

BIOS C6SDK 2.0 User Guide

Introduction

The BIOS C6 Software Development Kit (BIOS C6SDK) provides the core foundational building blocks that facilitate application software development on TI's high performance DSPs. The foundational components include:

- SYS/BIOS which is a light-weight real-time embedded operating system for TI devices
- Drivers
- Basic networking stack and protocols
- Optimized application-specific and application non-specific algorithm libraries
- Flash and boot utilities
- Demonstrations

The purpose of this User's Guide is to provide more detailed information regarding the software elements and infrastructure provided with BIOS C6SDK. BIOS C6SDK pulls together all the elements into demonstrable DSP applications and examples for supported EVMs. The objective being to demonstrate device, platform, software capabilities and functionality as well as provide the user with instructive examples. The software provided is intended to be used as a reference when starting their development.

Note: It is expected the user has gone through the EVM Quick Start Guide provided with their EVM. It is also assumed the user has gone through the BIOS C6SDK Getting Started Guide and have installed both CCS and the BIOS C6SDK.

Supported Devices/Platforms

The latest BIOS C6SDK Release supports the following Texas Instrument devices/platforms:

Platform Development Kit Supported Devices Supported EVM

Platform Development Kit	Supported Devices	Supported EVMs
C6748	C6748, OMAPL138	TMS320C6748/OMAP-L138
C6748	C6748, OMAPL138	LCDK (C6748 L13x/C674x Development Kit)

Other Documentation

Getting Started Guides

The getting started guides walk you through setting up your EVM . This is where you should start after receiving your EVM.

Document	Description
BIOS C6SDK Release Notes	Contains latest information on the release including what's changed, known issues and compatibility information. Each foundational component will have individual release notes as well.
BIOS C6SDK <i>Getting Started Guide</i>	Discusses how to install the BIOS-SDK and access the demonstration application

Tools Overview

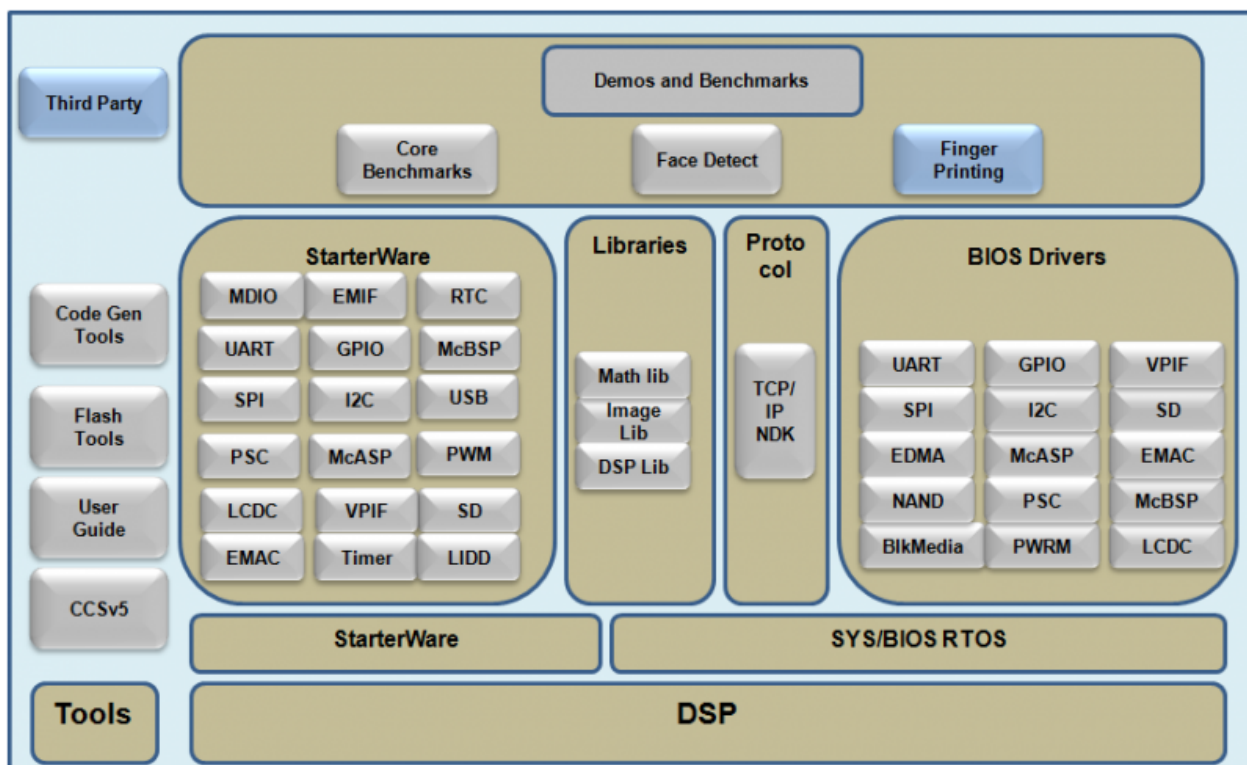
The following documents provide information on the various development tools available to you.

Document	Description
CCS v5 Getting Started Guide ^[1]	How to get up and running with CCS v5
TMS320C6000 Code Generation Tool v 7.3.1 ^[2]	Everything you wanted to know about the compiler, assembler, library-build process and C++ name demangler.
Starterware ^[3]	Peripheral code

Software Overview

As can be seen in the architecture diagram, the BIOS C6SDK consists of those components in the boxes. Specifically they are:

- Platform Software consisting Starterware/SYS BIOS Drivers, Network driver
- SYS/BIOS
- Network Developer's Kit (NDK) Package
- Algorithm Libraries
- Demonstration Applications



Platform Development Kit (PDK)

The Platform Development Kit (PDK) is a package that provides the foundational drivers and software to enable the device. It also contains the transport (NSP), Ethernet driver, Chip Select Configuration Files (CSLR).

