Linux Kinetis

BSP (Board Support Package) Guide for the Emcraft Systems K70 SOM Board

Release 1.8.0

Table of Contents

1. Overview

This document is a Linux Kinetis BSP (Board Support Package) Guide for the Emcraft Systems K70 SOM (System-On-Module) board, Release 1.8.0.

The BSP provides a software development environment for evaluation and development of Linux on the Cortex-M4 processor core of the Freescale Kinetis microcontroller using the Emcraft Systems K70 SOM board in harness with the Emcraft Systems SOM-BSB baseboard as a hardware platform.

This BSP is provided as part of the Emcraft Systems K70 SOM (System-On-Module) Starter Kit. The kit provides a hardware platform and Linux software development environment for the K70 SOM (System-On-Module).

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. K70 SOM board;
2. SOM-BSB baseboard;
3. Mini-USB cable.

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-1A-schem.pdf` - SOM-BSB schematics in PDF format;
2. `SOM-BSB-1A-bom.xls` - SOM-BSB Bill-Of-Materials (BOM) in Excel format;
3. `K70-SOM.IntLib` - Altium Designer 9.4 integrated library for the K70-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. `u-boot.bin` - prebuilt U-Boot file in the format suitable for installation into embedded Flash of Cortex-M4 on the K70 SOM board;
2. `networking.uImage` - prebuilt kernel image ready to be loaded to the K70 SOM board;
3. `linux-K70-1.8.0.tar.bz2` - Linux Kinetis 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 Kinetis.

2.4. Downloadable Documentation Materials

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

1. `k70-som-ha.pdf` - Emcraft Systems K70 SOM (System-On-Module) Hardware Architecture;
2. `k70-som-bsb-ha.pdf` - Emcraft Systems K70 SOM SOM-BSB Hardware Architecture;
3. `linux-cortexm-um-1.8.0.pdf` - Linux Cortex-M User's Manual;
4. `linux-K70-SOM-bspg-1.8.0.pdf` - Linux Kinetis BSP (Board Support Package) Guide for the Emcraft Systems K70 SOM Board (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux Kinetis, Release 1.8.0:

- 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 Kinetis;
  - 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 Kinetis 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 DMA interface;
  - Device driver for the USB host interface;
  - Framebuffer device driver;
  - Touchscreen 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. Provide optimal RAM timings for the K70 SOM.
   ID: RT 78138.
2. Develop a device driver for the USB-HS host interface for Linux Kinetis K70.
   ID: RT 79374.
3. Enable tickless kernel (`CONFIG_NO_HZ`).
   ID: RT 79812.
4. Validate support for USB-based WiFi connectivity in Linux Kinetis K70.
   ID: RT 78101.
5. Validate support for GPIO LEDs and pushbuttons in Linux Kinetis K70.
   ID: RT 78101.

3.3. Known Problems & Limitations

This section lists known problems and limitations of this release:

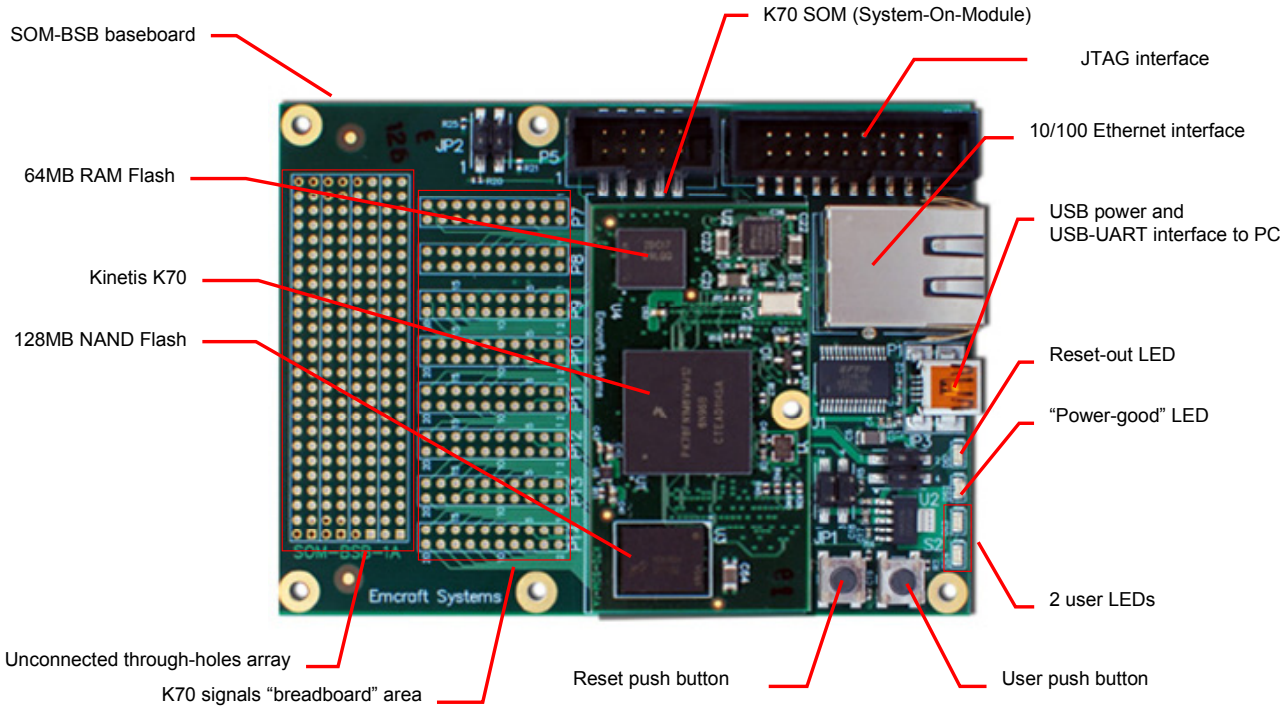
1. `CONFIG_KERNEL_IN_ENVM` requires disabling `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
   ID: RT 74683.
   Workaround: When enabling `CONFIG_KERNEL_IN_ENVM` in the kernel, disable `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
2. No support for hardware FPU on Cortex-M4.
   ID: RT 76490.
   Workaround: Float-point instructions are software-emulated in Linux Kinetis. Outside of a slower performance, applications that make use of float-point instructions work as if the hardware FPU was enabled.

