

MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800



HPC (High Performance Computing) Development Tools for MCSDK

Version 3.0

Getting Started Guide

Last updated: 10/07/2015

Contents

Introduction

- Acronyms
- Supported Devices/Platforms

ProLiant m800 Server Cartridge Pre-requisites and Setup

- System ROM Firmware
- Ubuntu Kernel Version
- Set System Clock

MCSDK-HPC Installation

- Option 1: Prepare ProLiant m800 Server Node to Fetch TI Packages from Partner repository
- Option 2: Prepare ProLiant m800 Server Node to Fetch TI Packages from PPA
- Install MCSDK-HPC
- Reboot ProLiant m800 Server Node
- Verify Installation

Compiling MCSDK HPC Examples

Running Out of Box Sample Applications

More Details on Runtime

Running Automated Tests

Uninstalling MCSDK-HPC

- Establish Two K2H Nodes for OpenMPI Demos

Useful Resources and Links

- Product Download and Updates
- DSP Debug and Profiling Utilities
- Technical Support

Troubleshooting

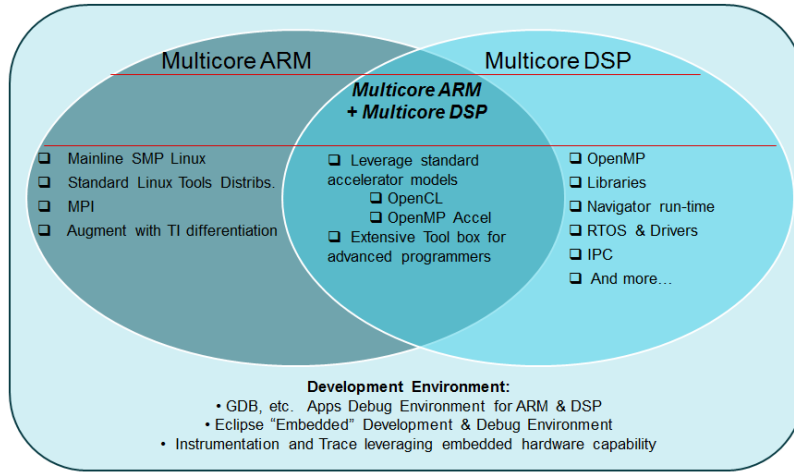
Archived Pages

Introduction

(Note: This page has the instructions for the MCSDK HPC 3.0.1.3 release and newer. If you are using a release older than 03.00.01.03. Please check specific instructions in the archived section below :[Archived Pages](http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Archived_Pages) (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Archived_Pages))

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



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 Parallelization on DSP	Use OpenMP to achieve parallelize task execution between multiple C66x cores using #pragmas
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
DSP	Digital Signal Processor
FFT	Fast Fourier Transform
HP	Hewlett-Packard
HPC	High Performance Computing
IPC	Inter-Processor Communication
m800	HP Moonshot server cartridge that has four TI 66AK2H12 (K2H) nodes and supports Ethernet, SRIO, and Hyperlink for OpenMPI communication between nodes.
MCSDK	Texas Instruments Multi-Core Software Development Kit
OpenCL	Open Computing Language
OpenMP	Open Multi-Processing
OpenMPI	Open Source Message Passing Interface
SRIO	Serial Rapid Input Output
TI	Texas Instruments

Supported Devices/Platforms

This release supports the following devices/platforms:

Platform	Supported Devices	Supported Hardware
[K2H (http://processors.wiki.ti.com/index.php/Keystone_II_Device_Architecture)]	66AK2H12 (http://www.ti.com/product/66ak2h12)	HP ProLiant m800 (http://www8.hp.com/us/en/products/proliant-servers/product-detail.html?oid=6532018)

ProLiant m800 Server Cartridge Pre-requisites and Setup

Firmware	Satellite Controller	10/01/2014
	ROM	10/04/3014
	Cartridge Data	10/31/2014
	Cartridge Mezz 1 Data	10/31/2014
	Cartridge CPLD	09
Ubuntu kernel	3.13.0-24-keystone	#38-Ubuntu

System ROM Firmware

The System ROM firmware version may be queried from the chassis manager with the following command:

```
<syntaxhighlight lang="bash">
```

```
hpiLO-> show firmware revisions c1
Product Name: ProLiant m800 Server Cartridge
Cartridge Satellite Firmware: 10/01/2014
Cartridge System ROM Firmware: U01 10/04/3014
Cartridge Data: 10/31/2014
Cartridge Mezz 1 Data: 10/31/2014
Cartridge Programmable Logic Device: 09
```

```
</syntaxhighlight>
```

Please visit <http://www.hp.com/go/moonshot/download> to download "Moonshot component pack 2014.12.0" for above firmware required for the MCSDK-HPC release.

