Linux LPC17XX

BSP (Board Support Package) Guide for the Emcraft Systems LPC-LNX-EVB Board

Release 1.12.0

Table of Contents

1. Overview

This document is a Linux LPC17XX BSP (Board Support Package) Guide for the Emcraft Systems LPC-LNX-EVB board, 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 NXP LPC17XX microcontroller using the Emcraft Systems LPC-LNX-EVB board as a hardware platform.

This BSP is provided as part of the Emcraft Systems Linux LPC17XX Evaluation Kit. The evaluation kit provides a hardware platform and Linux software development environment for the NXP LPC17XX microcontroller.

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. LPC-LNX-EVB board;
2. Mini-USB cable.

2.2. Downloadable Hardware Materials

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

1. `LPC-LNX-EVB-2_0-schem.pdf` - LPC-LNX-EVB schematics in PDF format;
2. `LPC-LNX-EVB-2_0-bom.xls` - LPC-LNX-EVB Bill-Of-Materials (BOM) in Excel format.

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.hex` - prebuilt U-Boot file in the format suitable for installation into embedded Flash of Cortex-M3 on the LPC-LNX-EVB board;
2. `networking.uImage` - prebuilt Linux image ready to be loaded to the LPC-LNX-EVB board;
3. `linux-LPC-1.12.0.tar.bz2` - Linux LPC17XX 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 LPC17XX.

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-LPC-LNX-EVB-bspg-1.12.0.pdf` - Linux LPC17XX BSP (Board Support Package) Guide for the Emcraft Systems LPC-LNX-EVB Board (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux LPC17XX, 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 LPC17XX;
  - 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 LPC17XX 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;
  - SD Card device driver;
  - I2C device driver;
  - I2S device driver;

  - RTC device driver;
  - Framebuffer device driver;
  - SPI controller master-mode device driver;
  - Support for RS-485 in the serial device driver;
  - Watchdog 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:

- None.

3.3. Known Problems & Limitations

This section lists known problems and limitations of this release:

