DSP/BIOS™ Link

Installation Guide

1.65.00.03

Published on 12 th JUL 2010

Platform Support Products

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address:

Texas Instruments,
Post Office Box 655303,
Dallas, Texas 75265

Table of Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes how to install and run samples for OMAPL138 DSP/BIOS™ Link.

How to Use This Manual

This document includes the following chapters:

- Chapter 1, *Install Guide for OMAPL138* - describes the steps required to install and run samples for OMAPL138.

Please go through the Release Notes document available in the release package before starting the installation.

Notation of information elements

The document may contain these additional elements:

Warning

This is an example of warning message. It usually indicates a non-recoverable change.

Caution

This is an example of caution message.

Important

This is an example of important message.

Note

This is an example of additional note. This usually indicates additional information in the current context.

> **Tip**
>
> This is an example of a useful tip.

If You Need Assistance

For any assistance, please send an mail to software support.

Trademarks

DSP/BIOS™ is a trademark of Texas Instruments Incorporated.

All other trademarks are the property of the respective owner.

Install Guide for OMAPL138

Abstract

This chapter describes how to install and run samples for OMAPL138 from the new DSP/BIOS™ Link.

Table of Contents

1.1. Introduction

1.1.1. Purpose and Scope

DSP/BIOS™ LINK is foundation software for the inter-processor communication across the GPP-DSP boundary. It provides a generic API that abstracts the characteristics of the physical link connecting GPP and DSP from the applications. It eliminates the need for customers to develop such link from scratch and allows them to focus more on application development.

This document provides the users necessary information to install DSP/BIOS™ LINK on the development host.

This document corresponds to the product release Version 1.65.00.03 dated JUL 12, 2010.

1.1.2. Terms and Abbreviations

CCS	Code Composer Studio
IPC	Inter Processor Communication
GPP	General Purpose e.g. ARM
DSP	Digital Signal Processor e.g. DM648
DSPLink	A generic term used for DSP/BIOS™ Link. It appears in italics in all usages
CGTools	Code Gen Tools, e.g. Compiler, Linker, Archiver

Table 1.1. Terms and Abbreviations

1.1.3. References

1	Documentation included with the OMAPL138 hardware.
2	Release notes and User Guide of DaVinci-PSP-SDK-03.20.00.11.tgz

Table 1.2. References

1.2. INSTALLATION

1.2.1. Basic Installation

The DSP/BIOS™ LINK is made available as a tar.gz file. To install the product follow the steps below:

• Unzip and untar the file dsplink_linux_1.65.00.03.tar.gz

> **Note**
>
> This document assumes the install path to be in the user home directory if working on a Linux PC. This path will be used in remainder of this document

> **Note**
>
> This document assumes the install path to be L:\dsplink if working on a Windows PC. This path will be used in remainder of this document.

> **Note**
>
> If the installation was done at different location, make appropriate changes to the commands listed in the document.

It is advisable to archive the released sources in a configuration management system. This will help in merging:

• The updates delivered in the newer releases of DSP/BIOS™ LINK.

• The changes to the product, if any, done by the users.

1.2.1.1. Installing Standalone DSP/BIOS™and CGTools

For compilation of DSP-side sources and applications, the CGTools version 6.1.2 can be used. This release has been validated with DSP/BIOS™ version 5_33_05.

The standalone DSP/BIOS™ and standalone CGTools are available for Linux platform as well. Refer to the URL mentioned below for getting the distribution of DSP/BIOS™ and the associated installation instructions
DSP/BIOS™ and XDC

The directory structure specified in Figure 1 is expected by the build system of DSP/BIOS™ LINK. If you install the tools to a different directory, you will also need to modify the make system and the scripts contained in the release package. You may need to copy the directories to create the structure expected for compiling sources. Refer to section on "Understanding The MAKE System" in the User Guide for details.

1.2.1.2. Installing GNU make 3.81

For compilation of DSPLINK sources the GNU make 3.81 can be used. Download the make 3.81 from the URL http://ftp.gnu.org/pub/gnu/make/make-3.81.tar.gz

The following are the installation steps required to install make on the development host machine.

