Linux SmartFusion2

Emcraft Systems M2S-FG484 SOM Starter Kit Guide

Release 1.12.0

Table of Contents

1. OVERVIEW 3

2. PRODUCT CONTENTS 3

2.1. Shippable Hardware Items 3
2.2. Downloadable Hardware Materials 3
2.3. Downloadable Software Materials 3
2.4. Downloadable Documentation Materials 4

3. SOFTWARE FUNCTIONALITY 4

3.1. Supported Features 4
3.2. New and Changed Features 5
3.3. Known Problems & Limitations 6

4. HARDWARE SETUP 6

4.1. Hardware Interfaces 6
4.2. Jumpers 7
4.3. Board Connections 7
4.4. Extension Interfaces 7

5. M2S-FG484 SOM BOARD LINUX SOFTWARE SET-UP 8

5.1. U-Boot Environment 8
5.2. Ethernet MAC Address 8
5.3. Network Configuration 9
5.4. Running Pre-installed Linux Image 9
5.5. Loading Linux Images 9
5.6. U-Boot Build 10
5.7. U-Boot Installation 10

6. FURTHER MATERIALS 11

7. SUPPORT 11

1. Overview

This document is the Emcraft Systems M2S-FG484 SOM Starter Kit Guide, Release 1.12.0.

The BSP provides a software development environment for evaluation and development of Linux on the Cortex-M3 processor core of the Microsemi SmartFusion2 microcontroller using the Emcraft Systems M2S-FG484 SOM board in harness with the Emcraft Systems SOM-BSB-EXT baseboard as a hardware platform.

This BSP is provided as part of the Emcraft Systems SmartFusion2 System-On-Module (M2S-FG484 SOM) Starter Kit. The kit provides a hardware platform and Linux software development environment for the SmartFusion2 System-On-Module (M2S-FG484 SOM).

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. M2S-FG484 SOM board;
2. SOM-BSB-EXT baseboard;
3. Mini-USB cable UART/power interface.

Note that this product does not include any JTAG programmer tools or associated hardware items. Such equipment needs to be purchased directly from respective vendors.

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-FG484-SOM.IntLib` - Altium Designer 9.4 integrated library for the M2S-FG484-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. `m2s010-fg484-som-1a.stp` - M2S-FG484-SOM Libero `.stp` file with the U-Boot image embedded, ready for installation onto the M2S010-FG484 SOM using the Microsemi FlashPro tool;
2. `m2s010-fg484-som-1a.zip` - M2S-FG484-SOM Libero and SoftConsole "Hello, world" demo project ready for the M2S010-FG484 SOM;
3. `m2s050-fg484-som-1a.stp` - M2S-FG484-SOM Libero `.stp` file with the U-Boot image embedded, ready for installation onto the M2S050-FG484 SOM using the Microsemi FlashPro tool;
4. `m2s050-fg484-som-1a.zip` - M2S-FG484-SOM Libero and SoftConsole "Hello, world" demo project ready for the M2S050-FG484 SOM;

   Note: *The Libero project included with the Release 1.12.0 is developed using Libero v11.1. For the M2S050-FG484 device, select M2S050T as the SmartFusion2 device when using that Libero version.*

Note: *Refer to the Microsemi web site for the Libero v11.1 minimum system requirements. To increase the virtual address space that is available to the applications on Windows 32-bit OS refer to Known Issue SL5583.*

5. `networking.uImage` - prebuilt Linux image ready to be loaded to the M2S-FG484 SOM board;
6. `linux-M2S-1.12.0.tar.bz2` - Linux SmartFusion2 software development environment, including:
   - U-Boot firmware;
   - Linux kernel;
   - `busybox` and other target components;
   - Linux-hosted cross-development environment;
   - 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-fg484-som-ha.pdf` - Emcraft Systems SmartFusion2 System-On-Module (M2S-FG484 SOM) Hardware Architecture specification;
2. `m2s-fg484-som-bsb-ext-ha.pdf` - Emcraft Systems SOM-BSB-EXT Baseboard Hardware Architecture specification;
3. `linux-cortexm-um-1.12.0.pdf` - Linux Cortex-M User's Manual;
4. `M2S-FG484-SOM-skg-1.12.0.pdf` - Emcraft Systems M2S-FG484 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 1.12.0:

- 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 (M2S010-FG484 SOM only);
  - I2C device driver;
  - SPI controller master-mode device driver;
  - Device driver for the embedded NVM;
  - GPIO device driver.
- 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);
  - Cross GDB for debugging user-space applications;
  - `mkimage` tool used by the Linux kernel build process to create a Linux image bootable by U-Boot.
- Development environment:
  - Linux-hosted cross-development environment;
  - Development of multiple projects (embedded applications) from a single installation;
  - `hello` sample project ("Hello, world!" single-process configuration);
  - `networking` sample project (basic shell, networking and Flash management tools demonstration);
  - `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. Port uClinux to M2S010-FG484.
   ID: RT 88436.
2. Support SPI0 and SPI1 and multiple chip selects in Linux SPI driver for SmartFusion2.
   ID: RT 91425.

