Linux LPC18XX

BSP (Board Support Package) Guide for the Hitex LPC1850 Eval Board

Release 1.12.0

Emcraft Systems

Table of Contents

1. Overview

This document is a Linux LPC18XX BSP (Board Support Package) Guide for the Hitex LPC1850 Eval 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 LPC18XX microcontroller using the Hitex LPC1850 Eval 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 LPC1850 Eval board from Hitex 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 LPC1850 Eval board directly from Hitex.

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-M3 on the LPC1850 Eval board;

2. `networking.uImage` - prebuilt Linux image ready to be loaded to the LPC1850 Eval board;

3. `linux-LPC18XX-1.12.0.tar.bz2` - Linux LPC18XX 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 LPC18XX.

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-LPC1850-EVAL-bspg-1.12.0.pdf` - Linux LPC18XX BSP (Board Support Package) Guide for the Hitex LPC1850 Eval Board (this document).

3. Software Functionality

3.1. Supported Features

The following list summarizes the features and capabilities of Linux LPC18XX, Release 1.12.0:

- 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 LPC18XX;
  - 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 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.
- 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:

- 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.

4.   Hardware Setup

This section explains how to set up the Hitex LPC1850 Eval board.

4.1. Hardware Interfaces

For a description of the hardware interfaces provided by the LPC1850 Eval board, refer to detailed technical documentation and other associated materials available from Hitex.

4.2. Jumpers

The following jumpers must be configured on the LPC1850 Eval board:

Jumper	Configuration	Notes
JP5	2-3 closed	Ethernet: MII mode
JP14	2-3 closed	Ethernet: MII mode
JP15	open	Ethernet: MII mode
JP23	open	Disable write protection of NOR flash
SV1	only 5-6 and 7-8 closed	Connect USART0 to the DB9 port "UART"
SV3	all closed	Connect Ethernet signals
SV6	all closed	Connect Ethernet signals
SV12	all open	Disconnect signals optionally shared with Ethernet
SV13	all open	Disconnect signals optionally shared with Ethernet

4.3. Board Connections

To power the LPC1850 Eval board up, simply connect it to a PC / notebook by plugging a USB-B cable into the X2 USB-B connector on the LPC1850 Eval 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 RS-232 cable to the X1 "UART" connector on the board. On the PC side, the serial link provides a serial console device to the LPC1850 Eval. 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/ttyS`*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.

5. LPC1850 Eval Board Linux Software Set-up

5.1. U-Boot Installation

To install U-boot onto the Hitex LPC1850 Eval, follow the step-wise procedure documented below:

1. Configure the boot select jumpers according to the table below for booting from USB0:

Jumper	Configuration
BOOT1	1-2 closed
BOOT2	2-3 closed
BOOT3	1-2 closed
BOOT4	2-3 closed

2. Connect the UART port of the LPC1850 Eval board to a host.

3. Run a terminal program (e.g. HyperTerminal on Windows, or kermit on Linux) and create a serial connection to the LPC1850 Eval board with the following COM-port settings: 115200 8N1.

4. Connect the LPC1850 Eval board to a Linux host by plugging a USB-B cable into the X2 USB-B connector on the LPC1850 Eval board.

5. Install the free tool `dfu-util` on the Linux host either from your Linux distribution packages or by building it from sources. The sources of `dfu-util` are available for download at http://dfu-util.gnumonks.org/.

6. Boot U-Boot on the LPC1850 Eval board by running `dfu-util` on the Linux host.

```
[psl@skywanderer ~]$ sudo dfu-util -R -D ~/Downloads/u-boot.bin
dfu-util 0.7

Copyright 2005-2008 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2012 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to dfu-util@lists.gnumonks.org

Opening DFU capable USB device... ID 1fc9:000c
Run-time device DFU version 0100
Claiming USB DFU Runtime Interface...
Determining device status: state = dfuIDLE, status = 0
WARNING: Runtime device already in DFU state ?!?
Found Runtime: [1fc9:000c] devnum=0, cfg=1, intf=0, alt=0, name="DFU"
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 0100
Device returned transfer size 2048
No valid DFU suffix signature
Warning: File has no DFU suffix
bytes per hash=1874
Copying data from PC to DFU device
```

```
Starting download: [#############################################] finished!
unable to read DFU status
can't detach
Resetting USB to switch back to runtime mode
[psl@skywanderer ~]$
```

7. Start a TFTP server in the local network of the LPC1850 Eval board and make the `u-boot.bin` image available on this TFTP server.