- Cd to make-3.81 directory

- Copy and untar make-3.81.tar.gz to your home directory.

- Type './configure' and press enter to configure the package for your system. Running `configure' takes awhile. By default, make package's files will be installed in `/usr/local/bin', `/usr/local/man', etc. You can specify an installation prefix other than `/usr/local' by giving `configure' the option `-- prefix=PREFIX'. For example, To install make at /usr/local/bin run the configure command like below.

    ```
    . /configure --prefix=/usr/local.
    ```

    To install make at /usr/bin run the configure command like below.

    ```
    ./configure --prefix=/usr
    ```

- Type `make' and press enter to compile the package.

- Optionally, type `./make check' and press enter to run any self-tests that come with the package.

- Type `make install' and press enter to install the programs and any data files and documentation.

- For additional details refer to INSTALL file located under make-3.81 directory

1.2.2. Creating development workspace

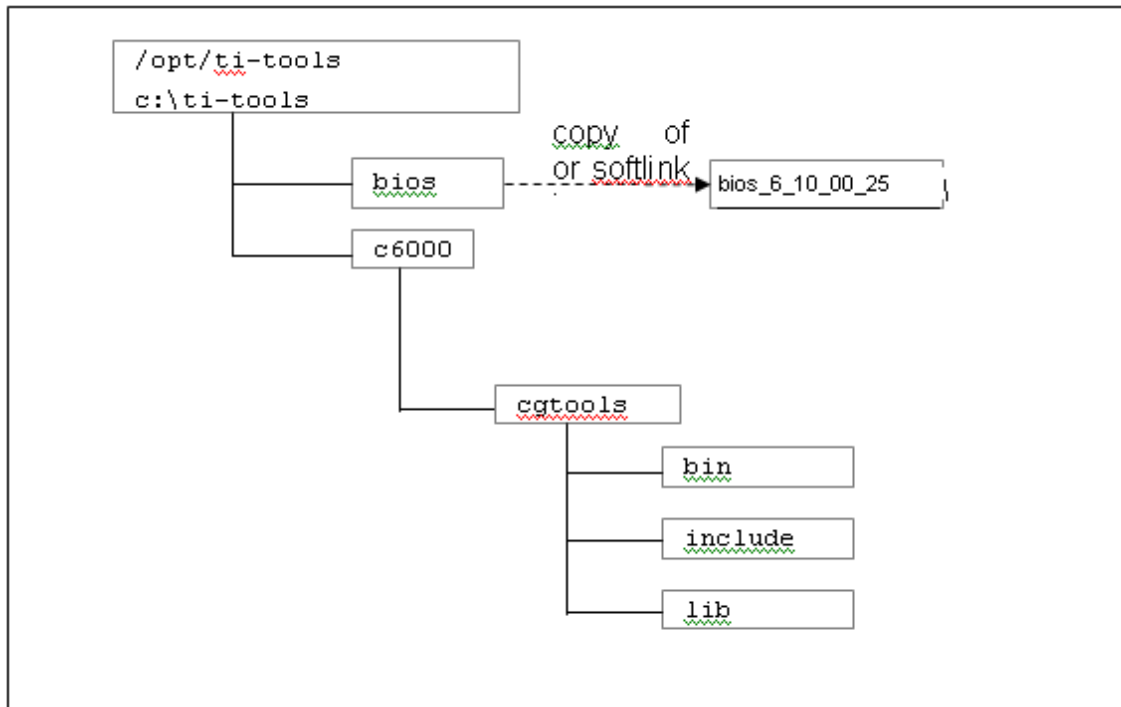This document and the scripts included in the release assume the following directory on your development host:

Figure 1.1. Development Workspace

> **Note**
>
> For Linux, the build system shipped with DSP/BIOS™ LINK assumes that the standalone DSP/BIOS™ is installed in the /opt/'ti-tools/' directory on the <ROOT-DRIVE> and CGTools and CSL are installed in the 'ti-tools/c6000' directory on the <ROOT-DRIVE>.