SYS BIOS Drivers

The C6748 BIOS PSP is aimed at providing fundamental software abstractions for onchip resources and plugs the same into SYS/BIOS operating system so as to enable and ease application development by providing suitably abstracted interfaces.

The source code for the Drivers is located under `pdk_c6748_<version>\biospsp_<version>\` directory.

The following table shows Driver Vs. SoC Availability

Driver	C6748	LCDK
BLKMEDIA	X	No
GPIO	X	X
I2C	X	X
LCDCRaster	X	No
McASP	X	X
McBSP	X	No
SD	X	X
NAND	X	X
PSC	X	No
SPI	X	No
UART	X	X
VPIF	X	No

Driver Library Summary	
Component Type	Library
Install Package	PDK
Install Directory	<code>pdk_c6748_<version>\</code>
Project Type	Eclipse RTSC ^[4]
Endian Support	Little
Linker Path	<code>\$(BIOSPSP_INSTALL_DIR)\drivers</code>
Linker Sections	N/A
Section Preference	N/A
Include Paths	<code>\$(BIOSPSP_<version number>)\drivers</code>
Reference Guides	See docs under Install Directory
License	BSD ^[5]

Chip Support Register Library (CSLR)

This document describes the Chip Support Register Configuration files provided. This layer provides low-level register and bit field descriptions for the device and its peripherals, and a set of macros for basic register configuration. It may be used as a foundation for building complex drivers or on its own to perform register configuration and check peripheral status. The source code is available at `pdk_c6748_<version>\biosp_<version>\cslr`.

Chip Support Library Summary	
Component Type	Library
Install Package	PDK
Install Directory	<code>pdk_c6748_<version>\</code>
Project Type	Eclipse RTSC ^[4]
Endian Support	Little
Linker Path	<code>\$(BIOSPSP_INSTALL_DIR)\cslr</code>
Include Paths	<code>\$(BIOSPSP_<version number>)\cslr</code>
Reference Guides	See docs under Install Directory
License	BSD ^[5]

Starterware

The C6748 Starterware is aimed at providing fundamental software abstractions for onchip resources which is independent of Operating system.

The source code for the Starterware is located under `pdk_c6748_<version>\C6748_StarterWare<version>\` directory.

The following table shows Driver Vs. SoC Availability

Driver	C6748	LCDK
Timer	X	X
EMAC	X	X
MDIO	X	X
EHRPWM	X	X
EDMA	X	X
EMIF	X	X
GPIO	X	X
I2C	X	X
LIDD	X	
Raster LCD	X	X
McASP	X	X
RTC	X	X
MMCSDB	X	X
NAND	X	X

SPI	X	X
USB (HID, CDC, MSC, Bulk)	X	X
UART	X	X
VPIF	X	Capture

Driver Library Summary	
Component Type	Library
Install Package	PDK
Install Directory	pdk_c6748_<version>\
Project Type	Eclipse RTSC ^[4]
Endian Support	Little
Linker Path	\$(C6748_StarterWare_INSTALL_DIR)\
Linker Sections	N/A
Section Preference	N/A
Include Paths	\$(C6748_StarterWare_<version number>)\drivers
Reference Guides	See docs under Install Directory
License	TI Commercial

NDK Support package (NSP)

The Network Development Kit Support Package (NSP) contains ethernet driver code, libraries and network examples for various TI platforms. It is meant to supplement the Network Development Kit (NDK), which consists of the platform independent networking code. The NSP includes demonstration software which showcases the OMAPL138 DSP's capabilities across a range of network enabled applications, including both SYS/BIOS 6.x. The NDK examples found in this NSP can serve as a rapid prototype platform for the development of network and packet processing applications, or to add network connectivity to existing applications for communications, configuration, and control. Using the components provided in the NDK and NSP, developers can quickly move from development concepts to working implementations attached to the network. The source code for example is available at `$(PDK_C6748_<version>/nsp_<version>)\packages\ti\ndk\examples\`

Chip Support Library Summary	
Component Type	Examples
Install Package	PDK
Install Directory	pdk_c6748_<version>\
Project Type	Eclipse RTSC ^[4]
Endian Support	Little
Linker Path	\$(NSP_INSTALL_DIR)\packages\ti\drv\omap138
Include Paths	\$(NSP_INSTALL_DIR)\packages\ti\drv\omap138\inc
Reference Guides	See docs under Install Directory
Support	Technical Support
Additional Resources	User Guide ^[6]

Downloads	Product Updates ^[7]
License	BSD ^[5]

EDMA3 Low Level Driver

EDMA3 Low Level Driver is targeted to users (device drivers and applications) for submitting and synchronizing EDMA3-based DMA transfers.

EDMA3 is a peripheral that supports data transfers between two memory mapped devices. It supports EDMA as well as QDMA channels for data transfer. This peripheral IP is re-used in different SoCs with only a few configuration changes like number of DMA and QDMA channels supported, number of PARAM sets available, number of event queues and transfer controllers etc. The EDMA3 peripheral is used by other peripherals for their DMA needs thus the EDMA3 Driver needs to cater to the requirements of device drivers of these peripherals as well as other application software that may need to use DMA services.

The EDMA3 LLD consists of an EDMA3 Driver and EDMA3 Resource Manager. The **EDMA3 Driver** provides functionality that allows device drivers and applications for submitting and synchronizing with EDMA3 based DMA transfers. In order to simplify the usage, this component internally uses the services of the **EDMA3 Resource Manager** and provides one consistent interface for applications or device drivers.

EDMA3 Driver Summary	
Component Type	Library
Install Package	EDMA3 Low level drivers
Install Directory	<root_install_dir>/edma3_lld_<version>
Project Type	N/A
Endian Support	Little
Library Name	edma3_drv_bios.lib (C64x) and edma3_drv_bios_c674.lib (C674x)
Linker Path	N/A
Linker Sections	N/A
Section Preference	N/A
Include Paths	N/A
Reference Guides	See docs under install directory
Support	Technical Support
Downloads	Product Updates ^[8]
License	BSD ^[5]

SYS/BIOS RTOS

SYS/BIOS 6.x is a real-time operating system that provides pre-emptive multitasking services for DSP devices. SYS/BIOS 6.x services include ISR dispatching, tasks, software interrupts, semaphores, messages, device I/O, memory management, and power management.

