Linux SmartFusion2

BSP (Board Support Package) Guide for the Microsemi SF2-DEV-KIT Board

Release 1.10.1

Emcraft Systems

Table of Contents

1. Overview

This document is a Linux SmartFusion2 BSP (Board Support Package) Guide for the Microsemi SF2-DEV-KIT board, Release 1.10.1.

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 Microsemi SF2-DEV-KIT board as a hardware platform.

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. None - THIS IS A SOFTWARE-ONLY PACKAGE. Please purchase the SF2-DEV-KIT board from Microsemi or its distributors.

2.2. Downloadable Hardware Materials

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

1. None - Please obtain hardware materials pertinent to the SF2-DEV-KIT board directly from Microsemi.

2.3. Downloadable Software Materials

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

1. `sf2-dev-kit.stp` - SF2-DEV-KIT Libero `.stp` file with the U-Boot image embedded, ready for installation onto the SF2-DEV-KIT using the Microsemi FlashPro tool;

2. `sf2-dev-kit.zip` - SF2-DEV-KIT Libero project;

3. `networking.uImage` - prebuilt Linux image ready to be loaded to the SF2-DEV-KIT board;

4. `linux-M2S-1.10.1.tar.bz2` - Linux SmartFusion2 software development environment, including:

    a) U-Boot firmware;

    b) Linux kernel;

    c) `busybox` and other target components;

    d) Linux-hosted cross-development environment;

    e) 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. `linux-cortexm-um-1.10.1.pdf` - Linux Cortex-M User's Manual;

2. `linux-SF2-DEV-KIT-bspg-1.10.1.pdf` - Linux SmartFusion2 BSP (Board Support Package) Guide for the Microsemi SF2-DEV-KIT Board (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux SmartFusion2, Release 1.10.1:

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

- Development tools:

  o 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 defect: Ethernet interface doesn't work under U-boot and uClinux.
   ID: RT 82124.

2. Resolve the following defect: `sf erase` U-Boot command doesn't support arbitrary range.
   ID: RT 85334.

3. Improve boot-up time of network-enabled configurations in Linux Cortex-M3/M4.
   ID: RT 85493.

3.3. Known Problems & Limitations

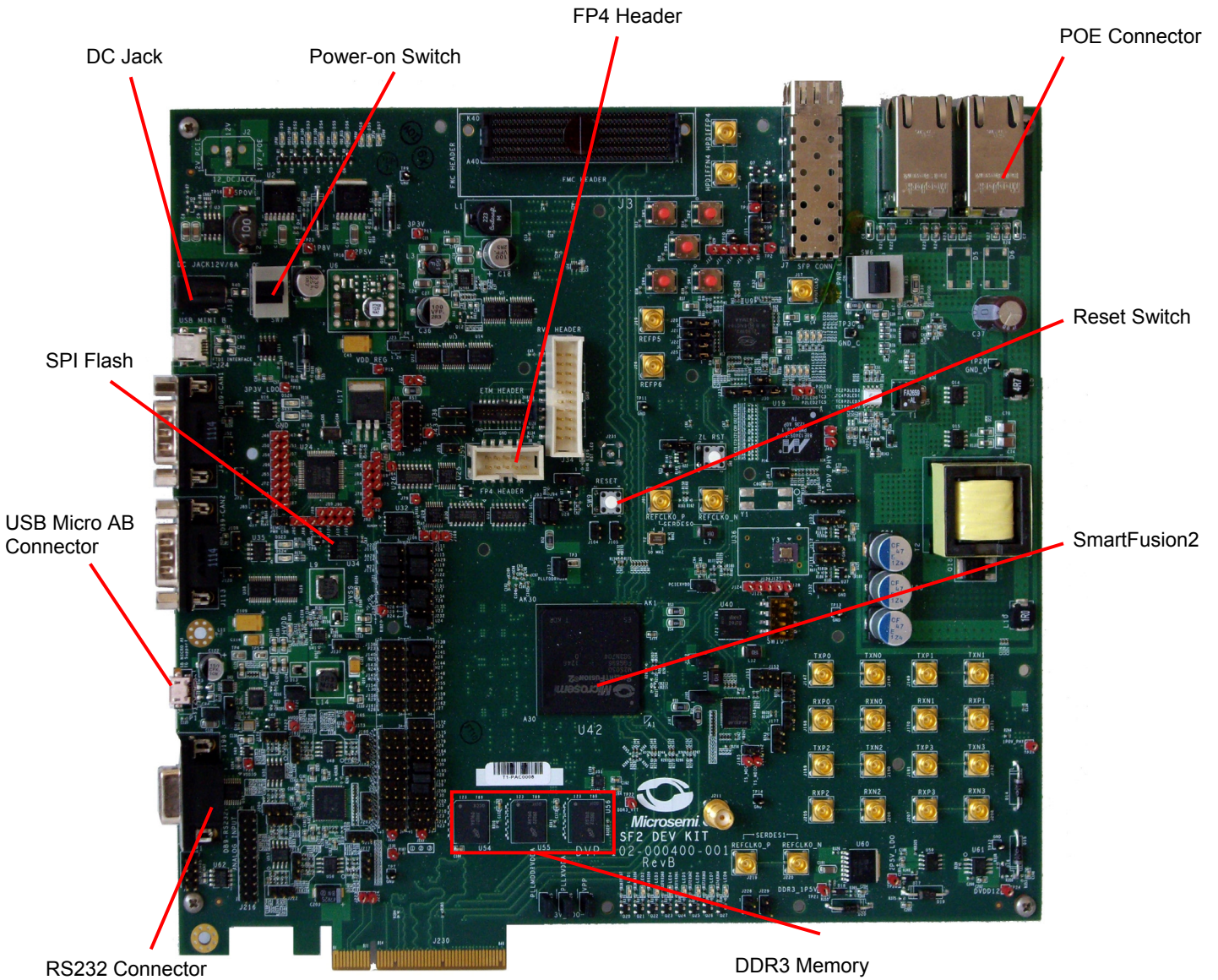This section lists known problems and limitations of this release:

1. Ethernet interface doesn't work in 100Mb networks.
   ID: RT 86022.
   Workaround: Connect the board to a 1000Mb network.

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. 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 Microsemi SF2-DEV-KIT board.

4.1. Hardware Interfaces

The SF2-DEV-KIT board provides the following components and interfaces:

DC Jack · Power-on Switch · FP4 Header · POE Connector · SPI Flash · Reset Switch · USB Micro AB Connector · SmartFusion2 · RS232 Connector · DDR3 Memory

4.2. Jumpers

The following jumpers must be configured on the SF2-DEV-KIT board. The jumpers not mentioned in the below table should be connected with the default settings described in the Microsemi's SF2-DEV-KIT User's Guide.

Jumper	Function	Settings
SPI_0		
J110	Jumper to select between SPI Flash SCK and FMC_V22.	
	Pin 1-2 SPI Flash	Closed
	Pin 2-3 FMC	Open
J118	Jumper to select between SPI Flash SDO and FMC_W27.	
	Pin 1-2 SPI Flash	Closed
	Pin 2-3 FMC	Open

Jumper	Function	Settings
J119	Jumper to select between SPI Flash SDI and FMC_Y30.	
	Pin 1-2 SPI Flash	Closed
	Pin 2-3 FMC	Open
J121	Jumper to select between SPI Flash SS and FMC_W28.	
	Pin 1-2 SPI Flash	Closed
	Pin 2-3 FMC	Open
SPI_1		
J20	Jumper to select between ZL30362 SPI_CS and FT4232 SPI_CS.	
	Pin 1-2 ZL30362	Closed
	Pin 2-3 FT4232	Open
J21	Jumper to select between ZL30362 SPI_SCL and FT4232 SPI_SCL.	
	Pin 1-2 ZL30362	Closed
	Pin 2-3 FT4232	Open
J22	Jumper to select between ZL30362 SPI_SO and FT4232 SPI_SI.	
	Pin 1-2 ZL30362	Closed
	Pin 2-3 FT4232	Open
J25	Jumper to select between ZL30362 SPI_SI and FT4232 SPI_SO.	
	Pin 1-2 ZL30362	Closed
	Pin 2-3 FT4232	Open
J54	Jumper to select between ZL30362 reset and FT4232 reset from VSS of SmartFusion2 device.	
	Pin 1-2 ZL30362	Closed
	Pin 2-3 FT4232	Open
USB		
J139	Jumper to select between USB reset and FMC_P24.	
	Pin 1-2 USB	Closed
	Pin 2-3 FMC	Open
J163	Jumper to select the USB mode of operation.	
	Pin 1-2 On-The-Go (OTG) mode	Closed
	Pin 2-3 Either host or device mode	Open
J164	Jumper to provide the VBUS supply to USB when using in Host mode.	1–2 Closed
MMUART_1		
J188	Jumper to select between RS232_DIN and FMC_H30.	
	Pin 1-2 RS232	Closed
	Pin 2-3 FMC	Open

Jumper	Function	Settings
J197	Jumper to select between RS232_ROUT and FMC_G29.	
	Pin 1-2 RS232	Closed
	Pin 2-3 FMC	Open

4.3. Board Connections

To provide the power to the SF2-DEV-KIT board, connect a power cable to the power jack of the board (J18).