> **Note**
>
> For the Windows development host, the build system shipped with DSP/BIOS™ LINK assumes that the standalone DSP/BIOS™ is installed in the 'ti-tools\' directory on the ROOT-DRIVE and CGTools and CSL are installed in the 'ti-tools\c6000\' directory on the <ROOT-DRIVE>.

> **Note**
>
> To support multiple installations of DSP/BIOS with a single DSP/BIOS™ LINK DSP-side distribution file, a standard /opt/ti-tools/bios on Linux and c:\ti-tools\bios directory is used for the BIOS installation. This can be a soft link or copy to the actual DSP/BIOS installation directory

1.3. Setting up Linux Workstation

The description in this section is based on the following assumptions:

- The workstation is running on Red hat Fedora Core 7 (Kernel ver 2.6.21). Fedora 7

- Services telnetd, nfsd, ftpd are configured on this workstation.

> **Note**
>
> The release package has been tested on Red hat Fedora Core 7 (Kernel ver 2.6.21) You may be able to build on a higher version depending on the compatibility of the build tools in your version with the tested version.

> **Note**
>
> Ensure that kernel sources for Linux have also been installed on the workstation.

1.3.1. Creating target system

This release of DSP/BIOS™ LINK for OMAPL138 has validated on Linux kernel versions 2.6.18. It is recommended to use this version to use DSP/BIOS™ Link.

1.3.1. Setting up to build the kernel

This step is not required if a ARM tool chain already available on your setup. This tool-chain required for building the LSP and DSPLink.

- The tool chain arm 2009q1-203 can be downloaded from http://www.codesourcery.com/sgpp/lite/arm/portal/subscription3057 .Un tar the tool chain and add the tool chain directory in your path.

    - `$ tar –xjvf arm-2009q3-203 –arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.tar`

    - `$ export PATH=$PATH:${HOME}/arm-2009q1-203/bin`

1.3.2. Setting up the Linux PSP for OMAPL138

- Configure the kernel for the OMAPL138 platform. Further, DHCP should be enabled. Also ensure that you disable the CONFIG_PREEMPT option if it is already selected and ensure that EABI option is selected. DSP/BIOS Link supports the following pre-emption mode

> **Important**
>
> DSP/BIOS Link does not support Complete Preemption (Real-Time)

Steps to build the kernel:

Configure Linux for OMAPL138

```
$make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi-
 da850_omapl138_defconfig
```

Create uImage for OMAPL138

```
$make ARCH=arm CROSS_COMPILE=arm-none-linux-gnueabi- uImage
```

The image for compressed kernel uImage is generated after successful completion of the build process in the arch/arm/boot directory.

1.3.3. Enable TFTP for downloading the kernel image to target

U-boot can be configured to download the kernel onto the target by various mechanisms:

- TFTP

- Serial Port

This section configures the Linux development host as a TFTP server. Modify the xinet.d/ tftp file to enable TFTP:

- Make the following changes:


```
disable     = no
server_args = -s /tftpboot
```


- Restart the network service


```
$ /etc/init.d/xinetd restart
```


The above configuration assumes that a directory tftpboot has been created at the root directory. The files in this directory are exposed through the TFTP protocol.

1.3.4. Create target file system and export through NFS

The target device needs a file system to boot from. The file system can be exported to the target through NFS.

- A directory on the development host can be setup and exported for this purpose.


```
PSP release contains the NFS file system (nfs.tar.bz).
```

- `tar xjvf nfs.tar.bz`

> **Important**
>
> Enter the command shown above in a single line. You need to be root to successfully execute this command. The file system can be copied to a different location. In such a case ~/OMAPL138/target can be a soft link to the actual location.

- The directory ~/OMAPL138/target will be mounted as root directory on the target through NFS. To do so, add the following line to the file /etc/exports.

```
/home/user/OMAPL138/target  *(rw,no_root_squash)
```

Replace the directory in the path above with the actual path of your home directory on the development workstation.

1.4. Configuring CCS

1.4.1. OMAPL138 EVM

To use CCS for debugging the DSP side application, you will need to configure CCS to use both ARM and DSP with the OMAPL138 EVM. The EVM must be configured to use the DSP self-boot mode.