Ubuntu Kernel Version

The Ubuntu Kernel version can be queried from command prompt:

```
<syntaxhighlight lang="bash">
```

```
root@He1c28N1:~# uname -a
```

```
Linux NewCo2N1 3.13.0-24-keystone #38-Ubuntu SMP Fri Mar 13 05:35:03 UTC 2015 armv7l armv7l armv7l GNU/Linux </syntaxhighlight>
```

If the Ubuntu version needs to be updated, use the following commands:

```
<syntaxhighlight lang="bash"> ubuntu@c1n1:~#sudo -E apt-add-repository ppa:marcola-team/ppa ubuntu@c1n1:~#sudo apt-get update ubuntu@c1n1:~#sudo apt-get install linux-image-mcsdk-anchor </syntaxhighlight>
```

Set System Clock

Add network time server in /etc/rc.local if needed. For example:

```
<syntaxhighlight lang="bash"> ubuntu@c1n1:~#sudo ntpdate 128.247.5.20 157.170.1.4 157.170.147.6 Or ubuntu@c1n1:~#sudo date 0225080014 </syntaxhighlight>
```

MCSDK-HPC Installation

MCSDK-HPC packages can be installed from either of two repositories: Partner or PPA.

Partner repository contains MCSDK-HPC packages and licenses that have been officially reviewed and meet Debian package requirements, whereas PPA has more experimental software.

Option 1: Prepare ProLiant m800 Server Node to Fetch TI Packages from Partner repository

NOTE: If you have already installed an earlier version of MCSDK-HPC, please see instructions [Uninstall HPC release \(http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Uninstalling_MCSDK-HPC\)](http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Uninstalling_MCSDK-HPC) to uninstall the software prior to installing this latest version.

Users will need to uncomment below section in /etc/apt/sources.list to fetch/install MCSDK HPC software.

```
<syntaxhighlight lang="bash"> deb http://archive.canonical.com/ubuntu trusty partner deb-src http://archive.canonical.com/ubuntu trusty partner </syntaxhighlight>
```

Option 2: Prepare ProLiant m800 Server Node to Fetch TI Packages from PPA

NOTE: If you have already installed an earlier version of MCSDK-HPC, please see instructions [Uninstall HPC release \(http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Uninstalling_MCSDK-HPC\)](http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Uninstalling_MCSDK-HPC) to uninstall the software prior to installing this latest version.

Users will have to add TI PPA repositories to fetch/install MCSDK HPC software.

```
<syntaxhighlight lang="bash"> ubuntu@c1n1:~#export http_proxy="http://<your_proxy>:<port>" ubuntu@c1n1:~#export https_proxy="http://<your_proxy>:<port>"
ubuntu@c1n1:~#sudo apt-get install python-software-properties ubuntu@c1n1:~#sudo apt-get install software-properties-common ubuntu@c1n1:~#sudo -E apt-add-repository
ppa:marcola-team/ppa ubuntu@c1n1:~#sudo -E apt-add-repository ppa:ti-keystone-team/keystone-hpc-3.0.1.12 </syntaxhighlight>
```

Install MCSDK-HPC

Users are required to add into "keystone-hpc" group for using MCSDK-HPC:

To created "keystone-hpc" group: `ubuntu@c1n1:~#sudo groupadd keystone-hpc` To add existing user "mpiuser" to "keystone-hpc" group: `ubuntu@c1n1:~#sudo usermod -a -G keystone-hpc mpiuser` To check if "mpiuser" is successfully added to the group: `ubuntu@c1n1:~#id mpiuser`

```
uid=1000(mpiuser) gid=1000(mpiuser) groups=1000(mpiuser),1001(keystone-hpc)
```

</syntaxhighlight>

Use the following commands to fetch the latest MCSDK-HPC (keystone-hpc) from Partner or PPA repository.

```
ubuntu@c1n1:~#sudo apt-get update ubuntu@c1n1:~#sudo apt-get install keystone-hpc
```

Reboot ProLiant m800 Server Node

The installation is now complete. Please reboot ProLiant m800 server node.

Verify Installation

After a reboot, the state of the CMEM kernel module can be used as a litmus test for the success of the installation.

Upon successful installation, the CMEM module will be automatically inserted and given the proper permissions.

1. Verify CMEM module was automatically inserted.

- Use the "lsmod" command to get a list of all kernel modules:

```
ubuntu@c1n1:~#lsmod
```

```
Module                Size  Used by
cmemk                 25691  0
uio_module_drv        5020   0
```

</syntaxhighlight>

If "cmemk" is not listed, then there was an issue with installation. Please consult the Troubleshooting Guide.

