MCSDK HPC 3.x Getting Started Guide

From Texas Instruments Wiki

HPC (High Performance Computing) Development Tools for MCSDK

Version 3.0

Getting Started Guide

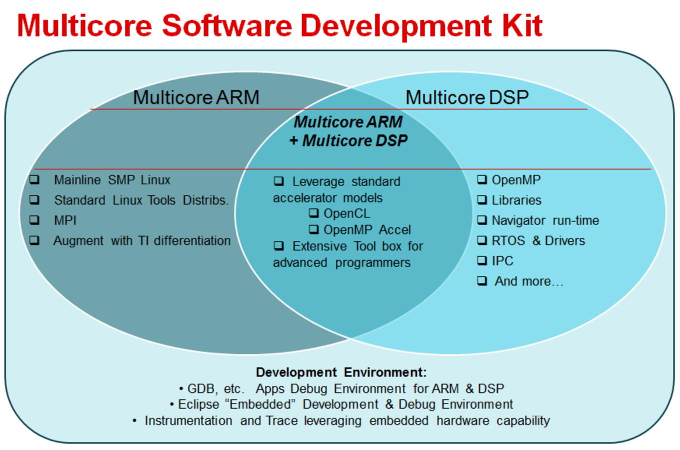
Last updated: 10/17/2014

Contents

Introduction

(Note: This page has the instructions for the MCSDK HPC 3.0.1.4 release. If you are using a release older than 03.00.01.04, Please check specific instructions in the Archived Pages section below.)

The Multicore Software Development Kit (MCSDK) provides foundational software for TI KeyStone II platforms, by encapsulating a collection of software elements and tools for both the ARM A15 and the C66x DSP. MCSDK-HPC (High Performance Computing), built as an add-on on top of the foundational MCSDK, provides HPC specific software modules and algorithm libraries along with several out of box sample applications. As highlighted in the picture below, SDKs together provides complete development environment [A15 + DSP] to offload HPC applications to TI C66x multi-core DSPs.


Multicore Software Development Kit
Multicore ARM
Multicore DSP
Multicore ARM + Multicore DSP
❑ Mainline SMP Linux
❑ Standard Linux Tools Distribs.
❑ MPI
❑ Augment with TI differentiation
❑ Leverage standard accelerator models
❑ OpenCL
❑ OpenMP Accel
❑ Extensive Tool box for advanced programmers
❑ OpenMP
❑ Libraries
❑ Navigator run-time
❑ RTOS & Drivers
❑ IPC
❑ And more…
Development Environment:
• GDB, etc. Apps Debug Environment for ARM & DSP
• Eclipse "Embedded" Development & Debug Environment
• Instrumentation and Trace leveraging embedded hardware capability

Listed below are the key components provided by MCSDK-HPC and a brief description about them:

Category	**Details**
OpenCL	OpenCL (Open Computing Language) is a multi-vendor open standard for general-purpose parallel programming of heterogeneous systems that include CPUs, DSPs and other processors. OpenCL is used to dispatch tasks from A15 to DSP cores
OpenMP on DSP	OpenMP is the de facto industry standard for shared memory parallel programming. Use OpenMP to achieve parallelism across DSP cores.
OpenMPI	Run on A15 cluster and use OpenMPI to allow multiple K2H nodes to communicate and collaborate.

Specifically, this Getting Started Guide for MCSDK-HPC provides information needed for running out of box MCSDK-HPC sample applications, recompiling MCSDK-HPC, and developing customer's HPC application leveraging MCSDK-HPC. By the end of this Getting Started Guide the user should have:

- Installed pre-requisite software for MCSDK-HPC
- Installed MCSDK-HPC along with the pre-built sample applications
- Run the out-of-box MCSDK-HPC sample applications on TI KeyStone II devices
- Recompiled MCSDK-HPC if needed
- Obtained instructions on how to develop customer's HPC application leveraging MCSDK-HPC

Acronyms

The following acronyms are used throughout this wiki page.