3.3. Known Problems & Limitations

This section lists known problems and limitations of this release:

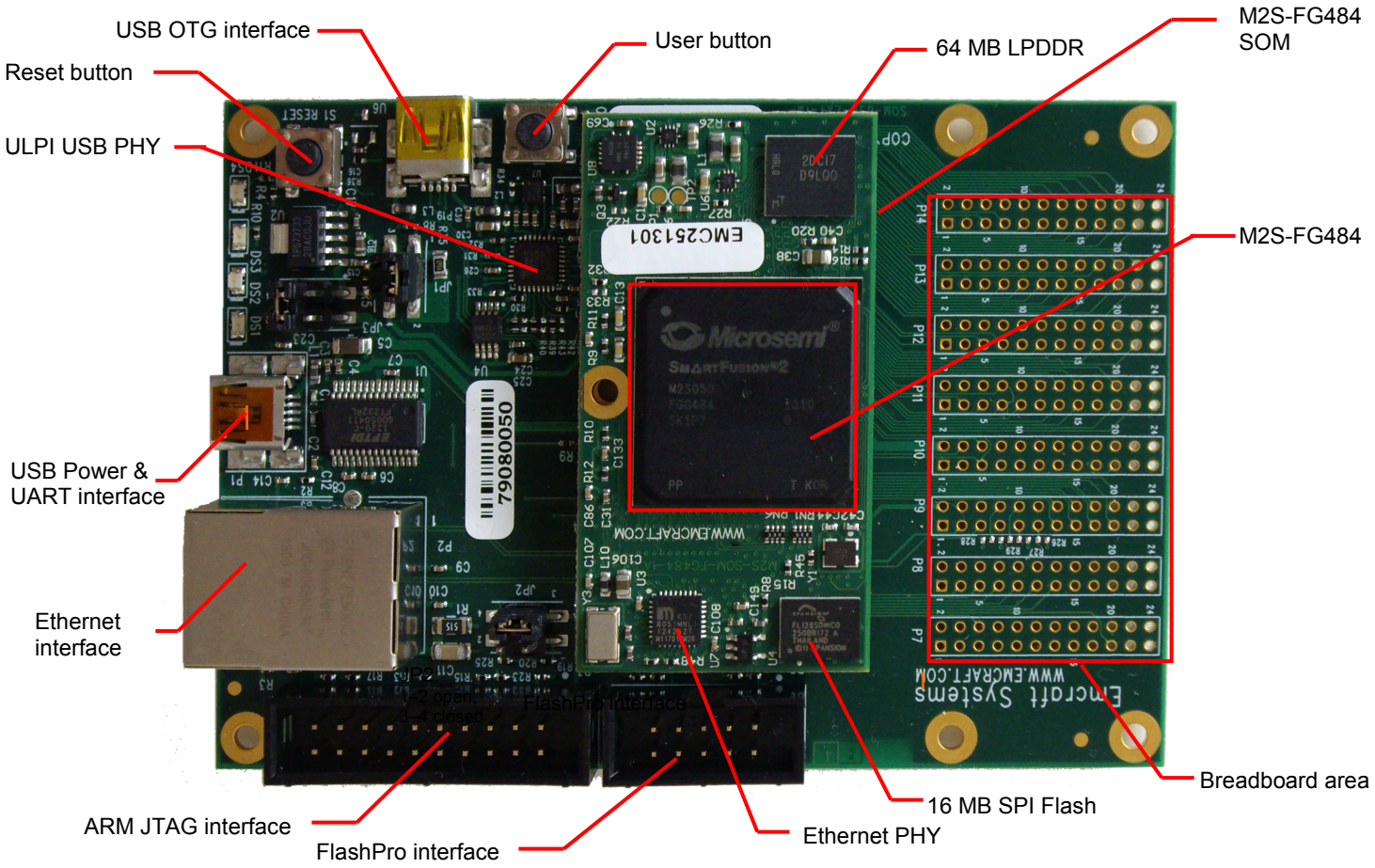
1. USB is not supported by this release on the M2S050-FG484 SOM since the USB ULPI interface is not supported on the M2S050-FG484 ES and PP (pre-production) devices.
2. `CONFIG_KERNEL_IN_ENVM` requires disabling `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
   ID: RT 74683.
   Workaround: When enabling `CONFIG_KERNEL_IN_ENVM` in the kernel, disable `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
3. Debugging of multi-threaded applications using GDB is not supported.
   ID: RT 77243.
   Workaround: None. This issue will be resolved in future releases of the product.
4. USB hot plug doesn't work on SmarFusion2 under uClinux.
   ID: RT 82415.
   Workaround: Limitation will be fixed in next release.

