OMAP-L138 LCDK Linux SDK Getting Started Guide

Welcome to the OMAP-L138 LCDK Getting Started Guide (GSG). It's purpose is to walk you through the steps for setting up the OMAP-L138 LCDK Linux Software Development Kit (SDK) on your host Linux development computer. If you are new to embedded Linux and OMAP-L138 LCDK, you should proceed through this guide in the order given for best results. By the end of this Getting Started Guide you will have the Linux host development environment configured to be able to create, compile a "hello world" application, download and run it on the LCDK.

Overview

This Linux SDK was put together to provide you the necessary tools to develop Linux applications on the L138/ C6748 Development Kit (LCDK).
The following table lists the major software components that were included and their purposes:

LCDK Linux SDK Main Software Components

Software Components	Purpose
U-boot source	LCDK Bootloader. Loads Linux kernel from MMC/SD, NAND or NFS
Linux Kernel source	Hi-level operating system with drivers and I/O stacks running on ARM9
ARMv5 File System	Bootable target file system that can be mounted via SD card or NFS
GCC Tool Chain	Open source GNU tools to cross compile ARM target programs
SysBIOS	Real-Time OS for C674 DSP
SysLink	Inter-Processor Communication (IPC) software to communicate between Linux and DSP applications
C6x Tool Chain	TI C6000 ISA tool chain to build DSP-side applications

Conventions

Command prompts in this guide

Commands are preceded by prompts that indicate the environment where the command is to be typed. For example:

host $
Indicates command to be typed into the shell window of the host Linux workstation.
EVM #
Indicates commands to be typed into the U-Boot shell in a console window connected to the LCDK board's serial port.
target $
Indicates commands to be entered into LCDK Linux shell in the serial terminal window connected to the LCDK board's serial port.

NOTE: The document lists various commands that need to be executed on the target's Linux shell or on the u-boot prompt. Some of these commands are quite long and may span several lines on this page as one command. These commands will be indicated as "enter as one command". A direct copy and paste of these commands might result in insertion of line breaks. Hence it would not work on the u-boot prompt or the target. So to be on the safe side, copy these long commands into a text editor (like notepad for Windows™), replace the line break between them with a space and then copy and paste it on the command prompts.

Version numbering

Throughout the document, sometimes package names are labeled with xx.xx.xx. This in general is the "major.minor.sub-minor" versioning scheme and should be substituted with the real version number of the same format. For example when the file is ti-sdk-omapl138-lcdk-xx.xx.xx.bz2, the real file could be ti-sdk-omapl138-lcdk-01.00.00.bz2.

SDK Installation

Pre-install Preparation

To begin application development with the LCDK, we recommend the following tools:
- Linux Host machine running Ubuntu 10.04 LTS to build code and as a TFTP and NFS server.
- Serial terminal application e.g. TeraTerm for Windows™ or Minicom for Linux OSes.
- Ethernet cable(s) and optionally a router to act as DHCP server and switch to connect between your Linux host computer and the LCDK.
- Other needed hardware such as VGA monitor, speakers, etc. if your application needs the related peripherals.