Acronym	**Meaning**
BLAS	Basic Linear Algebra Software
BOC	Breakout Cards
CCS	Texas Instruments Code Composer Studio
DSP	Digital Signal Processor
EVM	Evaluation Module, hardware platform containing the Texas Instruments DSP
FFT	Fast Fourier Transform
HPC	High Performance Computing
IPC	Inter-Processor Communication
MCSDK	Texas Instruments Multi-Core Software Development Kit
OpenCL	Open Computing Language
OpenMP	Open Multi-Processing
OpenMPI	Open Source Message Passing Interface
TI	Texas Instruments
.ipk	Debian-style software installation package used by the opkg package management system.

Supported Devices/Platforms

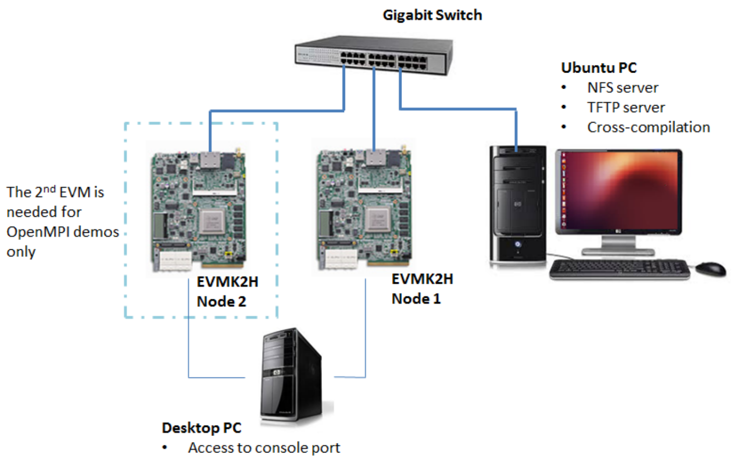
This release supports the following Texas Instruments devices/platforms:

Platform	**Supported Devices**	**Supported EVM**
[K2H (http://processors.wiki.ti.com/index.php/Keystone_II_Device_Architecture)]	66AK2H12 (http://www.ti.com/product/66ak2h12)	EVMK2H (http://www.ti.com/tool/evmk2h)

Hardware Requirements

As shown in the picture below, running HPC demos requires the below physical components

Physical Component	**Details**
K2H EVM (http://www.ti.com/tool/evmk2h)	One EVM is needed to run OpenCL, OpenMP sample applications. Two EVMs are needed for demonstrating OpenMPI cluster.
Gigabit Ethernet Switch	To interconnect the components (EVMs with NFS, TFTP) and to demonstrate OpenMPI over Ethernet.
PC running Ubuntu 12.04 LTS with NFS, TFTP	Ubuntu 12.04 LTS is available for free at Ubuntu 12.04 LTS (http://releases.ubuntu.com/precise/). The Ubuntu PC is also used as the TFTP server and NFS server.
PC for serial port connection (Can re-use Ubuntu PC)	To connect to K2H EVM Serial Port
Hyperlink Break Out Card (Optional)	In order to run OpenMPI-Hyperlink demos between two K2H EVMs, RTK breakout cards (BOC) (http://www.mistralsolutions.com/pes-products/development-platforms/rtm-break-out-card.html) and a Hyperlink cable (https://estore.ti.com/HL5CABLE-Hyperlink-Cable-P2888.aspx) are needed. For each K2H EVM, connect K2H EVM's Zone 3 ADF CONN (30 pin) into RTM breakout card's Micro TCA 4 Plug. Then use the Hyperlink cable to connect the breakout cards' Hyperlink IPASS HD connectors.
8 GB DIMM (Optional)	The EVMs come with 2GB DIMM. If the application needs larger memory (shared memory between A15 and C66x), users can upgrade to an 8 GB DIMM. We support all JEDEC-compliant DDR3 Unbuffered SODIMMs up to 8GB, both single and double rank. We do not support registered SODIMMs. The 8GB memory we used in our testing is MT18KSF1G72HZ-1G6E2ZE. Timing and configuration parameters will need to be adjusted for the specific SODIMM chosen



Software Installation

The table below provides an overview on all software required to run the MCSDK HPC examples.