SYS/BIOS Summary	
Component Type	Libraries
Install Package	BIOS
Install Directory	bios_6_<version>\
Project Type	Eclipse RTSC ^[4]
Endian Support	Little
Library Name	The appropriate libraries are selected for your device and platform as set in the RTSC build properties for your project and based on the use module statements in your configuration.
Linker Path	The appropriate path is selected to the libraries for your device and platform as set in the RTSC build properties for your project.
Linker Sections	N/A
Section Preference	N/A
Include Paths	BIOS_CG_ROOT is set automatically by CCS based on the version of BIOS you have checked to build with. \${BIOS_CG_ROOT}\packages\ti\bios\include
Reference Guides	See docs under Install Directory
Support	Technical Support
Additional Resources	Eclipse RTSC Home ^[4]
Downloads	SYS/BIOS Downloads ^[9]
License	BSD ^[5]

IPC

Interprocess communication mechanisms including message passing, streams, and linked lists, which work transparently in both uniprocessor and multiprocessor configurations.

The IPC product is comprised of the following main packages

ipc: Contains common interface files (e.g. MessageQ, ListMP, HeapMemMP, Notify, etc.)

io: Contains I/O modules (e.g. Stream, IDriver interface, etc.)

utils: Contains helper modules (e.g. MultiProc, List and NameServer)

heaps: Contains shared-memory Heaps

ipc: Contains multicore modules (e.g. MessageQ, Notify, ListMP, etc.)

IPC Summary	
Component Type	Libraries
Install Package	IPC
Install Directory	ipc_<version>\
Project Type	Eclipse RTSC ^[4]
Endian Support	Little
Library Name	The appropriate libraries are selected for your device and platform as set in the RTSC build properties for your project and based on the use module statements in your configuration.
Linker Path	The appropriate path is selected to the libraries for your device and platform as set in the RTSC build properties for your project.
Linker Sections	N/A
Section Preference	N/A
Include Paths	<pre> \${IPC_INSTALL_DIR}\packages\i\ipc \${IPC_INSTALL_DIR}\packages\i\sd\ipc \${IPC_INSTALL_DIR}\packages\i\sd\io \${IPC_INSTALL_DIR}\packages\i\utils </pre>
Reference Guides	See docs under Install Directory
Support	Technical Support
Additional Resources	Eclipse RTSC Home ^[4]
Downloads	IPC Downloads ^[10]
License	BSD ^[5]

Network Development Kit (NDK)

The NDK is a platform for development and demonstration of network enabled applications on DSP devices and includes demonstration software showcasing DSP capabilities across a range of network enabled applications. The NDK serves as a rapid prototype platform for the development of network and packet processing applications, or to add network connectivity to existing DSP applications for communications, configuration, and control. Using the components provided in the NDK, developers can quickly move from development concepts to working implementations attached to the network.

The NDK provides an IPv6 and IPv4 compliant TCP/IP stack working with the DSP BIOS real-time operating system. Its primary focus is on providing the core Layer 3 and Layer 4 stack services along with additional higher-level network applications such as HTTP server and DHCP.

The NDK itself does not include any platform or device specific software. The NDK interfaces through well-defined interfaces to the PDK and platform software elements needed for operation.

Network Development Kit Summary	
Component Type	Libraries
Install Package	NDK
Install Directory	ndk_<version>\
Project Type	Eclipse RTSC ^[4]
Endian Support	Little and Big
Library Name	binsrc.lib or binsrce.lib and cgi.lib or cgie.lib and console.lib or consolee.lib and hdlc.lib or hdlce.lib and miniPrintf.lib or miniPrintfe.lib and netctrl.lib or netctrlr.lib and nettool.lib or nettoole.lib and os.lib or ose.lib and servers.lib or serverse.lib and stack.lib or stacke.lib
Linker Path	\$(NDK_INSTALL_DIR)\packages\ti\ndk\lib\<arch>
Linker Sections	.far:NDK_OBJMEM, .far:NDK_PACKETMEM
Section Preference	L2 Cache
Include Paths	NDK_INSTALL_DIR is set automatically by CCS based on the version of NDK you have checked to build with. \${NDK_INSTALL_DIR}\packages\ti\ndk\inc \${NDK_INSTALL_DIR}\packages\ti\ndk\inc\tools
Reference Guides	See docs under Install Directory
Support	Technical Support
Additional Resources	N/A
Extended Support	Eclipse RTSC Home ^[4] NDK User's Guide ^[11] NDK Programmer's Reference Guide ^[12] NDK Support Package Ethernet Driver Design Guide ^[13] NDK_FAQ ^[14] Rebuilding NDK Core ^[15]
Downloads	NDK Downloads ^[7]
License	BSD ^[5]

Algorithm Libraries

TI provides several algorithm libraries, each specific to a particular arena. Each library provides a collection of C-callable low-level functions (kernels), each tailored for optimal performance on a specific TI processing device (or devices). The libraries are typically used in computationally intensive real-time applications where execution speed is a critical factor. Their use generally accelerates execution speeds well beyond that achieved by equivalent code written in standard ANSI C. Additionally, use of these libraries can significantly reduce application development time. Source code is provided in all cases to facilitate kernel modification when needed.