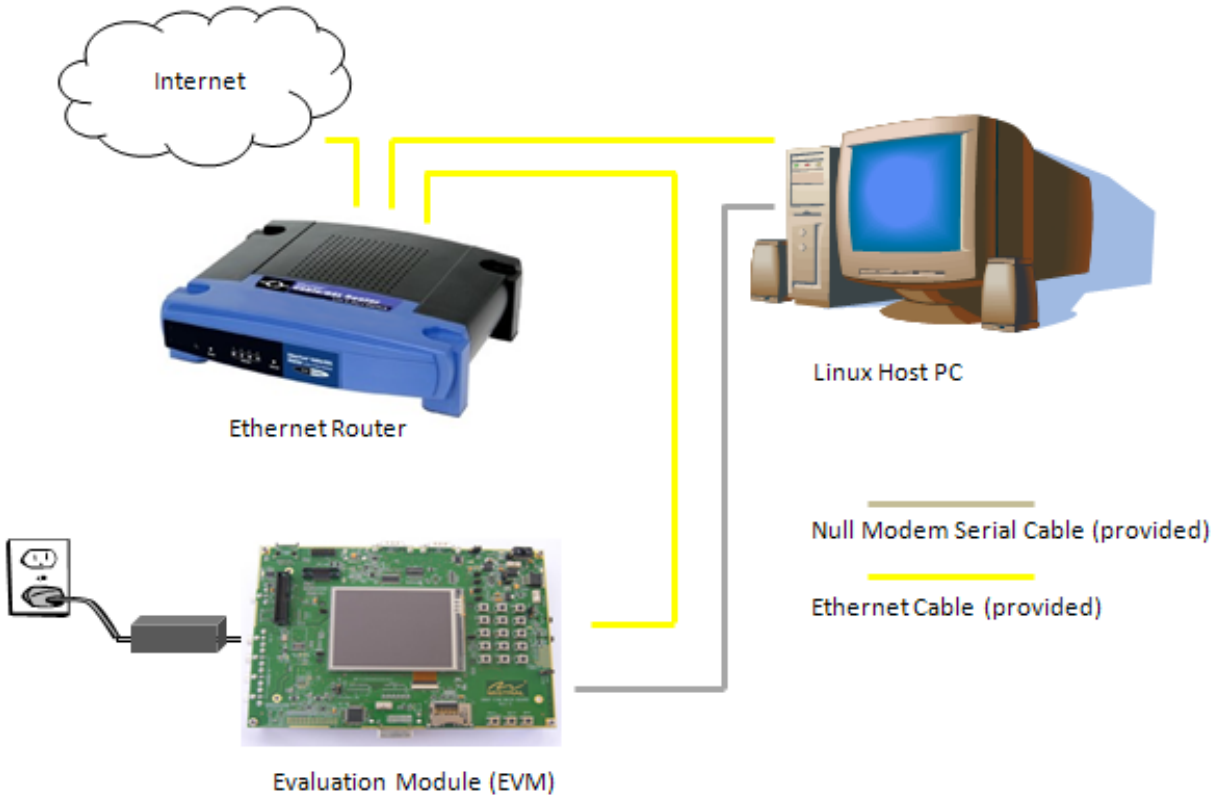
The procedures outlined in this GSG assume that you are working with a Linux host machine (i.e. PC). Our reference Linux host version is Ubuntu 10.04 LTS.

If you are new to Linux in general and are looking for information on how to set up Linux host machine, please use the "Linux Host Support" links provided in the "Resource" section below for additional information.

To see the output of Linux running on LCDK, you will need a serial terminal program. Several are available for both Windows ™ (if you are running a virtual Linux machine within Windows) and Linux host environments. The setting should be *115200 baud, 8 bit data, no parity, 1 stop-bit and no flow control*. Additional information is provided under "Serial Terminal" of the "Resource" section below.

The router with DHCP server is convenient although not necessary. What's really needed is a way for the LCDK to connect to your Linux host machine via Ethernet. This can be done using a router with built-in DHCP server and Ethernet switch, or using static IP. If your Linux host machine is in a larger network and the LCDK board can be plugged to this network to get an IP address via DHCP, then only a switch is needed. Once the Linux host machine and LCDK are on the same subnet, they can communicate to each other making it much easier to share file and file systems. These methods will be discussed in later procedures listed in the "Next Steps" section.

A typical embedded Linux development environment set up looks like the following picture:



NOTE: the picture is for a generic EVM. For the LCDK, the null serial cable is replaced by the USB cable. Please refer to the Quick Start Guide booklet included with the LCDK for which port to connect.