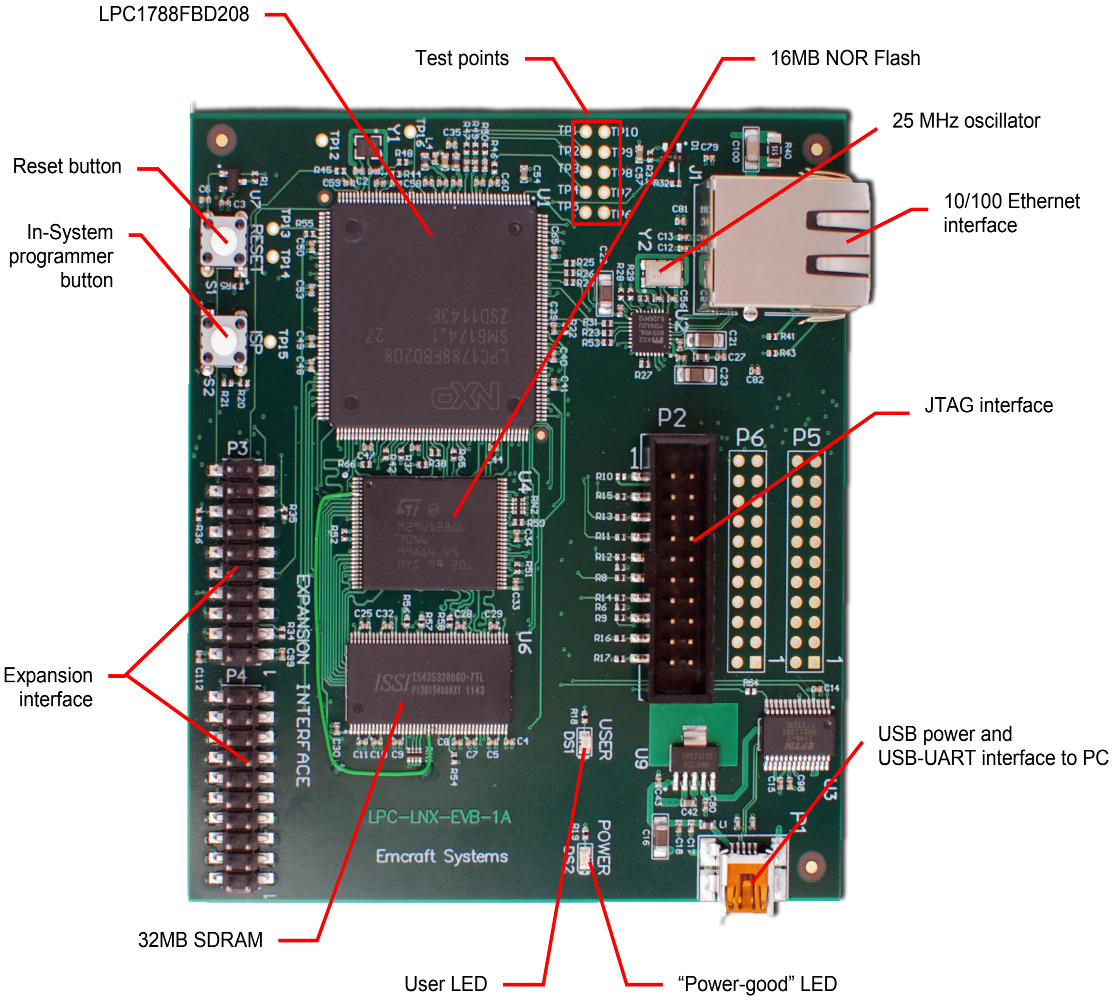
1. `CONFIG_KERNEL_IN_ENVM` requires disabling `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
   ID: RT 74683.
   Workaround: When enabling `CONFIG_KERNEL_IN_ENVM` in the kernel, disable `CONFIG_ARM_UNWIND` and `CONFIG_EARLY_PRINTK`.
2. Debugging of multi-threaded applications using GDB is not supported.
   ID: RT 77243.
   Workaround: None. This issue will be resolved in future releases of the product.
3. If the LPC-LNX-EVB board has been reset using a reset command of U-Boot, Ethernet may be mis-configured.
   ID: RT 78884.
   Workaround: Use hardware reset (power off/on or pushbutton).

4. Hardware Setup

This section explains how to set up the Emcraft Systems LPC-LNX-EVB board.

4.1. Hardware Interfaces

The LPC-LNX-EVB board provides the following components and interfaces:



4.2. Board Connections

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

On the PC side, the USB link provides a serial console device to the LPC-LNX-EVB. 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 10/100 Ethernet connector. The board is pre-configured with an IP address of 192.168.0.2.

4.3. Extension Interfaces

The Emcraft Systems LPC-LNX-EVB provides the following interfaces that can be used to prototype custom hardware extensions to the main hardware design of the board as well as to develop new software drivers and applications.

4.3.1. User LEDs

LED	LPC17XX Pin
CPU_LED	P5.1

4.3.2. Expansion Interface

P3 Pin	LPC17XX Pin	P3 Pin	LPC17XX Pin
1	VCC3	2	VCC3
3	P0.17	4	P0.18
5	P0.15	6	P0.16
7	P0.22	8	P0.20
9	P0.11	10	P2.7
11	GND	12	GND
13	P0.1	14	P0.0
15	P5.2	16	P5.3
17	GND	18	GND
19	P0.12	20	P0.13
P4 Pin	**LPC17XX Pin**	**P4 Pin**	**LPC17XX Pin**
1	GND	2	GND
3	P0.31	4	P0.30
5	P0.29	6	P0.28
7	GND	8	GND
9	P0.27	10	P2.15
11	P0.21	12	P0.19
13	GND	14	GND
15	P0.14	16	P5.0
17	P2.10	18	P1.31
19	P1.19	20	P1.18