4. Hardware Setup

This section explains how to set up the Emcraft Systems K70 SOM board in harness with the Emcraft Systems SOM-BSB baseboard in such a way as to allow running uClinux on this hardware platform.

4.1. Hardware Interfaces

The K70 SOM board in harness with the Emcraft Systems SOM-BSB baseboard provides the following components and interfaces:



4.2. Jumpers

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

Jumper	Configuration	Notes
JP1	1-2 closed, 3-4 closed	To enable power on the K70 SOM (VCC3) and save the battery life when the mini-USB is connected
JP2	1-2 open, 3-4 closed	The settings of JP2 do not affect the K70 SOM
JP3	1-3 open, 2-4 closed	To use the mini-USB port as the power source

4.3. Board Connections

To power the SOM-BSB baseboard with the K70 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 board. As soon as the connection to the PC has been made, the on-board LED DS2 should lit, indicating that the board is up and running.

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

To provide network connectivity to the board, connect it into your LAN by plugging a standard Ethernet cable into the 10/100 Ethernet connector.

4.4. Extension Interfaces

For description of the extension interfaces provided by the Emcraft Systems K70 SOM board refer to *Emcraft Systems K70 SOM (System-On-Module) Hardware Architecture*.

For description of the extension interfaces provided by the Emcraft Systems K70 SOM SOM-BSB baseboard refer to *Emcraft Systems SOM-BSB Hardware Architecture*.

5. K70 SOM Board Software Set-up

5.1. U-Boot Environment

When the K70 SOM board is reset, U-Boot comes up from the built-in Flash printing the following output to the serial console:

```
U-Boot 2010.03-linux-cortexm-1.8.0 (Sep 07 2012 - 19:43:46)

CPU  : Freescale Kinetis series (Cortex-M4)
Freqs: SYSTICK=120MHz,CCLK=120MHz,PCLK=60MHz,MACCLK=50MHz
Board: K70-SOM Rev 1.A, www.emcraft.com
DRAM:  64 MB
NAND:  128 MiB
Bad block table found at page 65472, version 0x01
Bad block table found at page 65408, version 0x01
In:    serial
Out:   serial
Err:   serial
Net:   FEC0
Hit any key to stop autoboot:  0
K70-SOM>
```

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 K70 SOM board.

This is how you can write the current U-Boot environment to the external Flash:

```
K70-SOM> saveenv
Saving Environment to NAND...
...
K70-SOM>
```

5.2. Ethernet MAC Address

In Linux Kinetis, 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:

```
K70-SOM> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
K70-SOM>
```

The K70 SOM board comes with `ethaddr` set to a MAC address uniquely allocated for the specific board. Given that each K70 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).