Software	**Download URL**	**Download File**	**Description**
MCSDK	Please visit MCSDK HPC download page (http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk_hpc/latest/index_FDS.html) to look for the download link of	*mcsdk_native<ver>_native_setuplinux32.bin*	Texas Instruments MultiCore Software Development

	MCSDK to be used with the HPC release.		Kit
MCSDK HPC	MCSDK HPC download page (http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk_hpc/latest/index_FDS.html)	*mcsdk-hpc_<ver>_setuplinux.bin*	Texas Instruments MultiCore Software Development Kit for High Performance Computing Applications
Linaro Toolchain	Linaro Toolchain Download (https://launchpad.net/linaro-toolchain-binaries/trunk/2013.03/+download/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux.tar.bz2)	*gcc-linaro-arm-linux-gnueabihf-<ver>_linux.tar.bz2*	x86 Cross compiler for ARM code.
C6000 Code Gen (x86)	TI CGTools 7.6.x Download (https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm)	*ti_cgt_c6000_<ver>_linux_installer_x86.bin*	C6000 DSP compiler for x86
C6000 Code Gen (ARM)	TI CGTools 7.6.x Download (https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm)	*ti_cgt_c6000_<ver>_armlinuxa8hf_installer.sh*	C6000 DSP compiler for ARM

Software Installation on Ubuntu Desktop

Follow the instructions below to install MCSDK, MCSDK-HPC packages on the Ubuntu PC. Also install TFTP server and NFS server on the same PC.

Install MCSDK

Please change the attribute of the installer to executable and run the installer as shown below.

Please note if your Ubuntu OS is 64 bit and you have not run a 32 bit binary before, then run this command first:

```
> sudo apt-get install ia32-libs
```

```
> chmod +x mcsdk_native<version>_native_setuplinux.bin
> ./mcsdk_native<version>_native_setuplinux.bin
```

Install MCSDK-HPC

Download the HPC Linux installer (mcsdk-hpc_<version>_setuplinux.bin) to the Ubuntu PC, change the attribute of the installer to executable and then run the installer as shown below.

```
> chmod +x mcsdk-hpc_<version>_setuplinux.bin
> ./mcsdk-hpc_<version>_setuplinux.bin
```

Install TFTP server

First, install TFTP related packages:

```
$ sudo apt-get install xinetd tftpd tftp
```

Then, create /etc/xinetd.d/tftp and put the following entry in the file

```
service tftp
{
protocol        = udp
port            = 69
socket_type     = dgram
wait            = yes
user            = nobody
server          = /usr/sbin/in.tftpd
server_args     = /tftpboot
disable         = no
}
```

After that, create a folder /tftpboot in your root folder. This should match whatever you entered in server_args in /etc/xinetd.d/tftp. This folder is where a TFTP client will get files from. The TFTP client will not have access to any other folder. Then, Change the mode of the folder, and the ownership.

```
$ sudo mkdir /tftpboot
$ sudo chmod -R 777 /tftpboot
$ sudo chown -R nobody /tftpboot
```

Last, restart the xinetd service

```
$ sudo /etc/init.d/xinetd restart or
$ sudo service xinetd restart
```

Install NFS Server

First, Install NFS server related package:

```
$ sudo apt-get install nfs-kernel-server
```

Then, restart the nfs-kernel-server as follows:

```
$ sudo service nfs-kernel-server restart
```

If there is firewall, follow the steps below to install NFS server. Bypass these steps if there is no firewall [Expand]

EVM Setup

Basic Setup of the EVM

Follow instructions at EVMK2H Hardware Setup (http://processors.wiki.ti.com/index.php/EVMK2H_Hardware_Setup) to get the basic setup of the K2H EVM, including

1. FTDI Driver Installation on PC Host
2. BMC Version Check and Update (make sure BMC is up to date)
3. UCD Power Management Modules In-Field Update (make sure UCD is up to date)
4. Attach the Ethernet cable to ENET0 port
5. Set the boot mode switch SW1 as SPI Little Endian Boot mode (http://processors.wiki.ti.com/index.php/EVMK2H_Hardware_Setup#SPI_Little_Endian_Boot_mode))
6. Attach the serial port cable to the SoC UART port
7. Connect the power cable

Install Terminal Software on PC for Serial Port Connection

Install PuTTY on the PC for connecting to the serial port of the EVM.