> **Note**
>
> CCS can attach to only ARM in the beginning. It can attach to the DSP only after the ARM-side application releases it from reset through a call to PROC_Start ().

1.5. WORKING ON TARGET PLATFORM

1.5.1. OMAPL138 EVM

1.5.1.1. Setting up Linux environment

Refer to the PSP Linux Release/User guide document for instructions on setting up the Linux execution environment for the OMAPL138 EVM.

```
DaVinci-PSP-SDK-03.20.00.11
```

1.5.1.2. U-Boot boot-loader

Please refer the PSP Linux release/install to load the u-boot.bin to the target. DaVinci-PSP-SDK-03.20.00.01.tgz contains images for the Boot Loader, NFS File system.

```
/DaVinci-PSP-SDK-03.20.00.11/images/u-boot/u-boot.bin
```

```
/DaVinci-PSP-SDK-03.20.00.11/images/fs/nfs.tar.gz
```

1.5.1.3. Configuring Kernel Parameters

DSP/BIOS™ LINK requires a few specific arguments to be passed to the Linux kernel during boot up. 2MB of memory is used by DSP/BIOS™ LINK for communication between GPP and DSP, and for DSP external memory. This must be reserved by specifying 2MB less as available for the Linux kernel for its usage.

1.5.1.4. Configure the DSP/BIOS LINK

The build configuration command must be executed to configure DSPLink for the various parameters such as platform, GPP OS, build configuration etc.

```
perl dsplinkcfg.pl --platform=OMAPL138 --nodsp=1 --dspcfg_0=OMAPL138GEMSHMEM --
dspos_0=DSPBIOS6XX --gppos=ARM --comps=ponslrmc
```

```
perl dsplinkcfg.pl --platform=OMAPL138 --nodsp=1 --dspcfg_0=OMAPL138GEMSHMEM --
dspos_0=DSPBIOS6XX --gppos=OMAPL138LSPuc --comps=ponslrmc
```

> **!** **Important**
>
> Enter the commands shown above in single line

> **!** **Important**
>
> For details please refer user guide.

> **Important**
>
> To build the GPP side update the Kernel (KERNEL_DIR) and toolchain (TOOL_PATH) path in dsplink/gpp/src/Rules.mk file.

1.5.2. Running the sample applications

Eight sample applications are provided with DSPLINK for the OMAPL138 platform. All the sample applications are described in detail in the user guide. This section describes the way to execute the sample applications

The specific instructions shown below refer to the loop sample. However, similar instructions can be used for the other applications also.

1.5.2.1. Copying files to target file system

The generated binaries on the GPP side and DSP side and the data files must be copied to the target directory. The commands below demonstrate this for the 'loop' sample application as reference. Appropriate sample directory name must be used for other sample applications.

1.5.2.1.1. GPP Side

For executing the DEBUG build, follow the steps below to copy the relevant binaries:

```
$ cd ~/dsplink

$ cp gpp/export/BIN/Linux/OMAPL138/DEBUG/loopgpp  /opt/dsplink/samples/loop

$ cp gpp/export/BIN/Linux/OMAPL138/DEBUG/dsplinkk.* /opt/dsplink/
```

For executing the RELEASE build, follow the steps below to copy the relevant binaries:

```
$ cd ~/dsplink

$ cp gpp/export/BIN/Linux/OMAPL138/RELEASE/loopgpp  /opt/dsplink/samples/loop

$ cp gpp/export/BIN/Linux/OMAPL138/RELEASE/dsplinkk.* /opt/dsplink/
```

> **Important**
>
> Enter the commands shown above in single line

1.5.2.1.2. DSP Side

The DSP binaries can be built either on the Linux workstation or the Windows host.

After the binaries have been built, they must be copied into the target file system. If the binaries are generated on Windows PC, any FTP client can be used for transferring these to the target file system.