4.3.3. Test Points

Test Point #	LPC17XX Pin	Test Point #	LPC17XX Pin
1	P0.23	9	GND
2	P0.24	10	GND
3	P0.25	-	-
4	P0.26	12	P2.27
5	P1.30	13	P2.26
6	GND	14	P2.23
7	GND	15	P2.22
8	GND	16	nRSTOUT

5. LPC-LNX-EVB Board Linux Software Set-up

5.1. U-Boot Environment

When the LPC-LNX-EVB 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 06 2013 - 19:43:45)

CPU  : LPC178x/7x series (Cortex-M3)
Freqs: SYSTICK=108MHz,EMCCLK=54MHz,PCLK=54MHz
Board: LPC-LNX-EVB rev 1.A, www.emcraft.com
DRAM:  32 MB
Flash: 16 MB
In:    serial
Out:   serial
Err:   serial
Net:   LPC178X_MAC
Hit any key to stop autoboot:  0
LPC-LNX-EVB>
```

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 LPC-LNX-EVB 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:

```
LPC-LNX-EVB> saveenv
Saving Environment to Flash...
...
LPC-LNX-EVB>
```

5.2. Ethernet MAC Address

In Linux LPC17XX, 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:

```
LPC-LNX-EVB> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
LPC-LNX-EVB>
```

The LPC-LNX-EVB board comes with `ethaddr` set to a MAC address uniquely allocated for the specific board. Given that each LPC-LNX-EVB 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:

```
LPC-LNX-EVB> 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`:

```
LPC-LNX-EVB> setenv ipaddr 192.168.0.2
LPC-LNX-EVB> 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 LPC-LNX-EVB board comes with a Linux bootable image for the `networking` project installed into external Flash. To boot this Linux configuration onto the LPC-LNX-EVB 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 LPC17XX 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:

```
LPC-LNX-EVB> setenv image vlad/networking.uImage
LPC-LNX-EVB> 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:

```
LPC-LNX-EVB> setenv image vlad/networking.uImage
LPC-LNX-EVB> 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)
................................ done
Un-Protected 32 sectors

................................ done
Erased 32 sectors
Copy to Flash... done
LPC-LNX-EVB>
```

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

```
LPC-LNX-EVB> 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 LPC-LNX-EVB board. If however you need to re-build U-Boot for your board, please follow the instructions below:

1. Install the Linux LPC17XX distribution to the development host, as described in the *Linux Cortex-M User's Manual*.
2. From the top of the Linux LPC17XX installation, activate the Linux LPC17XX 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 lpc-lnx-evb_config
Configuring for lpc-lnx-evb board...
[psl@pvr u-boot]$ make -s
[psl@pvr u-boot]$ make -s u-boot.hex
```

5.7. U-Boot Installation

The Emcraft Systems LPC-LNX-EVB board arrives with the U-Boot firmware pre-installed into the on-chip Flash of the LPC17XX. 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 LPC17XX, U-Boot can be re-installed using FlashMagic software tool. Please follow the procedure described below:

1. Connect the LPC-LNX-EVB board to a Windows host using a mini-USB cable;
2. Open the Windows Device Manager and determine the COM port number the board is available at in Windows;
3. Get the FlashMagic software at http://www.flashmagictool.com/ and install it;

4. Run FlashMagic and configure it as follows:
   a) Set device type to LPC1788;
   b) Set COM Port number to the value you already know from the Windows Device Manager;
   c) Set Baud Rate to `57600`;
   d) Set Interface to `None (ISP)`;
   e) Check the option `Erase blocks used by Hex File`.
5. On the LPC-LNX-EVB board, hold down the S2 button while resetting the board (push S1) to put it in ISP mode.
6. In FlashMagic, select u-boot.hex from the file system and click Start;
7. Reset the board after programming U-Boot.

6. Further Materials

Refer to *Linux Cortex-M User's Manual* for detailed information on the software architecture of the Linux LPC17XX distribution.

Visit Emcraft Systems' web site at www.emcraft.com to obtain additional materials related to Linux LPC17XX.

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.