2. Verify that CMEM has the correct permissions.

- Default installation will give CMEM world read/write permissions:

```
ubuntu@c1n1:~#ls -l /dev/cmем
```

```
crw-rw-rw- 1 root root 250, 0 Apr 30 08:46 /dev/cmем
```

</syntaxhighlight>

These permissions are controlled by the UDEV rules found in the file `/etc/udev/rules.d/20-tci6636k2h.rules`, and may be modified to suit system security requirements.

3. Verify that mpmsrv is running.

- Default installation is setting up upstart job to start mpmsrv (`/etc/init/mpm.conf`)

```
ubuntu@c1n1:~#ps ax | grep "mpmsrv" 1820 ? Sl 0:00 /usr/sbin/mpmsrv
```

</syntaxhighlight> mpmsrv need to be started prior to `/etc/init/keystone-hpc.conf` upstart task, as it is used to enable and configure Hyperlink ports

Compiling MCSDK HPC Examples

1. Copy examples to a user writable directory.

```
ubuntu@c1n1:~#mkdir -v ~/mcsdk-hpc-examples ubuntu@c1n1:~#cp -rv /usr/share/ti/examples/* ~/mcsdk-hpc-examples
```

2. Run top-level Makefile

```
ubuntu@c1n1:~#cd ~/mcsdk-hpc-examples ubuntu@c1n1:~#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 \(http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Examples\)](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)

Running Automated Tests

MCSDK-HPC provides a way of running all the HPC demos automatically over a set of SOCs. After a successful installation of MCSDK-HPC To run those tests, please follow the steps below

Test setup

Once we decide on the set of hosts(nodes) where the automated tests are going to be run, please ensure the following

- MCSDK-HPC installation was successful on all the hosts
- All the hosts are physically reachable directly or indirectly through SRIO and Hyperlink
- Seamless SSH amongst the node is setup. i.e ssh keys are copied among each other using 'ssh-copy-keys'

Copy the autotest files

Copy the /usr/share/ti/keystone-hpc/autotest folder to the current directory

```
> cp -r /usr/share/ti/keystone-hpc/autotest .
```

Run the automated test

```
> cd autotest
> ./test_release.sh --run-on-hosts <hostname1> <hostname2> ...
```

The above command will run the automated tests on hostname1, hostname2 and so on. For example

```
> ./test_release.sh --run-on-hosts skip-rebuild c1n1 c1n2 c1n3 c1n4
```

Part of the above command involves copying the demo examples to the current directory, recompiling them and copying the executables to all the hosts before the tests are being run. So, if you would like to skip the portion of recompiling the examples and copying (for example, to just run the test on the binaries produced in an earlier run), you could do it with the 'skip-rebuild' option appended to the script. So, the test command would look like

```
> ./test_release.sh --run-on-hosts skip-rebuild <hostname1> <hostname2> ...
```

For example

```
> ./test_release.sh --run-on-hosts skip-rebuild c1n1 c1n2 c1n3 c1n4
```

Please note that skip-rebuild should be used ONLY if you had successfully run the automated tests atleast once before from the same location (with no skip-rebuild option).

NOTE: Please note that auto test can be run on a maximum of 4 nodes now.

Verify the test results

The test takes about 15 minutes or so. The results of the test against each test ,are printed on screen like below,

```
> "openmpi+opencl:multinode_batch_fftdemo-np2-srio" ... [PASS]
> "opencl+openmp:dgemm" ... [PASS]
> "opencl+openmp:vecadd_openmp" ... [PASS]
> "opencl+openmp:vecadd_openmp_t" ... [PASS]
> Complete
> test_release_time : 8:13
> TEST COMPLETE
```

NOTE: The following failures are expected.

```
> "opencl:ooo" ... [FAIL]
> "opencl:mandelbrot" ... [FAIL]
> "opencl:mandelbrot_native" ... [FAIL]
```

The logs of test execution are kept under separate folder created every run, based on the timestamp. The folder name is printed at the beginning of the test, For example,

```
> CONFIG: HOSTS_FILE = /root/re1_autotest/autotest/hosts
> CONFIG: TEST_USER = root
> CONFIG: EXAMPLES_DIR = /root/re1_autotest/autotest/mcsdk-hpc-examples
> CONFIG: NODE_KNOWN_HOSTS = /root/.ssh/known_hosts
> mkdir: created directory /root/re1_autotest/autotest/20150202211040
> mkdir: created directory /root/re1_autotest/autotest/20150202211040/test_logs ---> This is the directory where the logs are present for each test case.
> mkdir: created directory /root/re1_autotest/autotest/20150202211040/logs
```

In the above example, the test logs for a particular run is kept at 20150202211040/test_logs directory

Uninstalling MCSDK-HPC

To uninstall prior installations of MCSDK-HPC (aka keystone-hpc), please do the following steps

- Purge the previous installation of HPC.