DSP Library (DSPLIB)

DSPLIB is an optimized DSP Function Library and includes many C-callable, optimized, general-purpose signal-processing routines including:

- Adaptive Filtering
- Correlation
- Fast Fourier Transform
- Filtering and convolution
- Matrix

DSPLIB Summary	
Component Type	Library
Install Package	DSPLIB
Install Directory	c674x_dsplib_<version>\
Project Type	CCS ^[1]
Endian Support	Big and Little
Library Name	dsplib.a674(COFF, little-endian) dsplib674.ae674 (ELF, little-endian)
Linker Path	<root_install_dir>\lib\
Linker Sections	N/A
Section Preference	N/A
Include Paths	<root_install_dir>\inc\ <root_install_dir>\packages\
Reference Guides	See docs under Install Directory
Support	BIOS E2e Forum ^[16]
Additional Resources	N/A
Downloads	DSPLIB Downloads ^[17]
License	BSD ^[5]

Image Library (IMGLIB)

IMGLIB is an optimized image/video processing library with kernels in the following functional categories:

- Compression & Decompression
- Image Analysis
- Image Filtering and Conversion

IMGLIB Summary	
Component Type	Library
Install Package	IMGLIB
Install Directory	c674x_imglib_<version>\
Project Type	CCS ^[18]
Endian Support	Little
Library Name	imglib.a64P(COFF, little-endian) imglib.ae64P(ELF little-endian)
Linker Path	<root_install_dir>\lib\
Linker Sections	N/A
Section Preference	N/A
Include Paths	<root_install_dir>\inc\ <root_install_dir>\packages\
Reference Guides	See docs under Install Directory
Support	BIOS E2e Forum ^[16]
Additional Resources	N/A
Downloads	IMGLIB Downloads ^[19]
License	BSD ^[5]

Floating Point Math Library (MATHLIB)

MATHLIB contains optimized versions of most commonly used floating point math routines contained in the RTS library. Kernels are offered in two flavors:

- Double precision floating point
- Single precision floating point

MATHLIB Summary	
Component Type	Library
Install Package	MATHLIB
Install Directory	mathlib_c674x_<version>\
Project Type	CCS ^[1]
Endian Support	Big and Little
Library Name	mathlib.a674(COFF, little-endian) mathlib.ae674 (ELF, little-endian)
Linker Path	<root_install_dir>\lib\
Linker Sections	N/A
Section Preference	N/A
Include Paths	<root_install_dir>\inc\ <root_install_dir>\packages\
Reference Guides	See docs under Install Directory
Support	BIOS E2e Forum ^[16]
Additional Resources	N/A
Downloads	MATHLIB Downloads ^[20]
License	BSD ^[5]

Demonstration Software

The BIOS C6SDK consist of demonstration software to illustrate device and software capabilities, benchmarks, and usage.

Face Detect Demo

Face detect algorithm demonstrates the capability of the device to capture images, process the image on DSP and then display the images on the output. The cxcore library of OpenCV is integrated for identifying the face. The input is captured with the standard Composite camera in the form of YUV images at 30fps. The images are passed to the DSP for processing the captured images. The output of the OpenCV algorithms provides the coordinates of the face identified. The Version 1.0 of OpenCV used is available in source here ^[21]. These images are then displayed on composite output.

Tools

Eclipse RTSC Tools (XDC)

RTSC is a C-based programming model for developing, delivering, and deploying Real-Time Software Components targeted for embedded platforms. The XDCtools product includes tooling and runtime elements for component-based programming using RTSC.

XDC Summary	
Component Type	Tools
Install Package	XDC
Install Directory	xdctools_xdctools_\
Project Type	Eclipse RTSC ^[4]
Endian Support	Little and Big
Library Name	The appropriate libraries are selected for your device and platform as set in the RTSC build properties for your project and based on the use module statements in your configuration.
Linker Path	The appropriate path is selected to the libraries for your device and platform as set in the RTSC build properties for your project.
Linker Sections	systemHeap
Section Preference	none
Include Paths	N/A
Reference Guides	See docs under Install Directory
Support	Technical Support
Additional Resources	Eclipse RTSC Home ^[4] Users Guide and Reference Manual ^[22]
Downloads	N/A
License	See XDC Manifest in the install directory

Build and Example Guide

The Build and Example Guide talks about setting up your build environment for C6SDK, how to build the various components and then walks you through a set of example programs that are designed to teach you how to start writing programs using the software development kit.

Setting up the Build Environment

To set up the build environment you need to have:

- Installed Code Composer Studio.
- Installed the C6SDK software.

The *Getting Started Guide* talks about how to do this.

Once CCS and BIOS-C6SDK are installed, they provide you with both Debug and Release versions of the demos, examples and components.

Creating a Target Configuration File

A Target Configuration File tells CCS how to connect to the EVM (they reference device specific files which have been supplied and contain information about the EVM, SoC and interface being used). Without one, you will not be able to load or debug applications over JTAG using CCS.

Refer to the 'Steps to create a CCS Configuration File' section of *Getting Started Guide* to get the detailed steps to create the configuration file.

XDS 510

You can find more detailed information about the XDS 510 interface here: <http://processors.wiki.ti.com/index.php/XDS510> ^[23]

XDS 560

You can find more detailed information about the XDS 560 interface here:

http://processors.wiki.ti.com/index.php/Xds_560

Building the Software

Build in Place vs. Build in Workspace

The C6SDK uses a "Build in Place" philosophy. This means you should not import the projects into the workspace. You can, but if you do, the projects may not re-build automatically and you may need to edit paths and other project settings to get them to build.

Note: It can be challenging to write a project that supports both build in place and build in workspace when the project is fairly rich and uses common source files (shared with other projects), etc.,.

Modifying a Library

- If you want to modify and re-build a library its best not to copy it into your workspace. We suggest building it "in-place". When you build in-place you will not need to change build macros and so forth. You will also not have to edit the example projects as they will already have the correct paths to the library.
- If you want to experiment with a library routine, debug it or try some new functionality, add the file to your project and use it there. Once you are done with it, if its a change you need to add then you can re-build it in the library.
- You may want to make a backup copy of any library before you begin modifying it. This will allow you to get to the original more easily should you need to do so.

Compiling C6SDK Demos

Facedetect

The Facedetect demo can be run on both EVMC6748 as well as LCDK. The pre-compiled Facedetect demo is provided in the `c6sdk/demos/facedetect/bin` folder.

