

MCSDK UG Chapter Getting Started



MCSDK User Guide: Getting Started with the MCSDK

Last updated: 06/17/2015

Contents

Overview

Acronyms

Supported Devices and Platforms

What's in this Kit?

Getting Started

- Setup Hardware

- Setup Software

 - Code Composer Studio

 - Emulator Support

 - Multicore Software Development Kit

- Update EVM On-Board Flash

- Run Out of Box Demonstrations

Next Steps

How To

- Manually get IP address from Linux

- Set up TFTP Server

- U-boot quick update for K2H EVM

 - Procedures

- Address basic issues with out of box demonstration

Overview

The Multicore Software Development Kit (MCSDK) provides foundational software for TI KeyStone II device platforms. It encapsulates a collection of software elements and tools intended to enable customer application development and migration.

This chapter provides information on installing the MCSDK, loading the EVM flash images via JTAG, and running the out-of-box demonstration application using the updated software.

By the end of this *Getting Started* chapter, the user will have

- Installed Code Composer Studio
- Installed latest Emulation package
- Installed the MCSDK software
- Update EVM flash with release images
- Executed the out-of-box demonstration application using latest images



Useful Tip

The EVM comes with a *Quick Start Guide* that provides instructions to setup hardware and boot the out-of-box demonstration loaded on the EVM.

Acronyms

The following acronyms are used throughout this chapter.

Acronym	Meaning
CCS	Texas Instruments Code Composer Studio
DDR	Double Data Rate
DHCP	Dynamic Host Configuration Protocol
DSP	Digital Signal Processor
EVM	Evaluation Module, hardware platform containing the Texas Instruments DSP
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
JTAG	Joint Test Action Group
MCSDK	Texas Instruments Multi-Core Software Development Kit
PDK	Texas Instruments Programmers Development Kit
RAM	Random Access Memory
TI	Texas Instruments
UBIFS	Unsorted Block Image File System

Supported Devices and Platforms

This release supports the following Texas Instrument devices/platforms:

Platform	Supported Devices	Supported EVM
[K2H (http://www.ti.com/multicore)]	TCI6636K2H (http://www.ti.com/product/tci6636k2h), 66AK2H06 (http://www.ti.com/product/66ak2h06), 66AK2H12 (http://www.ti.com/product/66ak2h12), 66AK2H14 (http://www.ti.com/product/66ak2h14)	XTCIEVMK2X, EVMK2H (http://www.ti.com/tool/evmk2h)
[K2K (http://www.ti.com/multicore)]	TCI6638K2K (http://www.ti.com/product/tci6638k2k)	XTCIEVMK2X
[K2E (http://www.ti.com/multicore)]	66AK2E02 (http://www.ti.com/product/66ak2e02), 66AK2E05 (http://www.ti.com/product/66ak2e05), AM5KE02 (http://www.ti.com/product/am5k2e02), AM5K2E04 (http://www.ti.com/product/am5k2e04)	EVMK2E (http://www.ti.com/tool/xevmk2ex)
[K2L (http://www.ti.com/multicore)]	TCI6630K2L (http://www.ti.com/product/tci6630k2l), 66AK2L06 (http://www.ti.com/product/66ak2l06)	TCIEVMK2L

What's in this Kit?

The kit contains

- **EVM Board:** This board contains a Multicore System-on-Chip
- **Universal Power Supply:** Both U.S. and European power specifications are supported
- **Cables:** USB, Serial, and Ethernet cables are included to allow for host development
- **Software:** Software installers, documentation, factory images
- **EVM Quick Start Guide:** Single page guide to get started run demos pre-flashed on board

The software media (e.g., DVD, SD card, or USB) contains

- Multicore Software Development Kit source packages
- Code Composer Studio installer
- Factory image restoration process
- Hardware documentation

All EVMs have both NAND and NOR flash, though they may be different in size. Typical use case is to load uboot on NOR and boot from NOR. Due to greater NAND flash size, the software file system is loaded on NAND in a UBIFS format.