4. Hardware Setup

This section explains how to set up the Emcraft Systems M2S-FG484 SOM board in harness with the Emcraft Systems SOM-BSB-EXT baseboard.

4.1. Hardware Interfaces

The M2S-FG484 SOM board in harness with the Emcraft Systems SOM-BSB-EXT baseboard provides the following components and interfaces:



4.2. Jumpers

The following jumpers must be configured on the SOM-BSB-EXT board:

Jumper	Configuration	Notes
JP1	1-2 closed, 3-4 open	Enable power on the M2S-FG484 SOM (VCC3)
JP2	1-2 open, 3-4 closed	Select appropriate JTAG mode and enable power to the SmartFusion2 JTAG controller
JP3	1-3 open, 2-4 closed	Use the mini-USB port as the power source

4.3. Board Connections

To power the SOM-BSB-EXT baseboard with the M2S-FG484 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 M2S-FG484 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 M2S-FG484 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/ttyUSB`*n* device.

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

The M2S-FG484 SOM comes with the U-Boot firmware and an appropriate Libero project pre-installed into SmartFusion2. 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.

Note: Only one of the FlashPro JTAG and the ARM JTAG interfaces must be used at any time. Using the two JTAG interfaces simultaneously may result in a mal-function of the M2S-FG484 SOM.

4.4. Extension Interfaces

For description of the extension interfaces provided by the Emcraft Systems M2S-FG484 SOM board refer to *Emcraft Systems SmartFusion2 System-On-Module (M2S-FG484 SOM) Hardware Architecture*.

For description of the extension interfaces provided by the Emcraft Systems SOM-BSB-EXT baseboard refer to *Emcraft Systems SOM-BSB-EXT Baseboard Hardware Architecture*.

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

http://www.emcraft.com/som/m2s-fg484

5. M2S-FG484 SOM Board Linux Software Set-up

5.1. U-Boot Environment

When the M2S-FG484 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-1.12.0 (Dec 06 2013 - 19:43:46)

CPU  : SmartFusion2 SoC (Cortex-M3 Hard IP)
Freqs: CORTEX-M3=166MHz,PCLK0=83MHz,PCLK1=83MHz
Board: M2S-FG484-SOM Rev 1A, www.emcraft.com
DRAM:  64 MB
In:    serial
Out:   serial
Err:   serial
Net:   M2S_MAC
Hit any key to stop autoboot:  0
M2S-FG484-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 M2S-FG484 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-FG484-SOM> saveenv
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
M2S-FG484-SOM>
```

5.2. 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-FG484-SOM> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
M2S-FG484-SOM>
```

The M2S-FG484 SOM board comes with `ethaddr` set to a MAC address uniquely allocated for the specific board. Given that each M2S-FG484 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-FG484-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.

5.3. 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-FG484-SOM> setenv ipaddr 192.168.0.2
M2S-FG484-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.

5.4. Running Pre-installed Linux Image

The M2S-FG484 SOM board comes with a Linux bootable image for the `networking` project installed into external Flash. To boot this Linux configuration onto the M2S-FG484 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.

5.5. 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-FG484-SOM> setenv image vlad/networking.uImage
M2S-FG484-SOM> run netboot
...
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 Fri Dec 06 15:43:44 MSK 2013
...
```

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

```
M2S-FG484-SOM> setenv image vlad/networking.uImage
M2S-FG484-SOM> run update
...
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-FG484-SOM>
```

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

```
M2S-FG484-SOM> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (Dec 06 2013 - 17:19:37)
~ #
```

5.6. U-Boot Build

The BSP distribution comes with U-Boot pre-built for the M2S-FG484 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-fg484-som_config
Configuring for m2s-fg484-som board...
[psl@pvr u-boot]$ make -s
[psl@pvr u-boot]$ make -s u-boot.hex
```

5.7. U-Boot Installation

The Emcraft Systems M2S-FG484 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.

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 for the M2S-FG484 SOM board 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. Further Materials

Refer to *Emcraft Systems SmartFusion2 System-On-Module (M2S-FG484 SOM) Hardware Architecture* for detailed information on the hardware architecture of the Emcraft Systems M2S-FG484 SOM board.

Refer to *Emcraft Systems SOM-BSB-EXT Baseboard 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.

Visit Emcraft Systems' web site at www.emcraft.com to obtain additional materials related to Linux SmartFusion2.

7. Support

We appreciate your review of our product and welcome any and all feedback. Comments can be sent directly by email to:

a2f-linux-support@emcraft.com

The following level of support is included with your purchase of this product:

- Email support for installation, configuration and basic use scenarios of the product during 3 months since the product purchase;
- Free upgrade to new releases of the downloadable materials included in the product during 3 months since the product purchase.

If you require support beyond of what is described above, we will be happy to provide it using resources of our contract development team. Please contact us for details.