There are two ways to compile it -

- Using Makefile
- Using CCS

Compiling facedetect using Makefiles

- Install a utility supporting make command, like cygwin which can be downloaded from <http://www.cygwin.com/install.html> ^[24]. Make sure to install the utility 'make' found in the category Devel in the installer, as it is

not a part of the default installation. You can cross check that 'make' is in the 'PATH' environment variable by running the command 'which make' on the cygwin terminal.

- The procedure to build the demo is the same for both EVMC6748 and LCDK.

Build Instructions for EVMC6748

1. Through the terminal, go to <BIOS-C6SDK INSTALL DIR>c6sdk_02_00_00_00/demos/facedetect/build/cgt
2. Open the makedefs file found in this location using a text editor
3. Check if the locations for the compiler, xdctools and SYS/BIOS on your system are the same as shown below. If not, edit the paths accordingly. (If there are spaces in the name of the folder, you need to write the paths as shown in the figure below.)

```
#
# Compiler, XDCTOOLS and SYS/BIOS path - edit!
#
CGT_DIR="C:/ti/ccsv5/tools/compiler/c6000"
XDCTOOLS_DIR=C:/Program\ Files/Texas\ Instruments/xdctools_3_23_00_32
SYSBIOS_DIR=C:/Program\ Files/Texas\ Instruments/bios_6_33_01_25
```

4. Now go to the directory evmC6748 through the terminal
5. To build both, Debug and Release modes, type 'make all' and hit enter. This will create the executable binary 'facedetect_evmC6748.out' in <BIOS-C6SDK INSTALL DIR>c6sdk_02_00_00_00\demos\facedetect\bin\cgt\evmC6748/{COMPILE MODE}.
6. If you want to build it only in Debug mode, type 'make debug' or 'make Debug'. Similarly type 'make release' or 'make Release' to build it only in Release mode. The executable binaries will be created in the path mentioned above.
7. To clean the project, type 'make clean' This will delete the executable binaries in both modes along with the object files.

Build instructions for LCDK

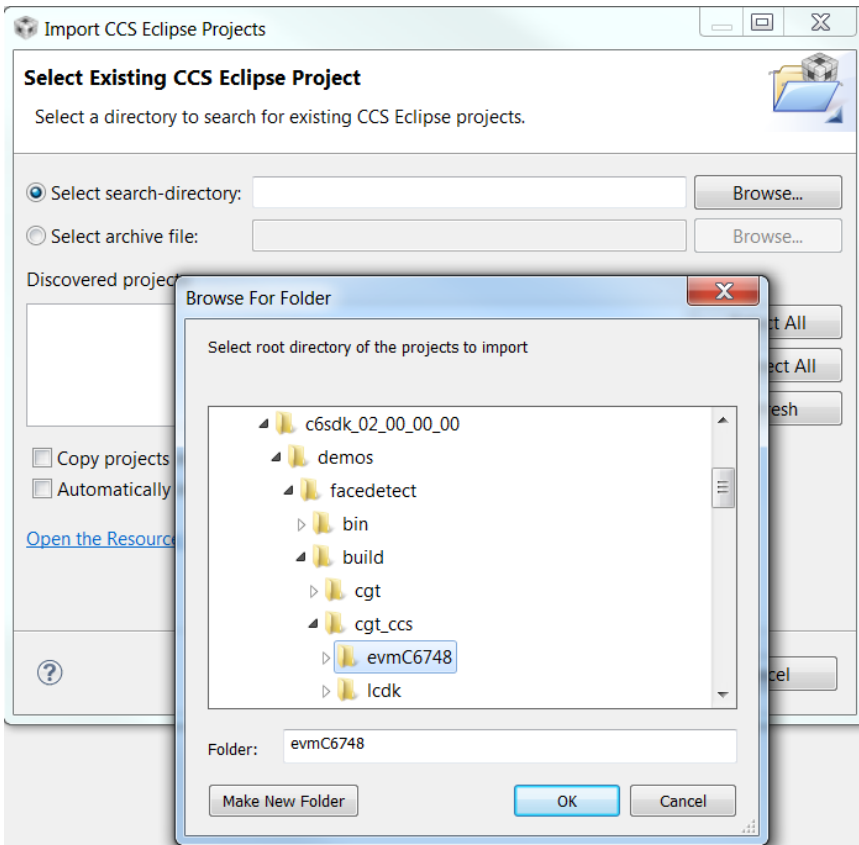
- To build the demo for LCDK, repeat steps 1-3 and now go to the lcdk directory through the terminal instead of step 4

Compiling facedetect using CCS

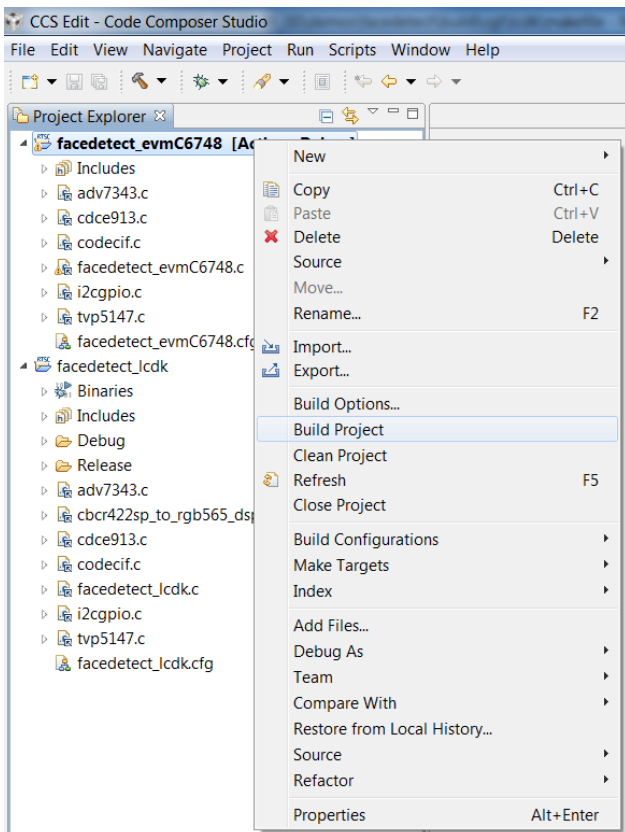
- The procedure to build the demo is the same for both EVMC6748 and LCDK other than importing it from different paths.