If a Ubuntu Desktop is used for serial port connection, follow the steps below to establish the serial connection to the board.

1. Connect your Ubuntu PC to the EVM

- The connection between the PC and the EVM is between a USB port on the PC and the mini-USB on the EVM main board (not the daughter card). Do not power up the board yet.

2. Obtain EVM serial device.

- By typing *dmesg | grep tty* on a terminal on the host, it shows a list of the available serial connections. *ttyUSB** (FTDI USB) should be the serial ports for the EVM. The lower number is the SOC (System on Chip) UART (*/dev/ttyUSB0*), while the higher number is the BMC (Boot Monitor Controller) UART (*/dev/ttyUSB1*).

3. Install a serial terminal emulator.

- *picocom* or another terminal emulator (such as putty, minicom) must be installed if it is not already.

```
sudo apt-get install picocom
```

Then, establish a serial connection to the USB0 port.

```
picocom -b 115200 /dev/ttyUSB0
```

4. Power up the EVM.

- The terminal will start displaying the Uboot text. It should continuously display "BOOTP BROADCAST #". Press crtl+C to get out of this loop, and this will come to the # prompt, which is now in Uboot.

Bring up Linux on EVM using an NFS filesystem

Follow the instructions below to bring up Linux on EVMK2H using an NFS filesystem.

1. Program or Upgrade Uboot Image

- Before booting up Linux, program or update the U-Boot image to the latest version provided in the MCSDK release following instructions at MCSDK Programming SPI NOR flash with U-Boot GPH image (http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Tools#Programming_SPI_NOR_flash_with_U-Boot_GPH_image). The Uboot image "u-boot-spi-keystone-evm.gph" can be found from the directory of [mcsdk_install_dir]/mcsdk_linux_3_00_0x_xx/images.

2. Copy Required Linux Kernel images

- On the Ubuntu Desktop, copy the following images from [mcsdk_install_dir]/mcsdk_linux_3_00_0x_xx/images to the TFTP server directory

```
skern-keystone-evm.bin (This is the boot monitor)
uImage-k2hk-evm.dtb (This is the device tree file)
uImage-keystone-evm.bin (This is the Linux kernel)
```

Please note that if a device tree (k2hk-evm.dtb) is provided along with HPC installation in [mcsdk-hpc_install_dir]/mcsdk-hpc_3_00_0x_xx/images directoy, use this .dtb file instead.i.e

```
cp [mcsdk-hpc_install_dir]/mcsdk-hpc_3_00_0x_xx/images/k2hk-evm.dtb uImage-k2hk-evm.dtb
```

3. Unpack TI SDK Root File System

- On the Ubuntu Desktop, create a directory which will contain the TI SDK Linux root file system, e.g., /evmk2h_nfs. Within that directory unpack the tisdk-rootfs.tar.gz from [mcsdk_install_dir]/mcsdk_linux_3_00_0x_xx/images folder:

```
sudo tar xvf tisdk-rootfs.tar.gz -C /evmk2h_nfs (path to folder you just created)
```

Be careful of where you unpack this root file system. Make sure that you do not unpack this file system in your root folder on the Ubuntu Desktop.

4. Edit NFS configuration for the Ubuntu Desktop

- Edit the NFS configuration file /etc/exports to include the path to the network file system. Add the following line to the end of the file. Please replace '/evmk2h_nfs' in the line below with the directory path where you have unpacked the TI embedded device's filesystem:

```
/evmk2h_nfs *(rw,subtree_check,no_root_squash,no_all_squash,sync)
```

After this change, use the “sudo service nfs-kernel-server restart” to restart the NFS server.

