Boot 2010.03-linux-cortexm-1.9.1 (Jan 23 2013 - 19:43:46)

CPU  : SmartFusion2 SoC (Cortex-M3 Hard IP)
Freqs: CORTEX-M3=166MHz,PCLK0=83MHz,PCLK1=83MHz
Board: M2S-SOM Rev A, www.emcraft.com
DRAM:  64 MB
In:    serial
Out:   serial
Err:   serial
Net:   M2S MAC
Hit any key to stop autoboot:  0
M2S-SOM>
```

U-boot makes use of the so-called environment variables to define various aspects of the system functionality. Parameters defined by the U-boot environment variables include: target IP address, target MAC address, address in RAM where a Linux bootable images will be loaded, and many more. To examine the current settings of the environment variables, run `printenv` from the U-Boot command interface.

U-Boot provides a command called `saveenv` that stores the up-to-date run-time environment to the persistent storage, which will be the external Flash for the U-Boot configuration used on the SmartFusion2 SOM board. You need to call `saveenv` any time when you want to copy current settings of the environment variables to the persistent storage in Flash. This is how you can write the current U-Boot environment to the external Flash:

```
M2S-SOM> saveenv
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
M2S-SOM>
```

5.3. Ethernet MAC Address

In Linux SmartFusion2, the MAC address of the Ethernet interface is defined by the `ethaddr` U-Boot environment variable. The value of the MAC address can be examined from the U-Boot command line monitor as follows:

```
M2S-SOM> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
M2S-SOM>
```

The SmartFusion2 SOM board comes with `ethaddr` set to a MAC address uniquely allocated for the specific board. Given that each SmartFusion2 SOM board has a unique MAC address allocated to it, there is no need to update the `ethaddr` variable (although it is possible to do so).

The MAC address can be changed by modifying the `ethaddr` variable as follows:

```
M2S-SOM> setenv ethaddr C0:B1:3C:88:88:89
```

Don't forget to store your update in the persistent storage using `saveenv` so it is remembered across resets and power cycles.

5.4. Network Configuration

You will have to update the network configuration of your board to match settings of your local environment.

Typically, all you have to allow loading images over network from a TFTP server is update the U-Boot environment variables `ipaddr` (the board IP address) and `serverip` (the IP address of the TFTP server). Here is how it is done.

Update `ipaddr` and `serverip`:

```
M2S-SOM> setenv ipaddr 192.168.0.2
M2S-SOM> setenv serverip 192.168.0.1
```

and then save the updated environment to the external Flash so that your changes are persistent across resets/power cycles.

5.5. Running Pre-installed Linux Image

The SmartFusion2 SOM board comes with a Linux bootable image for the `networking` project installed into external Flash. To boot this Linux configuration onto the SmartFusion2 SOM board just reset the board and let U-Boot perform the autoboot sequence.

Detailed information on functionality of the pre-installed Linux image can be found in *Linux Cortex-M User's Manual*, Section 3.

5.6. Loading Linux Images

At this point, you are able to load Linux bootable images to the board over TFTP and either boot them directly or install them to the external Flash to allow booting Linux from Flash on power-up/reset.

On the host, activate the Linux SmartFusion2 development environment and build the `networking` project:

```
-bash-3.2$ . ACTIVATE.sh
-bash-3.2$ cd projects/networking/
-bash-3.2$ make
...
-bash-3.2$
```

Copy the Linux bootable image to the TFTP download directory:

```
-bash-3.2$ cp networking.uImage /tftpboot/vlad/
-bash-3.2$
```

To load the image directly, use the `netboot` U-Boot macro:

```
M2S-SOM> setenv image vlad/networking.uImage
M2S-SOM> run netboot
Auto-negotiation...completed.
Core10/100: link UP (100/Full)
Using Core10/100 device
TFTP from server 172.17.0.1; our IP address is 172.17.5.100
Filename 'vlad/networking.uImage'.
...
Loading: #################################################################
         #################################################################
         #############
done
Bytes transferred = 2084704 (1fcf60 hex)
...
   Image Name:   Linux-2.6.33-arm1
   Image Type:   ARM Linux Kernel Image (uncompressed)
...
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...