Building facedetect for EVMC6748

- Open CCS and goto Project -> Import Existing CCS/CCE Eclipse Projects and import project from <BIOS-C6SDK INSTALL DIR>c6sdk_02_00_00_00\demos\facedetect\build\cgt_ccs\evmC6748



- It should import the project facedetect_evmC6748. Click Finish. Note: Do not check copy projects into workspace
- Now you can build the demo by right clicking on the project name and selecting Build Project as shown below



- You can change the build configuration by going to Build Configurations seen above -> Set Active -> Release

- The executable binaries will be created in <BIOS-C6SDK INSTALL DIR>c6sdk_02_00_00_00\demos\facetect\bin\cgt_ccs\{EVM}\{COMPILE MODE}

Building facetect for LCDK

- To build the demo for LCDK, import the project from <BIOS-C6SDK INSTALL DIR>c6sdk_02_00_00_00\demos\facetect\build\cgt_ccs\lcdk and repeat the steps above. The demo integrates prebuild OpenCV functions for facetect.

OpenCV libraries are located at c6sdk_02_00_00_00\demos\facetect\bin

The following functions are used to detect the face from Open CV algorithms

```
Apply OpenCV haardetect algorithm
dsp_sequence = cvHaarDetectObjects(image, cascade, storage, 3, 2, 0,
cvSize(30,30));
```

Hardware Setup for demos

Face Detect Hardware setup

EVMC6748

- Refer *Getting Started Guide* for setting up the EVM.
- Connect XDS510 emulator and power on board.
- Connect the Composite camera input at J6 on composite video input port.
- Connect the Composite display out at J4 on Composite Video out port.

LCDK

- Refer *Getting Started Guide* for setting up the LCDK.
- Connect XDS510 emulator and power on board.
- Connect the Composite camera input at the J9 on composite video input port.
- Connect the LCD screen at J8 on the VGA connector on the LCDK.

Finger Print Hardware setup

LCDK

- Refer *Getting Started Guide* for setting up the LCDK.
- Connect XDS510 emulator and power on board.
- Connect a mini USB cable at J3 on the UART port with the other end going to the host computer at one of its USB ports.
- Now open a serial port terminal like Tera Term on the host computer and connect to the appropriate Serial port.
- Set the baud rate to 115200. (To do this in Tera Term, go to Setup -> Serial port and set the Baud rate as 115200 from the drop down menu)