Getting Started

Setup Hardware

The EVM provides the ability to utilize a variety of capabilities of the SoC. Follow instructions in one of the links below that correspond to the EVM that you are using for information on configuration, board firmware updates, and further pertinent information:

- XTCIEVMK2X, EVMK2H : Use [EVMK2H Hardware Setup \(http://processors.wiki.ti.com/index.php/EVMK2H_Hardware_Setup\)](http://processors.wiki.ti.com/index.php/EVMK2H_Hardware_Setup)
- EVMK2E: Use [EVMK2E Hardware Setup \(http://processors.wiki.ti.com/index.php/EVMK2E_Hardware_Setup\)](http://processors.wiki.ti.com/index.php/EVMK2E_Hardware_Setup)
- TCIEVMK2L : Use [TCIEVMK2L Hardware Setup \(http://processors.wiki.ti.com/index.php/TCIEVMK2L_Hardware_Setup\)](http://processors.wiki.ti.com/index.php/TCIEVMK2L_Hardware_Setup)

Setup Software

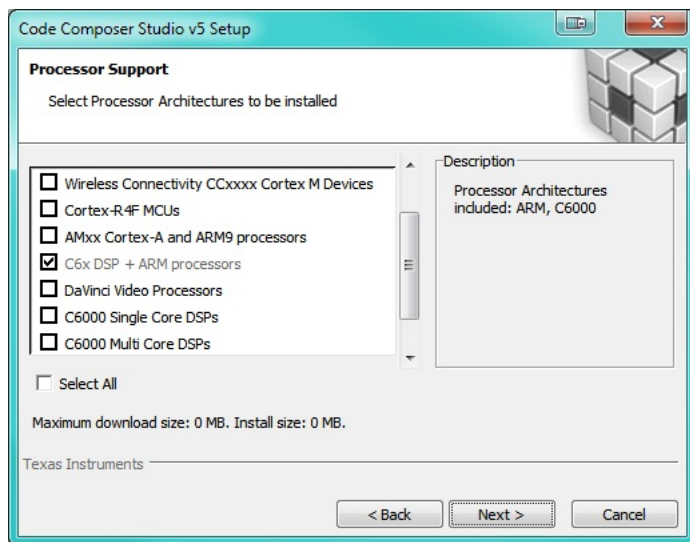
Code Composer Studio

The MCSDK uses *Code Composer Studio* as the host integrated development environment for development and debug. Install the version of CCSv5 indicated in the MCSDK Release Notes. This is the version of CCS that was validated with the software provided in the MCSDK and will provide the best user experience. To install CCS, please refer to the instructions provided in the [CCSv5 Getting Started Guide \(http://processors.wiki.ti.com/index.php/CCSv5_Getting_Started_Guide\)](http://processors.wiki.ti.com/index.php/CCSv5_Getting_Started_Guide).

 **Note:** MCSDK version 3.1.2 and later have been validated with CCS version 6.0.1. The instructions can be found in the [CCSv6 Getting Started Guide \(http://processors.wiki.ti.com/index.php/CCSv6_Getting_Started_Guide\)](http://processors.wiki.ti.com/index.php/CCSv6_Getting_Started_Guide).

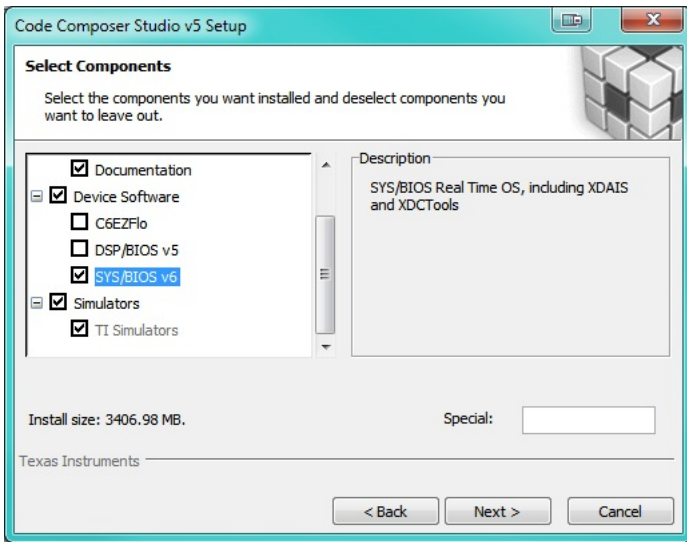
When installing CCS you can choose to control what is installed for processor architecture and components. The minimum required for MCSDK are

Processor Architecture option:



 **Note:** If CCS v5.5 is used, please select "C6000 Multi Core DSPs" option, and if ARM environment is desired, please also select "Sitara AMxxxx processors" so the ARM compiler will be installed. Though Sitara A8 ARM compiler works for A15 family processor, it is suggested that Linaro GCC compiler should be installed and used for ARM A15 family processor.