Obtaining the SDK

A copy of the Linux SDK is provided on the microSD card that comes with the LCDK. It is stored in the 3rd partition. To copy the file to your Linux host machine, put the microSD card into the included SD-card adapter. Insert the card via a USB to SD card adapter or SD-card slot of your Linux machine.

You can also download the SDK from the LCDK product folder listed in your LCDK Quick Start Guide booklet included with the LCDK.
The file name should read something like ti-sdk-omapl138-lcdk-xx.xx.xx.bz2 where xx.xx.xx is the version number. For example a version of the SDK install file is "ti-sdk-omapl138-lcdk-01.00.00.bz2".

NOTE: Removing the microSD card which is inserted in the LCDK while Linux is running may damage the content of the filesystem since there could be pending file I/O operations. The cleanest way to do this before removing power to the LCDK is by executing a “shutdown” command at the serial terminal window as in the following line.

```
target $ shutdown –h now
```

A series of messages will be outputted on the screen. Wait for this activity to cease before removing power to the LCDK. The microSD card can now be removed safely from the board.

Linux SDK Installation Steps

The SDK can be installed to any directory in your home folder. For example if you downloaded the SDK to your "Downloads" folder, the following command will uncompress the SDK in the current directory:

```
host $ tar jxf ~/Downloads/ti-sdk-omapl138-lcdk-xx.xx.xx.bz2
```

A folder ti-sdk-omapl138-lcdk-xx.xx.xx (where xx.xx.xx will be substituted with the actual version number) should be created in the current directory as the result.

Following is an example of what the top-level folders and files should be found with the SDK.

ti-sdk-omapl138-lcdk-01.00.00
├── bin
├── board-support
├── docs
├── dsp-tools
├── example-applications
├── filesystem
├── host-tools
├── linux-devkit
├── Rules.make
└── setup.sh

More details for these folders are available with the LCDK Linux SDK Software Developer's Guide [1]. The next step is to set up the Linux build tools and verify its correct operation.

Setup and Verify Linux Tool Chain

Set up paths

To set up the tool chain paths, change to the just installed SDK directory and use the following procedure.

1) Export SDK Installation Path

The setup scripts provided with the SDK relies on the environment variable TI_SDK_PATH to know where you have installed the SDK. To make this known, use the “export” command on the host machine.

For example if you install the SDK under SDKs folder, the export command will look like

```
host $ export TI_SDK_PATH=/home/user/SDKs/ti-sdk-omapl138-lcdk-01.00.00
```

You can make this step automatic by adding this export command to your shell setup file. For bash shell, edit the file .bashrc to include this export line.

- Edit Rules.make (optional)

This file contains the paths that will be used by the “setup” script that you may need later so it is a good time to update your SDK path to be consistent.
Using the same example, the updated TI_SDK_PATH line in “Rules.make” file will be:

```
# Points to the root of the TI SDK
export TI_SDK_PATH=/home/user/SDKs/ti-sdk-omapl138-lcdk-01.00.00
```

2) Run the environment set up script:

A script called "environment-setup" is included to help automate the setting of other needed environment variables for development. To start it, run the following command:

```
host $ source linux-devkit/environment-setup
```

The script will change the shell prompt to start with “linux-devkit” followed by the current directory. Using the same example, after execution the prompt should change to

```
[linux-devkit]:~/SDKs/ti-sdk-omapl138-lcdk-01.00.00>
```

3) Verify that your path has been updated to include the GCC tool chain provided by the SDK using the command:

```
host $ echo $PATH
```

The listed path should include the <sdk-install-path>/linux-devkit/bin of the just installed SDK.