```
<syntaxhighlight lang="bash"> ubuntu@c1n1:~#sudo apt-get purge keystone-hpc ubuntu@c1n1:~#sudo apt-get autoremove
```

```
</syntaxhighlight>
```

- If **ti-keystone-ppa** was added manually to `/etc/apt/sources.list`, please remove the entries from this file.
- If you used "apt-add-repository" command (as suggested [here](http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Option_2:_Prepare_ProLiant_m800_Server_Node_to_Fetch_TI_Packages_from_PPA) (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800#Option_2:_Prepare_ProLiant_m800_Server_Node_to_Fetch_TI_Packages_from_PPA)) to add PPAs, then do the following to remove those PPAs.

```
<syntaxhighlight lang="bash"> ubuntu@c1n1:~#sudo -E apt-add-repository -r ppa:ti-keystone-team/keystone-hpc-3.0.1.12 </syntaxhighlight>
```

Establish Two K2H Nodes for OpenMPI Demos

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

1. Modify `/etc/hostname` to use `k2hnode1` for K2H node 1 and `k2hnode2` for K2H node 2.

2. Find IP addresses of the K2H nodes (eth0) using `ifconfig`. Then, edit `/etc/hosts` of both K2H nodes to include IP addresses of both K2H nodes and their corresponding hostname.

```
<syntaxhighlight lang="bash"> 127.0.0.1 localhost.localdomain localhost [ip address of K2HEVM node1] k2hnode1 [ip address of K2HEVM node2] k2hnode2 </syntaxhighlight>
```

3. For both K2H nodes, add `mpiuser` (user: `mpiuser`, password: `gguser502`) if it has not been added earlier.

```
<syntaxhighlight lang="bash"> ubuntu@k2hnode1:~#sudo adduser mpiuser </syntaxhighlight>
```

4. Do SSH between the two nodes.

Log in as `mpiuser` on both setups and then do SSH.

```
<syntaxhighlight lang="bash"> [k2hnode1] ssh mpiuser@k2hnode2, accept [w/ yes] then exit [k2hnode2] ssh mpiuser@k2hnode1, accept [w/ yes] then exit </syntaxhighlight>
```

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

5. Set the password for root (user: `root`, password: `gguser502`) and then do SSH.

```
<syntaxhighlight lang="bash"> root@k2hnode1:~# passwd [k2hnode1] ssh root@k2hnode2, accept [w/ yes] then exit [k2hnode2] ssh root@k2hnode1, accept [w/ yes] then exit </syntaxhighlight>
```

After this step, file `~/.ssh/known_hosts` is updated with the root as user.

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
TI CGTools Download	https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm

DSP Debug and Profiling Utilities

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

	Product Download Link
dsptop	http://processors.wiki.ti.com/index.php/Dsptop
Hosted C66x GDB	http://processors.wiki.ti.com/index.php/Hosted_C66x_GDB

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
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. "FF1").

Troubleshooting

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

- I see an error "pycurl.error: (60, 'server certificate verification failed. CAfile: /etc/ssl/certs/ca-certificates.crt CRLfile: none)" when doing apt-add-repository [\[Expand\]](#)
- I see timeout report if Hyperlink BTL (MPI) on two nodes is used. [\[Expand\]](#)
- MPI examples using Hyperlink BTL are not working (SoC may get stuck requiring reboot). [\[Expand\]](#)
- Upgrading from MCSDK-HPC-3.0.1.0 (or earlier) to MCSDK-HPC-3.0.1.1** [\[Expand\]](#)
- New Question Template


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) (http://processors.wiki.ti.com/index.php/MCSDK_HPC_3.x_Trouble_shooting).

Archived Pages

- MCSDK HPC Releases Prior to 3.0.1.3** : [1] (http://processors.wiki.ti.com/index.php?title=MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800&oldid=185242)

<p>1. switchcategory:MultiCore=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>Keystone=</p> <ul style="list-style-type: none"> For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum <p>Please post only comments related to the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>DaVinci=For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>	<p>MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article MCSDK HPC 3.x Getting Started Guide for HP ProLiant m800 here.</p>
--	---	---	--	--	---	--	---

Links

 <ul style="list-style-type: none"> Amplifiers & Linear Audio Broadband RF/IF & Digital Radio Clocks & Timers Data Converters 	<ul style="list-style-type: none"> DLP & MEMS High-Reliability Interface Logic Power Management 	<ul style="list-style-type: none"> Processors <ul style="list-style-type: none"> ARM Processors Digital Signal Processors (DSP) Microcontrollers (MCU) OMAP Applications Processors 	<ul style="list-style-type: none"> Switches & Multiplexers Temperature Sensors & Control ICs Wireless Connectivity
--	--	---	---

Retrieved from "https://processors.wiki.ti.com/index.php?title=MCSDK_HPC_3.x_Getting_Started_Guide_for_HP_ProLiant_m800&oldid=207877"

This page was last edited on 7 October 2015, at 23:49.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.