8. Set the U-Boot environment variables `serverip` and `ipaddr` to the IP address of the TFTP server and the desired IP address of the LPC1850 Eval board, respectively.

```
LPC1850-EVAL> setenv serverip 192.168.0.10
LPC1850-EVAL> setenv ipaddr 192.168.0.18
LPC1850-EVAL>
```

9. Download the `u-boot.bin` image into the on-board RAM.

```
LPC1850-EVAL> tftp lpc18xx/u-boot.bin
Auto-negotiation...completed.
LPC18XX MAC: link UP (100/Full)
Using LPC18XX MAC device
TFTP from server 192.168.0.10; our IP address is 192.168.0.18
Filename 'lpc18xx/u-boot.bin'.
Load address: 0x28000000
Loading: #######
done
Bytes transferred = 93200 (16c10 hex)
LPC1850-EVAL>
```

10. Write the `u-boot.bin` image into the beginning of the on-board NOR Flash.

```
LPC1850-EVAL> era 1c000000 +${filesize}

...................... done
Erased 23 sectors
LPC1850-EVAL> cp.b ${fileaddr} 1c000000 ${filesize}
Copy to Flash... done
LPC1850-EVAL>
```

11. Reconfigure the boot select jumpers according to the table below for booting from NOR Flash:

Jumper	Configuration
BOOT1	1-2 closed
BOOT2	1-2 closed
BOOT3	2-3 closed
BOOT4	2-3 closed

12. Reset the LPC1850 Eval board and see the U-Boot start-up banner in the terminal program.

5.2. U-Boot Environment

When the LPC1850 Eval board is reset, the Linux bootstrap will proceed to boot the U-Boot firmware from the external NOR Flash printing the following output to the serial console:

```
U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 19:43:46)

CPU  : LPC18xx series (Cortex-M3)
Freqs: SYSTICK=180MHz,CCLK=180MHz
Board: Hitex LPC1850 Eval rev 1
DRAM:   8 MB
Flash:  4 MB
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   LPC18XX_MAC
Hit any key to stop autoboot:  0
```

```
LPC1850-EVAL>
```

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 LPC18XX 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 LPC1850 Eval 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:

```
LPC1850-EVAL> saveenv
Saving Environment to Flash...
...
LPC1850-EVAL>
```

Reset the LPC1850 Eval 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:

```
LPC1850-EVAL> reset
resetting ...

U-Boot 2010.03-linux-cortexm-1.12.0 (Dec 06 2013 - 19:43:45)
...
Hit any key to stop autoboot:  0
LPC1850-EVAL>
```

5.3. Ethernet MAC Address

In Linux LPC18XX, 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:

```
LPC1850-EVAL> printenv ethaddr
ethaddr=C0:B1:3C:88:88:88
LPC1850-EVAL>
```

The default U-Boot environment for the LPC1850 Eval 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 LPC1850 Eval 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 LPC1850 Eval 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:

```
LPC1850-EVAL> 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`:

```
LPC1850-EVAL> setenv ipaddr 192.168.0.2
LPC1850-EVAL> 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 LPC18XX 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:

```
LPC1850-EVAL> setenv image vlad/networking.uImage
LPC1850-EVAL> 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:

```
LPC1850-EVAL> setenv image vlad/networking.uImage
LPC1850-EVAL> 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
LPC1850-EVAL>
```

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

```
LPC1850-EVAL> 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 LPC1850 Eval board. If however you need to re-build U-Boot for your board, please follow the instructions below:

1. Install the Linux LPC18XX distribution to the development host, as described in the Linux Cortex-M User's Manual.

2. From the top of the Linux LPC18XX installation, activate the Linux LPC18XX 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 lpc1850-eval config
Configuring for lpc1850-eval board...
[psl@pvr u-boot]$ make -s
```

6. Further Materials

Refer to *Linux Cortex-M User's Manua*l for detailed information on the software architecture of the Linux LPC18XX distribution.

Visit Emcraft Systems' web site at www.emcraft.com to obtain additional materials related to Linux LPC18XX.

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.