Component option:



Note: There is a known issue with ARM debug on CCS v5.5, see this [forum post \(http://e2e.ti.com/support/development_tools/code_composer_studio/f/81/t/317804.aspx\)](http://e2e.ti.com/support/development_tools/code_composer_studio/f/81/t/317804.aspx) for instructions on applying a patch to fix the issue.

Emulator Support

The GEL and XML files for connecting a JTAG to the KeyStone devices supported in MCSDK are provided in the *TI KeyStone2 Emupack* release. See the Downloads (http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Downloads) chapter for information on how to get the latest version of this package. This software package is required for using an emulator to debug software over JTAG as well as the `program_evm` tool to update the EVM flash with latest images.

Multicore Software Development Kit

The final step is to install the MCSDK. See the Downloads (http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Downloads) chapter for information on how to get the latest version of this software. The release contents are delivered as installers and source tarball. Some notes:

- To avoid manually updating file locations, it is recommended to install the MCSDK in the same directory as CCSv5.
- For a Linux host, run the `mcsdk_<build-id>_setuptoolslinux.bin` executable. Starting with MCSDK 3.1.0, the GUI interface has been replaced with a command line installer.
- For a Windows host, run the `mcsdk_<build-id>_setupwin32.exe` executable. Starting with MCSDK 3.1.0, this package is focused on DSP source; use the above Linux package for Linux sources and images.
- Starting with MCSDK 3.1.x, the release includes a `mcsdk_<build-id>.tar.gz`, which can be untarred on any platform.
- Some of the components installed by CCS may be updated by the MCSDK installer (e.g., SYS/BIOS); see the MCSDK release notes for a mapping of each component to the software package (CCS or MCSDK).

Note: For MCSDK 3.0.x Linux installer, `mcsdk_x_xx_xx_xx_setuptoolslinux.bin`, is a 32-bit binary. To run this installer on a 64-bit machine, an adaption/compatibility library, `ia32-libs`, needs to be installed. On Ubuntu 12.04 machine, the following command is used to install `ia32-libs`:

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

Note: For MCSDK 3.1.x or later releases: The source tarball included in the release can be installed on a Linux machine using the steps as follows. To see the list of packages in the MCSDK release tar ball:

```
tar tzvf mcsdk_<build-id>.tar.gz --exclude="*/**"
```

To install the full package in a specified directory (Say `~/ti`)

```
tar xzvf mcsdk_<build-id>.tar.gz -C ~/ti
```

This will install the different packages under `~/ti`, with their respective package_<version> directories). In addition, the XDC package `xdc_<version>_core6x` in the tar package is meant for ARM native compilation only. If using x86 based host, this can be removed to avoid issues with XDC compilation.

To install a specific package only from the tar ball:

```
tar xzvf mcsdk_<build-id>.tar.gz -C ~/ti --wildcards "mcsdk_linux"
```

Update EVM On-Board Flash

To update on-board flash media on the EVM (NOR, NAND, EEPROM), use the `program_evm` script located in the `mcsdk_bios_3_xx_xx_xx/tools/` directory that updates all EVM images automatically via command line. The script uses CCS DSS scripts to load and run a utility from a SoC C6x DSP core to load and burn flash.

The steps for this are described in the [Program EVM User Guide](#). To use *program_evm* you need to install CCS and have a connection between your host PC and the EVM via JTAG. When running the *program_evm*, make sure the CCS GUI is not connected to the target.