5. Set the Environment variables in Uboot of the EVM

- Through Terminal, after stopping the autoboot process of the EVM, modify u-boot of target EVMs to set the environment variables.

```
env default -f -a
setenv boot net
setenv mem_reserve 1536M [A larger size can be used when using more than 2GB DIMM]
setenv gatewayip 128.247.102.1 [This is the gateway IP of the subnet on which the host PC and the board are present]
setenv serverip xxx.xxx.xxx.xxx [This is the IP of the host linux machine]
setenv tftp_root /tftpboot [Path to the tftp server on your host machine]
setenv name_fdt uImage-k2hk-evm.dtb [Optional as default definition matches it already]
setenv name_kern uImage-keystone-evm.bin [Optional as default definition matches it already]
setenv nfs_root /evmk2h_nfs [Path to the NFS on your host machine]
setenv nfs_serverip [ip address of nfs root, set this if the nfs_root is on a different computer than the tftp server]
saveenv (This saves the envirnoment variables to the flash)
```

6. Modify hostname environment variable in u-boot.

- This step is required for running the MPI examples as MPI requires that all participating hosts have a unique hostname. The instructions in this guide assume that one EVM is set with hostname "k2hnode1" and a second EVM is set with hostname k2hnode2., e.g.,

On EVM1:

```
setenv hostname k2hnode1
saveenv
```

On EVM2:

```
setenv hostname k2hnode2
saveenv
```

7. Power cycle the EVM

- Power cycle the board so that the new version of Uboot loads. You can stop the autoboot process to check your environment variables or the version of Uboot by tying version on the Uboot command prompt if you'd like. If you have stopped the auto boot process you can continue the booting process by typing in the boot into the command prompt.

```
boot
```

Set up non-root login

It is good practice to use a non-root account to protect the system while doing most work. The following steps provide instructions for creating a user account on the EVM that is tied to the user account on the x86 Ubuntu development PC.

1. On the x86 Ubuntu development PC, obtain the user account ID (UID).

- The UID of the account is kept in the UID environment variable. The value of this variable will be used to set up the user account on the EVM.

```
> echo $UID
```

2. Log in to the EVM as root user.

- NOTE: First-time login does not require a password.

3. Set root password

- This step is optional, though it is strongly recommended for security reasons.

```
> passwd
```

4. Add a non-root account.

- While the user account name can be any valid string, the rest of this document assumes that user name chosen is "mpiuser"

```
> useradd -s /bin/bash -u <UID> mpiuser
```

5. Set the user account password to "gguser502"

```
> passwd mpiuser
```

Please see the Troubleshooting section: ***Why must the account password be "gguser502"?*** for more information about passwords.

Establish two K2H EVM nodes for OpenMPI Demos

The steps below establish two K2H EVM nodes on a trust-worthy network so that they can communicate securely in OpenMPI applications. Bypass these steps for non-OpenMPI demos.

1. Log in as root user.

2. Find IP addresses of the K2H EVM nodes during bootup or use ifconfig to get the IP addresses after bootup. Then, edit /etc/hosts to include correct hostname and IP address.

```
127.0.0.1 localhost.localdomain localhost
[ip address of K2HEVM node1] k2hnode1
[ip address of K2HEVM node2] k2hnode2
```

3. Login as mpiuser.

4. Do SSH between the two nodes.

```
[k2hnode1] ssh mpiuser@k2hnode2, accept [w/ yes] then exit
[k2hnode2] ssh mpiuser@k2hnode1, accept [w/ yes] then exit
```

After this step, file ~/.ssh/known_hosts (/home/mpiuser/.ssh/known_hosts) is properly set with information about the other node.

Software Installation on File System of the EVM

After setting up the file system of the EVM, install the following packages on the file system of the EVM.

A reboot of the EVMs will be required once the installation steps have been completed.

Install HPC IPKs and CGTools on to the EVM

The HPC release includes pre-compiled binaries which are packaged in *.ipk (Debian-style software installation package format) and cgtools. They can be installed natively on the target EVMs with an installer script.

- Copy IPK files,Cgtools and the installer script to target filesystem

The IPK files & Cgtools are located under [mcsdk-hpc install dir]/mcsdk_hpc_<version>/images directory.