To provide a serial interface to a PC / notebook plug a null-modem RS-232 cable to the J198 connector on the board. On the PC side, the serial link provides a serial console device to the SF2-DEV-KIT. 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/ttySn` device.

To provide network connectivity to the board, connect it into your LAN by plugging a standard Ethernet cable into the J4 connector. The board is pre-configured with an IP address of 192.168.0.2.

Connect the SF2-DEV-KIT board to a Microsemi FlashPro programmer tool by plugging a standard JTAG cable into the J59 connector. As soon as you have installed the U-Boot firmware and an appropriate Libero project onto the board (refer to Section 5.1), you can remove the connection to the FlashPro programmer.

You can also plug either a USB MicroA or a USB MicroB cable into the P1 USB Micro AB connector. It can be used to plug a USB Flash device, to make SmartFusion2 visible as a USB storage device to a USB host, and for other applications.

Apply the power to the SF2-DEV-KIT board using the on-board power switch.

5. SF2-DEV-KIT Board Linux Software Set-up

5.1. U-Boot Installation

U-Boot can be installed onto the Microsemi SmartFusion2 SF2-DEV-KIT board 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 SmartFusion2 SF2-DEV-KIT 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.

5.2. U-Boot Environment

When the SF2-DEV-KIT board is reset, the Linux bootstrap will proceed to boot the U-Boot firmware from the built-in Flash printing the following output to the serial console:

```
U-Boot 2010.03-linux-cortexm-1.10.1 (June 11 2013 - 19:43:46)

CPU  : SmartFusion2 SoC (Cortex-M3 Hard IP)
Freqs: CORTEX-M3=166MHz,PCLK0=83MHz,PCLK1=83MHz
Board: SF2-DEV-KIT
*** Warning - bad CRC, using default environment
```

```
DRAM:  64 MB
In:    serial
Out:   serial
Err:   serial
Net:   M2S_MAC
Hit any key to stop autoboot:  0
SF2-DEV-KIT>
```

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.

The reason for the warning about the "bad CRC" is that U-Boot is configured to store its environment variables in the external Flash. However, this being the first time when you boot Linux SmartFusion2 on the development board, obviously there is no U-Boot environment programmed to the external Flash. U-Boot goes to the external Flash, fails to find its environment there, prints the warning message and resorts to using the default environment integrated into the U-Boot image at build time.

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 SF2-DEV-KIT 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:

```
SF2-DEV-KIT> saveenv
Saving Environment to SPI Flash...
Erasing SPI flash...Writing to SPI flash...done
SF2-DEV-KIT>
```

Reset the SF2-DEV-KIT board and check that there is no warning about the bad CRC in the boot-up messages. This is expected since now U-Boot successfully finds its environment in the external Flash:

```
SF2-DEV-KIT> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.10.1 (June 11 2013 - 19:43:45)
...
Hit any key to stop autoboot:  0
SF2-DEV-KIT>
```

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:

```
SF2-DEV-KIT> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
SF2-DEV-KIT>
```

The default U-Boot environment for the SF2-DEV-KIT board sets `ethaddr` to a fixed MAC address. This address should work for you in a general case, however if you have more than two SF2-DEV-KIT boards in your LAN, use of the same address on multiple boards may result in packet collisions in your LAN and overall may render your LAN mal-functioning.

If you have more than one SF2-DEV-KIT boards in your LAN, you have to assign a unique MAC address to each board.

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

```
SF2-DEV-KIT> 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`:

```
SF2-DEV-KIT> setenv ipaddr 192.168.0.2
SF2-DEV-KIT> 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. 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:

```
SF2-DEV-KIT> setenv image vlad/networking.uImage
SF2-DEV-KIT> 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 Tue June 11 15:43:44 MSK 2013
...
```

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

```
SF2-DEV-KIT> setenv image vlad/networking.uImage
SF2-DEV-KIT> 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
SF2-DEV-KIT>
```

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

```
SF2-DEV-KIT> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.10.1 (June 11 2013 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (June 11 2013 - 17:19:37)
~ #
```

5.6. U-Boot Build

The BSP distribution comes with U-Boot pre-built for the SF2-DEV-KIT 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 sf2-dev-kit_config
Configuring for sf2-dev-kit board...
[psl@pvr u-boot]$ make -s
[psl@pvr u-boot]$ make -s u-boot.hex
```

6. Further Materials

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.