Useful Tip

Helpful
Tips

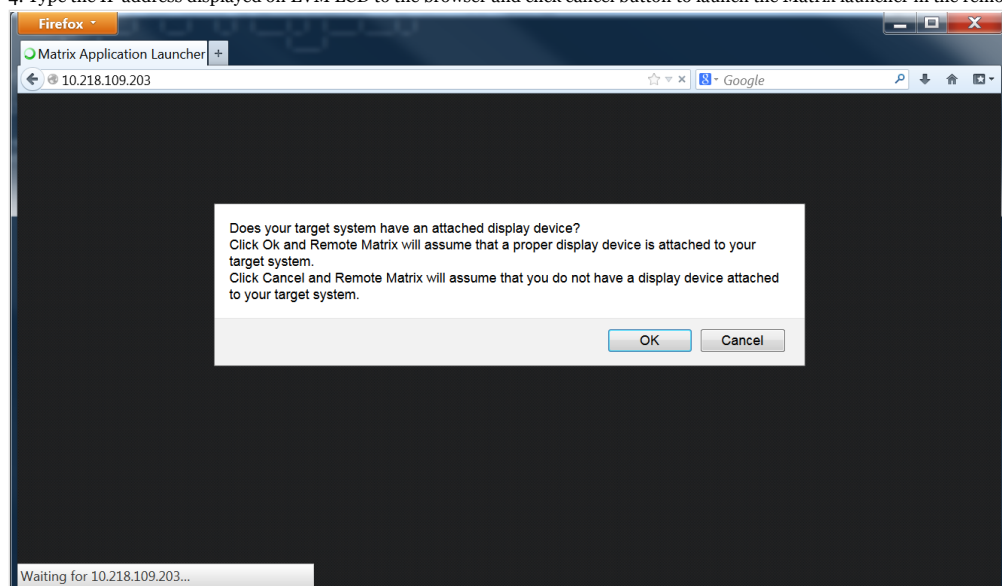
The images that are loaded by *program_evm* are kept in the `/program_evm/binaries/evmk2x/` directory. Since you can substitute any image here, this is a quick way to update flash with custom images.

Run Out of Box Demonstrations

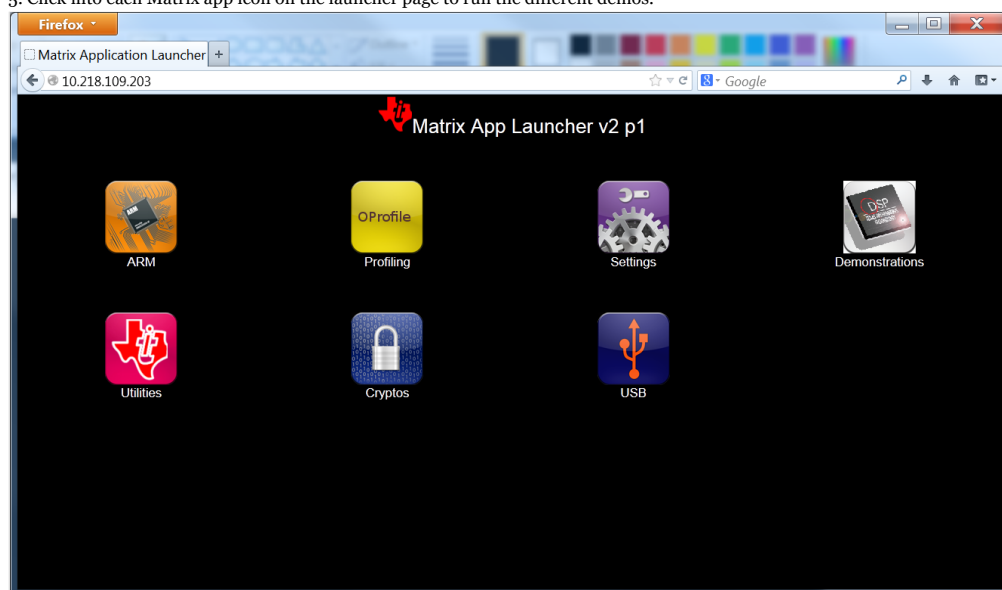
Please follow the below steps to run the Matrix Application demos, the demos can be launched by a GUI Matrix launcher from a remote browser.

1. Configure uboot to set the DSP load reserve area to match pre-build DSP libraries:

```
<syntaxhighlight lang="c"> env default -f -a setenv mem_reserve 1536M saveenv boot</syntaxhighlight>
```
2. Connect the EVM Ethernet port 0 to a corporate or local network with DHCP server running, when the Linux kernel boots up, the rootfs start up scripts will get an IP address from the DHCP server and print the IP address to the EVM on-board LCD.
3. Open an Internet browser (e.g. Mozilla Firefox) on a remote computer that connects with the same network as the EVM.
4. Type the IP address displayed on EVM LCD to the browser and click cancel button to launch the Matrix launcher in the remote access mode instead of on the on-board display device.



5. Click into each Matrix app icon on the launcher page to run the different demos.



Note: Linux kernel `vmlinux` image currently is not included in the rootfs due to the NAND flash size limitation, to run the Oprofile without profiling the kernel, enter the following command in the Linux command shell:

- `opcontrol --no-vmlinux`

Please refer to the Oprofile Init demo instructions for details.

Next Steps

See [Software Design Guide \(http://processors.wiki.ti.com/index.php/66AK2Hx_Software_Design_Guide\)](http://processors.wiki.ti.com/index.php/66AK2Hx_Software_Design_Guide) on timeline guidance for your Keystone II software development.