- Log on to the EVM as root user, and run the EVM installer script (install_hpc_packages_evm.sh) which installs the IPKs and cgtools on to the EVM file system.

i.e, on the EVM,

```
> mkdir temp
> cd temp
> cp [mcsdk-hpc install dir]/mcsdk_hpc_<version>/images/*.ipk .
> cp [mcsdk-hpc install dir]/mcsdk_hpc_<version>/images/*.sh .
> chmod +x install_hpc_packages_evm.sh
> ./install_hpc_packages_evm.sh
> cd ..
> rm -rf temp
```

NOTE: To install the HPC IPKs & Cgtools from the x86 Ubuntu machine which hosts the EVM filesystem, do the following

```
> cd [mcsdk-hpc install dir]/mcsdk_hpc_<version>/images
> chmod +x install_hpc_packages_x86.sh
> ./install_hpc_packages_x86.sh /evmk2h_nfs
```

In the above commands, "/evmk2h_nfs" is the root directory of the EVM filesystem.

Please reboot the EVMs so that they will be properly configured to build and run the sample applications.

Setting up Build Environment for Examples

The MCSDK HPC examples are installed in the EVM's filesystem at "/usr/share/ti/examples", and they are provided as source. Thus, in order to run the examples, they must be compiled. They can be either compiled natively on the EVM, or cross compiled on the x86 Ubuntu development PC.

The sections below provide steps for setting up the build environment for each case.

Install cross compilation tools

In order to rebuild and/or debug HPC binaries and demos, install the following software in addition on the Ubuntu Desktop.

1. Install libelf1:i386

- Install libelf1 32-bit libraries:

```
> sudo apt-get install libelf1:i386
```

2. Install Linaro Tool Chain

- Unzip Linaro tools:

```
> mkdir -pv ~/linaro
> tar -xjvf gcc-linaro-arm-linux-gnueabihf-<VER>_linux.tar.bz2 -C ~/linaro
```

Setup HPC environment

1. Edit installation paths

Modify variables in <mcsdk-hpc_install_dir>/scripts/setup_hpc_env.sh as needed to point to install location of MCSDK 3.0.x, MCSDK-HPC, TI CGTools, and etc on the Desktop Linux PC.

```
...
# Add search paths here. Multiple colon-seperated directories may be given to a
# single variable.
#
LINARO_INSTALL_PATH=~/linaro:~/ti/linaro
TI_CGT_INSTALL_PATH=~/ti:/opt/ti
MCSDK_INSTALL_PATH=~/ti
MCSDK_HPC_INSTALL_PATH=~/ti
...
```

By default, the target root directory is set as */evmk2h_nfs* in setup_hpc_env.sh. Modify it if a different target root directory is used.

```
TARGET_ROOTDIR=/evmk2h_nfs
```

2. Execute environment script

```
> source ./setup_hpc_env.sh
```

This will set environment variables to point to the required packages. A warning will be displayed for any package which could not be found. If this happens, please revisit the previous step to add all the necessary search paths.

Install the Linux DevKit

The Linux Devkit is required for both native and cross compilation. This is because the filesystem only contains the files necessary for runtime (executables and shared libraries), while the DevKit contains the files necessary for development (static libraries and header files). The Linux Devkit is provided by the base MCSDK can be found inside the mcsdk_linux package.

The cross compilation procedure makes an assumption as to the location of the DevKit once it is installed. This is in the same location as the Linux DevKit installer (<mcsdk_install_dir>/mcsdk_linux_3_00_xx_xx/linux-devkit). Because of this assumption, MCSDK HPC provides a script to install the devkit in this location. The script is located in <mcsdk-hpc_install_dir>/mcsdk-hpc_03_00_00_xx/scripts. The recommended procedure is as follows:

- NOTE: The follow step assumes that the environment has been setup by following the instructions in the previous section.

1. Execute "install_devkit.sh"

Once the environment has been set up, execute the "install_devkit.sh" script:

```
> cd <mcsdk-hpc_install_dir>/mcsdk-hpc_<VER>/scripts
> ./install_devkit.sh
```

Native Compilation Setup

Native compilation requires that the development files are available on the EVM filesystem. This is achieved by copying the Linux DevKit onto the EVM filesystem along with a few DSP components. A helper script is provided to verify that all files are installed correctly.

- NOTE: The follow step assumes that the environment has been setup by following the instructions in the previous section.