You can also test the compiler by checking the version using

```
host $ arm-arago-linux-gnueabi-gcc --version
```

which should return something like the following:

```
arm-arago-linux-gnueabi-gcc (GCC) 4.5.3 20110311 (prerelease)
Copyright (C) 2010 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Edit and compile "Hello World!"

Now that the Linux tool chain is in place, you can quickly whip up the “Hello World!” program to perform a confidence build. Cut and paste the following program to your favorite Linux text editor (for Ubuntu the default editor is gedit) and save the file as hello.c

```
#include <stdio.h>
int main(void)
{
    printf ("Hello World! LCDK\n");
    return 0;
}
```

Cross-compile the program:

```
host $ arm-arago-linux-gnueabi-gcc hello.c –o hello
```

Booting Linux on LCDK using Pre-installed Binaries

Your LCDK should come with the boot-loader u-boot pre-installed. The provided microSD card contains a default Linux kernel and an ARMv5 file system designed for the LCDK. Setup the LCDK and boot up Linux as described in the “Quick Start Guide” booklet included with the LCDK.

Note: the procedure below involves switching back and forth between the host linux shell (indicated by host $) and the LCDK linux shell (indicated by target $) in the serial terminal window.

Transfer and Run "Hello World!" application to LCDK

At the serial terminal window, enter the following command to find out the IP address of the board:

```
target $ ifconfig
```

Once the IP address is known, use SCP to transfer the file to the LCDK. For example the following command entered on your Linux host shell will drop the “hello” executable to the “/home/root” folder on the LCDK (make sure you substitute <LCDK IP address> with the actual one):

```
host $ scp hello root@<LCDK IP address>:/home/root/
```

To verify that the file has been transferred OK, on the serial terminal window, list the /home/root directory content to check:

```
target $ cd /home/root
target $ ls
```

The output should look like the following:

```
root@omapl138-lcdk:~# ls
hello
```

Now run the application by executing it in the serial terminal window:

```
target $ ./hello
```

Note that in actual the interaction and result should look more like this:

```
root@omapl138-lcdk:~# ./hello
Hello World! LCDK
root@omapl138-lcdk:~#
```

Congratulations!

At this point, you have successfully
- Installed the LCDK Linux SDK
- Set up and verified the embedded Linux development Cross-Compile tool chain
- Created, downloaded and ran a Linux application on the OMAP-L138 LCDK board

You now have a running embedded Linux system based on the OMAP-L138 SoC and can develop applications for it. You can continue using this method to develop Linux applications to run on the LCDK, or explore more advanced procedure such as using NFS to mount the file system, customizing the Linux kernel, and writing applications that can communicate with signal processing programs running on C674 DSP core using the link listed in the “Next Steps” section.

Next Steps

More advanced procedures are listed in the LCDK Linux SDK Software Developers Guide [1]

Resource

OMAP-L138/ C6748 Product Information

TI OMAP-L138 SoC product folder [2]
TI OMAP-L138/ C6748 Product Introduction Wiki link [3]

Linux Host Support

Ubuntu Linux Host Configuration steps [Http://processors.wiki.ti.com/index.php/Linux Host Configuration - Ubuntu How-to]

Virtual PC for Ubuntu Linux using VirtualBox [4]
Virtual PC for Ubuntu Linux using VMware [5]

Serial Terminal

TeraTerm [6] for Windows ™ OS
Minicom [7] help page for Linux

References

[1] http://processors.wiki.ti.com/index.php/OMAP-L138_LCDK_Linux_Software_Developer%27s_Guide
[2] http://www.ti.com/product/OMAP-L138
[3] http://processors.wiki.ti.com/index.php/C674x/OMAPL1x_Introductory_Information
[4] http://processors.wiki.ti.com/index.php/How_to_Build_a_Ubuntu_Linux_host_under_VirtualBox
[5] http://processors.wiki.ti.com/index.php/How_to_Build_a_Ubuntu_Linux_host_under_VMware
[6] http://www.ayera.com/teraterm/
[7] https://help.ubuntu.com/community/Minicom

Article Sources and Contributors

OMAP-L138 LCDK Linux SDK Getting Started Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=107222 *Contributors*: Loc

Image Sources, Licenses and Contributors

Image:Evm environ.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Evm_environ.png *License*: unknown *Contributors*: Kevinsc

License

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.