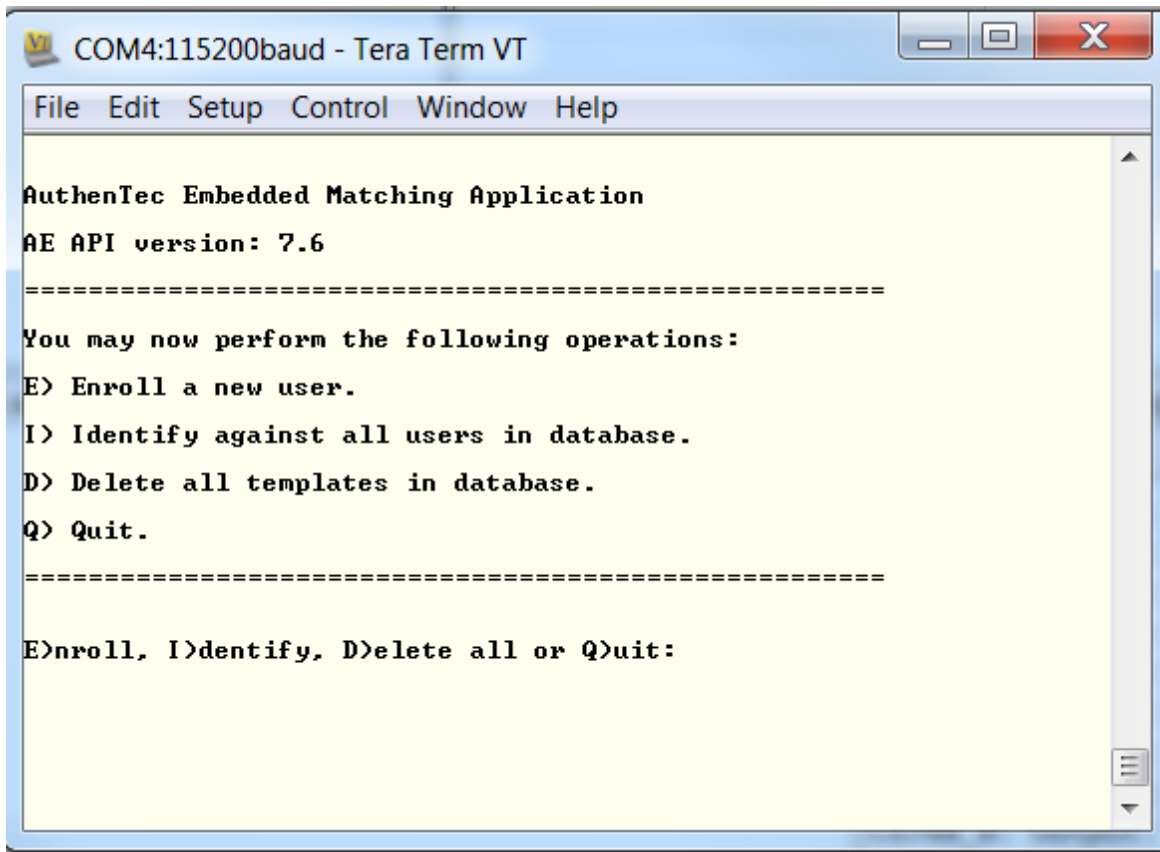
Executing the Demos

Running Facedetect Demo

- Ensure that the Hardware is setup as explained in Hardware Setup for Demos.
- Open CCSv5.1 and launch the target configuration file for TMS320C6748 or OMAPL138
- Go to Tools -> GEL Files
- Right click inside the GEL Files window and select Load GEL.
- To run it on EVMC6748, browse to the pdk_C6748 folder -> gel and select C6748.gel
- To run it on LCDK, select the C6748_LCDK gel file in the gel folder.
- The loaded gel file will be in GEL Files window.
- In the Debug window, Right click on "Spectrum Digital XDS510USB Emulator_0/C674X_0 [Non-Project Debug Session]" -> Connect Target (Make sure the EVMC6748/LCDK is ON before saying Connect Target)
- In case of EVMC6748, make sure DIP switches 5 and 8 of S7 are ON to avoid the " The target is being held in reset. The applied reset must be released before progressing." error.
- In the console window, you will find the PLL and DDR frequencies getting set by the GEL file.
- It is a good practice to reset the CPU by going to Run -> Reset -> CPU Reset
- Now go to Run -> Load -> Load Program
- Browse to c6sdk_02_00_00_00\demos\facedetect\bin\{COMPILATION_METHOD}\{EVM}\Release amd open facedetect_evmC6748.out
- Wait until the program is loaded.
- Now run the program by going to Run -> Resume. If your hardware is set up correctly, you should be able to see a box around faces captured by the camera. For best results, stand a few feet away from the camera and make sure the camera is placed in such a way that your face and the camera are exactly at the same level. This is needed as it is a frontal face detection demo. Also, for the same reason, make sure your head is not tilted with respect to the camera.

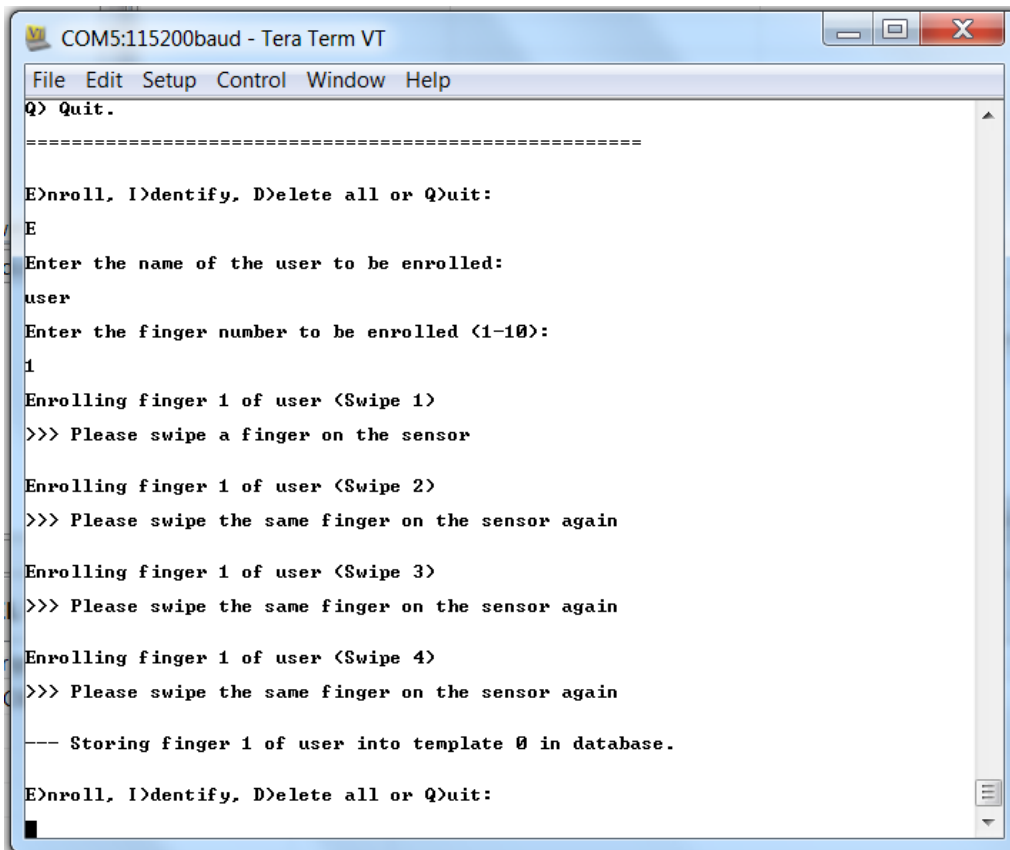
Running Fingerprint Demo

- Ensure that the Hardware is setup as explained in Hardware Setup for Demos.
Open CCSv5.1 and launch a target configuration file for TMS320C6748 or OMAPL138
- Go to Tools -> GEL Files
- Right click inside the GEL Files window and select Load GEL.
- Browse to the pdk_C6748 folder -> gel and select C6748_LCDK gel file.
- The loaded gel file will be in the GEL Files window.
- In the Debug window, Right click on "Spectrum Digital XDS510USB Emulator_0/C674X_0 [Non-Project Debug Session]" -> Connect Target (Make sure the LCDK is ON before saying Connect Target)
- In the console window, you will find the PLL and DDR frequencies getting set by the GEL file.
- It is a good practice to reset the CPU by going to Run -> Reset -> CPU Reset
- Now go to Run -> Load -> Load Program
- Browse to c6sdk_02_00_00_00\demos\fingerprint\bin and open fingerprint_lcdk.out
- Wait until the program is loaded.
- Now run the program by going to Run -> Resume
- The serial port terminal should look as below

A screenshot of a Tera Term VT window titled "COM4:115200baud - Tera Term VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The main text area displays the following content:

```
AuthenTec Embedded Matching Application
AE API version: 7.6
=====
You may now perform the following operations:
E> Enroll a new user.
I> Identify against all users in database.
D> Delete all templates in database.
Q> Quit.
=====
E>nroll, I>dentify, D>elete all or Q>uit:
```

