

Linux Kinetis

BSP (Board Support Package) Guide for the Freescale TWR-K70F120M Board

Release 1.12.1

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	3
3. SOFTWARE FUNCTIONALITY	4
3.1. SUPPORTED FEATURES.....	4
3.2. NEW AND CHANGED FEATURES	5
3.3. KNOWN PROBLEMS & LIMITATIONS.....	5
4. HARDWARE SETUP	5
4.1. HARDWARE INTERFACES	5
4.2. JUMPERS	5
4.3. BOARD CONNECTIONS	6
5. TWR-K70F120M BOARD LINUX SOFTWARE SET-UP	6
5.1. U-BOOT INSTALLATION.....	6
5.2. U-BOOT ENVIRONMENT	7
5.3. ETHERNET MAC ADDRESS.....	8
5.4. NETWORK CONFIGURATION	8
5.5. LOADING LINUX IMAGES	8
5.6. U-BOOT BUILD.....	10
6. FURTHER MATERIALS	10
7. SUPPORT	10

1. Overview

This document is a Linux Kinetis BSP (Board Support Package) Guide for the Freescale TWR-K70F120M board, Release 1.12.0.

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 Freescale TWR-K70F120M 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 TWR-K70F120M board from Freescale 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 TWR-K70F120M board directly from Freescale.

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.srec` - prebuilt U-Boot file in the format suitable for installation into embedded Flash of Cortex-M4 on the TWR-K70F120M board;
2. `networking.uImage` - prebuilt Linux image ready to be loaded to the TWR-K70F120M board;
3. `linux-k70-1.12.0.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. `linux-cortexm-um-1.12.0.pdf` - Linux Cortex-M User's Manual;
2. `linux-TWR-K70F120M-bsp-1.12.0.pdf` - Linux Kinetis BSP (Board Support Package) Guide for the Freescale TWR-K70F120M Board (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux Kinetis, 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 Kinetis;
 - Serial device driver and Linux console;
 - Ethernet device driver and networking (`ping`, NFS, Telnet, FTP, `ntpd`, etc.);
 - `busybox` v1.17;
 - POSIX `pthread`s;
 - Hardened exception handling; an exception triggered by a process affects only the offending process;
 - Loadable kernel modules;
 - Support for the hardware FPU;
 - 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;
 - ADC 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. Enable access to GPIO interrupts from user space in Kinetis Linux.
ID: RT 86844.
2. Develop an SPI device driver for the uClinux Kinetis.
ID: RT 88844.
3. Develop a Linux device driver for the Kinetis USB FS interface.
ID: RT 89239.
4. Clean in the Kinetis Linux SD Card device driver.
ID: RT 91454.

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. 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. Hardware Setup

This section explains how to set up the Freescale TWR-K70F120M board.

4.1. Hardware Interfaces

For a description of the hardware interfaces provided by the TWR-K70F120M board, refer to detailed technical documentation and other associated materials available from Freescale.

4.2. Jumpers

The following jumpers must be configured on the TWR-K70F120M and TWR-SER boards:

Board	Jumper	Configuration	Notes
TWR-K70F120M	J19	open	To disable the on-board 50 MHz oscillator

Board	Jumper	Configuration	Notes
TWR-K70F120M	J18	close	Only on the schematics Rev C boards. To provide proper Ethernet clocking
TWR-SER	J2	3-4 closed	To select 50 MHz for the Ethernet PHY clock
TWR-SER	J3	2-3 closed	To route the 50 MHz clock to CLOCKINO
TWR-SER	J12	9-10 closed	To select the RMII mode of the Ethernet PHY

4.3. Board Connections

To power the TWR-K70F120M board up, simply connect it to a PC / notebook by plugging a mini-USB cable into the J13 mini-USB connector on the TWR-K70F120M board. As soon as the connection to the PC has been made, the various on-boards LEDs should lit up, indicating that the board is up and running.

To provide a serial interface to a PC / notebook plug a null-modem RS-232 cable to the TWR-SER J8 serial connector on the board. On the PC side, the serial link provides a serial console device to the TWR-K70F120M. 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 TWR-SER 10/100 Ethernet connector. The board is pre-configured with an IP address of 192.168.0.2.

5. TWR-K70F120M Board Linux Software Set-up

5.1. U-Boot Installation

To install U-boot onto the Freescale TWR-K70F120M, follow the step-wise procedure documented below:

1. Connect the TWR-K70F120M board to a Windows host using a mini-USB cable;
2. On the 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>
3. 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 `Open Source JTAG`, 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_OSJTAG`, 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.srec`) 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`.
 - o If the CodeWarrior IDE asks you to update the OSBDM/OSJTAG bootloader, follow its instructions.
 - o If the CodeWarrior IDE cannot find your board connected to the Windows host, go to the Windows Device Manager and reinstall the drivers for the USB devices listed.
4. After programming the flash using CodeWarrior, re-plug the mini-USB cable into the TWR-K70F120M board;
5. Connect the UART port of the TWR-K70F120M board to a host;
6. Run a terminal program (e.g. HyperTerminal on Windows, or `kermit` on Linux) and create a serial connection to the TWR-K70F120M board with the following COM-port settings: 115200 8N1;
7. Reset the TWR-K70F120M board and see the U-Boot start-up banner in the terminal program.

5.2. U-Boot Environment

When the TWR-K70F120M 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-1.12.0 (Dec 6 2013 - 17:19:38)

CPU : Freescale Kinetis series (Cortex-M4)
Freqs: SYSTICK=120MHz,CCLK=120MHz,PCLK=60MHz,MACCLK=50MHz
Board: Freescale TWR-K70F120M rev 1
DRAM: 128 MB
NAND: 256 MiB
Bad block table found at page 131008, version 0x01
Bad block table found at page 130944, version 0x01
*** Warning - bad CRC or NAND, using default environment

In: serial
Out: serial
Err: serial
Net: FEC0
Hit any key to stop autoboot: 0
TWR-K70F120M>
```

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 Kinetis 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 TWR-K70F120M 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:

```
TWR-K70F120M> saveenv
Saving Environment to NAND...
...
TWR-K70F120M>
```

Reset the TWR-K70F120M 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:

```
TWR-K70F120M> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 6 2013 - 19:43:45)
...
Hit any key to stop autoboot:  0
TWR-K70F120M>
```

5.3. 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:

```
TWR-K70F120M> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
TWR-K70F120M>
```

The default U-Boot environment for the TWR-K70F120M 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 TWR-K70F120M 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 TWR-K70F120M 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:

```
TWR-K70F120M> 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`:

```
TWR-K70F120M> setenv ipaddr 192.168.0.2
TWR-K70F120M> 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 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:

```
TWR-K70F120M> setenv image vlad/networking.uImage
TWR-K70F120M> 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 6 15:43:44 MSK 2013
...

```

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

```
TWR-K70F120M> setenv image vlad/networking.uImage
TWR-K70F120M> 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)

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
TWR-K70F120M>
```

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

```
TWR-K70F120M> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 6 2013 - 17:19:37)
...
Starting kernel ...
...
init started: BusyBox v1.17.0 (Dec 6 2013 - 17:19:37)
~ #
```

5.6. U-Boot Build

The BSP distribution comes with U-Boot pre-built for the TWR-K70F120M 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 twr-k70f120m_config
Configuring for twr-k70f120m board...
[psl@pvr u-boot]$ make -s
[psl@pvr u-boot]$ make -s u-boot.srec
```

6. Further Materials

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 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.