See the "Developing with MCSDK" chapter of this User Guide on a section to create a simple "Hello World" application. More advanced topics are covered in different sections of that chapter.

How To

Manually get IP address from Linux

If IP address is not known, please connect USB console and get IP address from Linux using *ifconfig* command. Log in as root (no password) if prompt for user.

Set up TFTP Server

If loading images using TFTP, an example on Ubuntu TFTP server set up can be found [here \(http://askubuntu.com/questions/201505/how-do-i-install-and-run-a-tftp-server\)](http://askubuntu.com/questions/201505/how-do-i-install-and-run-a-tftp-server).

On Ubuntu 12.04, it is recommended to use tftpd-hpa server as the TFTP server since the default one has shown some slower transfer speeds.

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

You will need to update the `/etc/default/tftpd-hpa` config file

```
# /etc/default/tftpd-hpa
```

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS="0.0.0.0:69"
TFTP_OPTIONS="--secure"
```

U-boot quick update for K2H EVM

The following instructions is an example of loading U-boot to a **K2H EVM** in a **Linux** environment.

The main assumption here is that you are familiar with the overall process, but you should review the following related subjects:

http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Exploring#Loading_and_running_U-Boot_on_EVM_through_SPI_Boot

http://processors.wiki.ti.com/index.php/Program_EVM_UG

Procedures

[1] Install Code Composer Studio (**CCS**) from TI download website to your Linux PC.

[2] Install **CCS EMUPACK files** from: http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk/latest/index_FDS.html

[3] Install **MCSDK files** from: http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk/latest/index_FDS.html

[4] Note: You may need to change the MCSDK file ownership and rights for all the installed files in your environment.

[5] Put the board in **no boot mode** and connect emulator USB-mini cable to the top USB daughter card (emulator board).

You should record your SW1 **Normal** switch settings on the board and then **push all the switches away from the edge of the board** for programming U-boot into NOR in **No Boot Mode**.

For **Normal** Operations:
SW 1 SW2 SW3 SW4
OFF OFF ON OFF

For **No Boot Mode** operations:
SW 1 SW2 SW3 SW4
ON ON ON ON (Note: Away from the edge of the board)

[6] `cd <MCSDK_INSTALL_PATH>/mcsdk_bios_3_01_xx_yy/tools/program_evm/`

[7] Execute the following command in the **program_evm** folder:

```
<CCS_INSTALL_PATH>/ccsvX/ccs_base/scripting/bin/dss.sh program_evm.js evmk2h nor
```

[8] **Power off EVM and place the configuration switches back to the normal mode** recorded earlier.

Typically from left to right OFF, OFF, ON, OFF
Connect your serial port to Normal console port (marked as SOC which is closest to the board edge) and power on the EVM to see the normal boot operations.

If you run into any issues with running out of box demonstration, be sure the following was done properly:

- BMC update from HW Setup Guide
- UCD update from HW Setup Guide
- Correct boot mode setting from HW Setup Guide (typically SPI NOR Little Endian)

1. switchcategory:MultiCore= - 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	Keystone= - 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	C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article MCSDK UG Chapter Getting Started here.	DaVinci=For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article MCSDK UG Chapter Getting Started here.	MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article MCSDK UG Chapter Getting Started here.	OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article MCSDK UG Chapter Getting Started here.	OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article MCSDK UG Chapter Getting Started here.	MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article MCSDK UG Chapter Getting Started here.
---	--	---	--	--	---	--	---

Links



<u>Amplifiers & Linear</u>	<u>DLP & MEMS</u>	<u>Processors</u>	<u>Switches & Multiplexers</u>
<u>Audio</u>	<u>High-Reliability</u>	▪ <u>ARM Processors</u> ▪ <u>Digital Signal Processors (DSP)</u> ▪ <u>Microcontrollers (MCU)</u> ▪ <u>OMAP Applications Processors</u>	<u>Temperature Sensors & Control ICs</u>
<u>Broadband RF/IF & Digital Radio</u>	<u>Interface</u>		<u>Wireless Connectivity</u>
<u>Clocks & Timers</u>	<u>Logic</u>		
<u>Data Converters</u>	<u>Power Management</u>		

Retrieved from "https://processors.wiki.ti.com/index.php?title=MCSDK_UG_Chapter_Getting_Started&oldid=201928"

This page was last edited on 17 June 2015, at 11:09.

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