For executing the DEBUG build, follow the steps below to copy the relevant binaries:

```
$ cd ~/dsplink
```

```
$ cp dsp/export/BIN/DspBios/OMAPL138/OMAPL138GEM_x/DEBUG/loop.out  /opt/dsplink/
samples/loop
```

For executing the RELEASE build, follow the steps below to copy the relevant binaries:

```
$ cd ~/dsplink
```

```
$ cp dsp/export/BIN/DspBios/OMAPL138/OMAPL138GEM_x/RELEASE/loop.out  /opt/
dsplink/samples/loop
```

> **!** **Important**
>
> Enter the commands shown above in single line

1.5.2.2. Loading the kernel module: dsplinkk.ko

To load the device driver, login as 'root' and enter following commands on the command prompt.

```
$ cd /opt/dsplink
```

```
$ mknod /dev/dsplink c 230 0
```

```
$ insmod dsplinkk.o
```

This action generates a warning indicating that the kernel module does not contain the GPL license. This warning can be safely ignored.

1.5.2.3. Invoking the application

1.5.2.3.1. Loop sample

To invoke the application enter the following commands:

```
$ cd /opt/dsplink/samples/loop
```

```
$ ./loopgpp loop.out <buffersize> <iterations>  <processor identifier>
```

> **!** **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

e.g.

```
$ ./loopgpp loop.out 1024 10000
```

1.5.2.3.2. Message sample

```
$ cd /opt/dsplink/samples/message
```

```
$ ./messagegpp message.out <number of iterations>  <processor identifier>
```

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

---

e.g.

```
$ ./messagegpp message.out
```

1.5.2.3.3. Scale sample

```
$ cd /opt/dsplink/samples/scale
```

```
$ ./scalegpp scale.out <buffer size> <iterations>   <processor identifier>
```

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

---

e.g.

```
$ ./scalegpp scale.out 1024 10000
```

1.5.2.3.4. Ring_IO sample

```
$ cd /opt/dsplink/samples/ring_io
```

```
$ ./ringiogpp ringio.out <RingIO data buffer size in bytes> <number of Bytes to
 transfer>    <processor identifier>
```

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

---

e.g.

```
$ ./ringiogpp ringio.out 1024 10240
```

> **Important**
>
> By default Ring_IO sample runs in multithread mode. To run the sample in multi process mode ,define RINGIO_MULTI_PROCESS flag in $DSPLINK\gpp\src\samples\ring_io \Linux\COMPONENT file and build the sample

---

1.5.2.3.5. Readwrite sample

```
$ cd /opt/dsplink/samples/readwrite
```

```
$ ./readwritegpp readwrite.out <DSP address> <buffer size> <iterations>
 <processor identifier>
```

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

e.g.

```
$ ./readwritegpp readwrite.out 0xC3F5B000 1024 1000
```

1.5.2.3.6. MPCSXFER sample

```
$ cd /opt/dsplink/samples/mpcsxfer
```

```
$ ./mpcsxfergpp mpcsxfer.out <buffer size> <iterations>    <processor
 identifier>
```

> **Note**
>
> The sample can be executed for infinite iterations by specifying the number of iterations as 0.

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

e.g.

```
$ ./mpcsxfergpp mpcsxfer.out 1024 10000
```

1.5.2.3.7. MP_LIST sample

```
$ cd /opt/dsplink/samples/mp_list
```

```
$ ./mplistgpp mplist.out <iterations> <number of elements>    <processor
 identifier>
```

> **Note**
>
> The sample can be executed for infinite iterations by specifying the number of iterations as 0.

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

e.g.

```
$ ./mplistgpp mplist.out 1000 20
```

1.5.2.3.8. MESSAGE_MULTI sample

```
$ cd /opt/dsplink/samples/message_multi
```

```
$ ./messagemultigpp messagemulti.out <number of transfers> <Application instance
 number 1 -> MAX_APPS>    <processor identifier>
```

> **Note**
>
> Argument processor identifier is optional, if it not provided assumed as default processor (zero).

e.g.

```
$ for i in 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16;
```

```
do ./messagemultigpp messagemulti.out 10000 $i    & done
```

1.5.2.4. Unloading the kernel module: dsplinkk.ko

To unload the device driver, enter following commands on the command prompt.

```
$ cd /opt/dsplink
```

```
$ rmmod dsplinkk
```

```
$ rm /dev/dsplink
```

None.