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`:

```
K70-SOM> setenv ipaddr 192.168.0.2
K70-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:

```
K70-SOM> saveenv
Saving Environment to NAND...
...
K70-SOM>
```

5.4. Installation of Linux Images to Flash

The K70 SOM board arrives with a Linux bootable image for the `networking` project installed into external Flash. To boot this Linux configuration onto the K70 SOM board just reset the board and let U-Boot perform the autoboot sequence.

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 in the auto-boot mode.

On the host, activate the Linux Kinetis 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:

```
K70-SOM> setenv image vlad/networking.uImage
K70-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 Mon Mar 12 15:43:44 MSK 2012
...
```

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

```
K70-SOM> setenv image vlad/networking.uImage
K70-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)

NAND erase: device 0 offset 0x100000, size 0x1f00000
Erasing at 0x1fe0000 -- 100% complete.
OK

NAND write: device 0 offset 0x100000, size 0x1f00000
 32505856 bytes written: OK
K70-SOM>
```

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

```
K70-SOM> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.8.0 (Sep 07 2012 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (Sep 07 2012 - 17:19:37)
~ #
```

5.5. U-Boot Build

The BSP distribution comes with U-Boot pre-built for the K70 SOM board. If however you need to re-build U-Boot for your board, please follow the instructions below:

1. Install the Linux Kinetis distribution to the development host, as described in the Linux Cortex-M User's Manual.
2. From the top of the Linux Kinetis installation, activate the Linux Kinetis 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 k70-som_config
Configuring for k70-som board...
[psl@pvr u-boot]$ make -s
```

5.6. U-Boot Installation

The Emcraft Systems K70 SOM board arrives with the U-Boot firmware pre-installed into the on-chip Flash of the Kinetis. 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 Kinetis, U-Boot can be re-installed using a P&E USB MultiLink Universal JTAG debugger (not included in the kit) and the Freescale CodeWarrior IDE. Please follow the procedure described below:

1. On a Windows host download and install the Special Edition of CodeWarrior 10.2 or higher from the Freescale web site at:

   http://www.freescale.com/cwmcu10

2. Open the cover of the P&E USB MultiLink Universal debugger and programmer device;
3. Remove all cables currently connected to the P&E USB MultiLink Universal. Connect the 20-pin flat ribbon cable to the `PORT B - STANDARD ARM` group of pins. The red pin must be oriented towards pin 1;
4. Connect the other end of the ribbon cable to the `P3` connector of the SOM-BSB baseboard. The red pin must also be oriented towards pin 1;
5. Connect the SOM-BSB board to a Windows host using a mini-USB cable;
6. Connect the P&E programmer to a Windows host using a USB cable.
7. On the Windows development host, open the Device Manager and install the driver for the `USB-ML-12 (FS) Rev A` device. The driver is already available on your Windows system; it is part of the CodeWarrior installation. After the driver installation, the blue LED on the P&E programmer should be on.
8. On the Windows host run the CodeWarrior for Microcontrollers 10.2:
   a) Create a project by selecting `File -> New -> Bareboard Project`. Choose `Kinetis -> K70 Family -> K70F (120 MHz) Family -> MK70FN1M0` from the list of devices to connect to. On the next page, in the list of connection types check only the `P&E USB MultiLink Universal [FX] / USB MultiLink`, after that you can click `Finish`;
   b) Open the `Target Tasks` view. It is accessible via `Window -> Show View -> Other...`;
   c) In the `Target Tasks` view, create a new task. The `Run Configuration` should be `[project-name]_MK70FN1M0_INTERNAL_FLASH_PnE U-MultiLink`, the `Task Type` should be `Flash Programmer for Kinetis`;
   d) In the `Kinetis Flash Programmer Task` view, click the `Add Device` button under the list of devices. In the dialog box choose `FTFE_PFlash1M0 (128Kx64x1)` as the type of the internal flash and click `Add Device`;
   e) In the `Kinetis Flash Programmer Task` view, in the `Target RAM` group box, enter the following region of internal SRAM memory: `Address: 1fff0000`, `Size: 00010000`;
   f) In the `Kinetis Flash Programmer Task` view, click `Add Action -> Program / Verify`. Choose the U-Boot image (`u-boot.bin`) from the file system, select the `Erase sectors before program` checkbox and click `Add Program Action`;
   g) In the `Target Tasks` view, click the right mouse button on your flash programming task and select `Execute`.
9. After programming the Kinetis using CodeWarrior, disconnect the 20-pin Standard ARM JTAG cable from the SOM-BSB board and re-plug the mini-USB cable into the SOM-BSB board.

6. Further Materials

Refer to *Emcraft Systems K70 SOM (System-On-Module) Hardware Architecture* for detailed information on the hardware architecture of the Emcraft Systems K70 SOM board.

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

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

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

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 6 months since the product purchase;
- Free upgrade to new releases of the downloadable materials included in the product during 6 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.