Linux version 2.6.33-arm1 (vlad@ocean.emcraft.com) (gcc version 4.4.1 (Sourcery G++ Lite
2010q1-189) ) #1 Wed Jan 23 15:43:44 MSK 2013
...
```

To load the image into the Flash, use the U-Boot `update` macro:

```
M2S-SOM> setenv image vlad/networking.uImage
M2S-SOM> run update
Auto-negotiation...completed.
Core10/100: link UP (100/Full)
Using Core10/100 device
TFTP from server 172.17.0.1; our IP address is 172.17.5.100
Filename 'vlad/networking.uImage'.
...
Loading: #################################################################
         #################################################################
         #############
done
Bytes transferred = 2084704 (1fcf60 hex)
16384 KiB S25FL128S 64K at 0:0 is now current device
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
M2S-SOM>
```

Reset the board and verify that the newly programmed image boots on the target in the autoboot mode:

```
M2S-SOM> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.9.1 (Jan 23 2013 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (Jan 23 2013 - 17:21:30)
~ #
```

5.7. U-Boot Build

The BSP distribution comes with U-Boot pre-built for the SmartFusion2 SOM board. If however you need to re-build U-Boot for your board, please follow the instructions below:

1. Install the Linux SmartFusion2 distribution to the development host, as described in the *Linux Cortex-M User's Manual*.

2. From the top of the Linux SmartFusion2 installation, activate the Linux SmartFusion2 cross-compile environment by running `. ACTIVATE.sh`.

3. Go to the U-Boot source directory (`cd u-boot/`).

4. Run the following commands:

```
[psl@pvr u-boot]$ make m2s-som_config
Configuring for m2s-som board...
[psl@pvr u-boot]$ make -s
[psl@pvr u-boot]$ make -s u-boot.hex
```

5.8. U-Boot Installation

The Emcraft Systems SmartFusion2 SOM board arrives with the U-Boot firmware pre-installed into the on-chip Flash of the SmartFusion2. The U-Boot command line interface provides commands that allow upgrading U-Boot on the running target in self-upgrade mode.

However, should you program a faulty U-Boot image into SmartFusion2, U-Boot can be re-installed using the Emcraft-provided Linux SmartFusion2 Libero project and a Microsemi FlashPro tool.

Note: *It is critical that the `.stp` file match the kit version (revision of the SF2 SOM); otherwise, damage to the hardware may result.*

Here is an example of how this can be done:

1. Start FlashPro on a Windows host;

2. From the FlashPro IDE, create a new project with an arbitrary name;

3. From the main FlashPro window, push `Configure Device`;

4. Push `Browse next` to load existing programming file. Browse to the Linux SmartFusion2 project `.stp` file corresponding to your M2S-SOM hardware revision and choose it.

5. Push `Program` at the top of the main window to program the project onto the SmartFusion2 device and wait for the programming procedure to complete. If the programming completes successfully, a next reset should bring the U-Boot start-up messages and the command line interface onto the serial console interface.

6.   Running the SoftConsole Demo Project

The SF2-STARTER-KIT-ES is fully compatible with the Microsemi SoftConsole IDE. A simple SoftConsole "Hello, world" demo project, immediately ready for running on the SF2-STARTER-KIT-ES, can be downloaded from this web page:

http://www.emcraft.com/som/m2s

As configured, the demo project runs from the eSRAM of the SmartFusion. Connect the FlashPro JTAG programmer to the FlashPro interface connector (P5) on the SOM-BSB-EXT and then build and run the demo project using the SoftConsole IDE.

When run on the SF2-STARTER-KIT-ES, the demo prints messages to the serial console and blinks the user LEDs on the SOM-BSB-EXT.

7. Software and Licensing

The SmartFusion2 Starter Kit is supported by Libero® System-on-Chip (SoC) v11.0 Beta and SoftConsole v3.4 Beta.

The releases can be downloaded from

http://www.actel.com/download/software/liberosocv11/default.aspx

Libero SoC requires a valid license. You can request a free license from

www.microsemi.com/soc/Portal/default.aspx?r=1

8. Further Materials

Emcraft Systems provides detailed kit information and materials from this page:

http://www.emcraft.com/som/m2s

As new SmartFusion2 and kit related materials become available, they will be posted on the Microsemi and Emcraft Systems web sites. Microsemi recommends that you sign up for Product Updates to be notified when new material is available.

Microsemi product updates: https://www.actel.com/portal/default.aspx?r=2

Emcraft Systems product updates: http://www.emcraft.com/som/m2s/register

Refer to *Emcraft Systems SmartFusion2 SOM (System-On-Module) Hardware Architecture* for detailed information on the hardware architecture of the Emcraft Systems SmartFusion2 SOM board.

Refer to *Emcraft Systems SOM-BSB Hardware Architecture* for detailed information on the hardware architecture of the Emcraft Systems SOM-BSB-EXT baseboard.

Refer to *Linux Cortex-M User's Manual* for detailed information on the software architecture of the Linux SmartFusion2 distribution.

9. Support

Technical support is available online at www.microsemi.com/soc/support and by email at

soc_tech@microsemi.com.

Microsemi SoC Sales offices, including Representatives and Distributors, are located worldwide. To find your local representative visit www.microsemi.com/soc/company/contact.

Emcraft Systems provides support for any matters related to the Linux Board Support Package for the SF2-STARTER-KIT-ES: a2f-linux-support@emcraft.com.