1. Copy DevKit to EVM filesystem

- NOTE: Root privileges are required to copy to EVM filesystem. The command 'sudo -E' achieves this along with retaining environment variables from the current shell session.

```
> cd <mcsdk-hpc_install_dir>/mcsdk-hpc_<VER>/scripts
> sudo -E ./install_devfs.sh
```

Compiling MCSDK HPC Examples

1. Copy examples to a user writable directory on the EVM filesystem.

On EVM:

- Log in as "mpiuser"

```
> mkdir -v /home/mpiuser/mcsdk-hpc-examples
> cp -rv /usr/share/ti/examples/* /home/mpiuser/mcsdk-hpc-examples
```

On x86 Development PC:

```
> mkdir -v /evmk2h_nfs/home/mpiuser/mcsdk-hpc-examples
> cp -rv /evmk2h_nfs/usr/share/ti/examples/* /evmk2h_nfs/home/mpiuser/mcsdk-hpc-examples
```

2. Run top-level Makefile

On EVM:

- Log in as "mpiuser"
- Please see Troubleshooting section if you notice any "clock skew" warnings.

```
> cd /home/mpiuser/mcsdk-hpc-examples
> make
```

On x86 Development PC:

```
> cd /evmk2h_nfs/home/mpiuser/mcsdk-hpc-examples
> make
```

Running Out of Box Sample Applications

MCSDK-HPC provides multiple categories of demo applications to demonstrate OpenCL, OpenMPI, OpenMP, OpenMP accelerator Model run times. Most examples will produce a single executable which can be run without arguments. Example executables with more complex usage (such as the OpenMPI examples) will contain a README with the usage.

Detailed descriptions of each example can be found at MCSDK HPC 3.x Examples (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Examples).

More Details on Runtime

Please refer to the links below for User Guide / API documentation for OpenCL, OpenMP, and OpenMPI runtimes.

- OpenCL (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_OpenCL)
- OpenMP (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_OpenMP)
- OpenMPI (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_OpenMPI)

Useful Resources and Links

Product Download and Updates

For product download and updates, please visit the links listed in the table below.

	Product Download Link
MCSDK HPC Download	http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk_hpc/latest/index_FDS.html
MCSDK Download	http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk/3_00_04_18/index_FDS.html

Technical Support

For technical discussions and issues, please visit the links listed in the table below.

	Forum/Wiki Link
MCSDK HPC forum	http://e2e.ti.com/support/applications/high-performance-computing/f/952.aspx
C66x Multicore forum	http://e2e.ti.com/support/dsp/c6000_multi-core_dsps/f/639.aspx
TI-RTOS forum	http://e2e.ti.com/support/embedded/f/355.aspx
Code Composer Studio forum	http://e2e.ti.com/support/development_tools/code_composer_studio/f/81/t/3131.aspx
TI C/C++ Compiler forum	http://e2e.ti.com/support/development_tools/compiler/f/343/t/34317.aspx
Embedded Processors wiki	http://processors.wiki.ti.com

Note: When asking for help in the forum you should tag your posts in the Subject with “MCSDK HPC”, the part number (e.g. “TCI6636K2H”) and additionally the component (e.g. “FFT”).

Troubleshooting

Listed below are some Frequently Asked Questions. Please click on the "Expand" link adjacent to any question to see the answer.

- ***Can the IPK's be installed from the X86 development machine?*** [Expand]
- ***Why must the account password be "gguser502"? I see it listed in the world-readable file '/etc/profile.d/mcsdk-hpc-profile.sh'. Is there a more secure method?*** [Expand]
- ***How do I remove the "clock skew" warnings when compiling natively?*** [Expand]

For more questions and answers, please visit MCSDK-HPC Trouble Shooting Wiki (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Trouble_shooting).

Archived Pages

- **MCSDK HPC Releases prior to 3.0.1.2** : [1] (http://processors.wiki.ti.com/index.php?title=MCSDK_HPC_3.x_Getting_Started_Guide&oldid=185144)

Retrieved from "http://processors.wiki.ti.com/index.php?title=MCSDK_HPC_3.x_Getting_Started_Guide&oldid=186706"

- This page was last modified on 17 October 2014, at 13:15.
- This page has been accessed 8,202 times.
-