- Type E to enroll a user and press enter.
- Enter the name of the user to be enrolled.
- Enter the finger number to be enrolled. Multiple fingers of the same user can be enrolled as different templates in the database.
- Now swipe the desired finger on the sensor. A video of the swipe sensor usage is available in the c6sdk_02_00_00_00\docs folder. Remember to swipe the finger in the direction going from the power cord to away from the board the other way but NOT in the direction going from or to the USB/UART port. You will have to swipe the finger 3-4 times to get a good image of the finger.
- Once the application finds a good image, it will store the finger image as a template in the database.
- The console should look like below

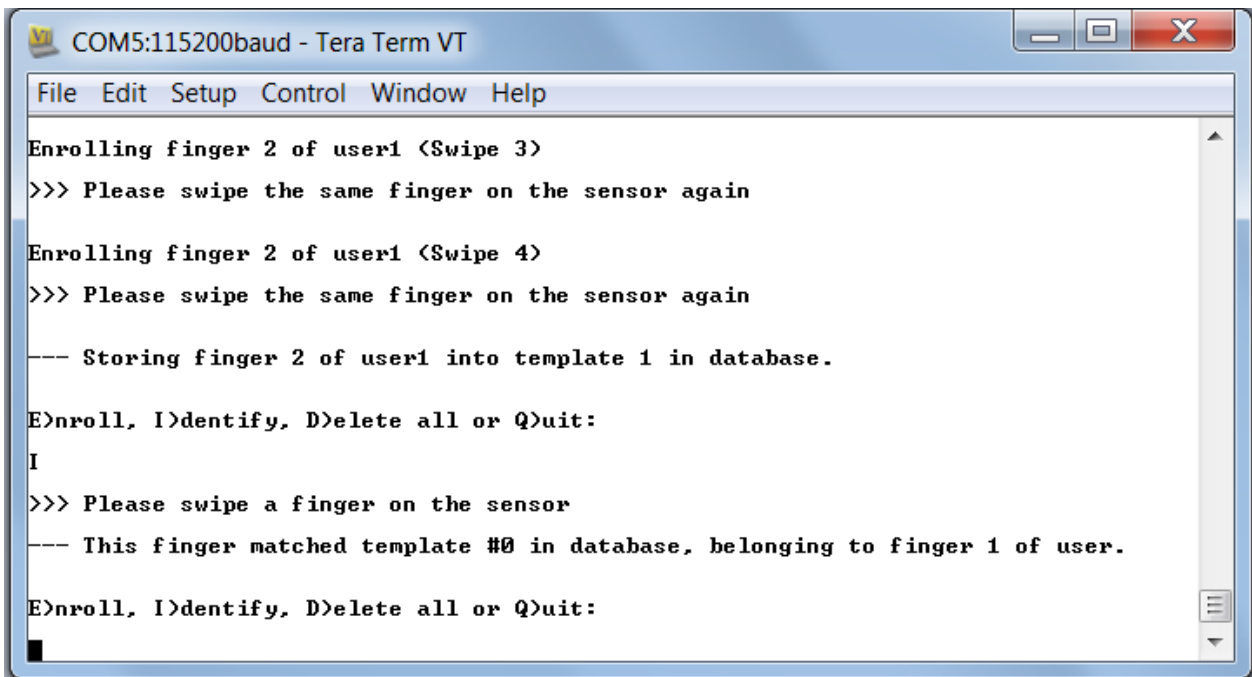


```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
Q> Quit.
-----
E)nroll, I)dentify, D)elete all or Q)uit:
E
Enter the name of the user to be enrolled:
user
Enter the finger number to be enrolled (1-10):
1
Enrolling finger 1 of user (Swipe 1)
>>> Please swipe a finger on the sensor
Enrolling finger 1 of user (Swipe 2)
>>> Please swipe the same finger on the sensor again
Enrolling finger 1 of user (Swipe 3)
>>> Please swipe the same finger on the sensor again
Enrolling finger 1 of user (Swipe 4)
>>> Please swipe the same finger on the sensor again
--- Storing finger 1 of user into template 0 in database.
E)nroll, I)dentify, D)elete all or Q)uit:

```

- You can enroll up to 10 users using the procedure above.
- To identify a fingerprint, type I in the console and hit enter.
- Swipe the finger to be identified.
- If the fingerprint exists in the database, the application will print the name and finger number of the identified user on the console; otherwise it will say finger did not match.



```

COM5:115200baud - Tera Term VT
File Edit Setup Control Window Help
Enrolling finger 2 of user1 (Swipe 3)
>>> Please swipe the same finger on the sensor again
Enrolling finger 2 of user1 (Swipe 4)
>>> Please swipe the same finger on the sensor again
--- Storing finger 2 of user1 into template 1 in database.
E)nroll, I)dentify, D)elete all or Q)uit:
I
>>> Please swipe a finger on the sensor
--- This finger matched template #0 in database, belonging to finger 1 of user.
E)nroll, I)dentify, D)elete all or Q)uit:

```

- You can delete all templates in the database by typing D on the console. It will ask for confirmation in case you select D mistakenly.
- You can stop the demo by selecting Q.

Booting and Flash

The Serial Boot and Flash Loading Utility for OMAP-L138/C6748 is detailed http://processors.wiki.ti.com/index.php/Serial_Boot_and_Flash_Loading_UTILITY_for_OMAP-L138

How to Flash Face detect demo on EVMC6748

This is a two step process. At first, create the AIS version of facedetect_evmC6748.out and then the AIS file is flashed using the OMAP-L138 FlashAndBootUtils.

NOTE - The demo can be flashed only on C6748 stand alone core and not on the one available in OMAPL138.

You can skip STEP 1 by using the AIS output file (facedetect_evmC6748.bin) provided in c6sdk folder -> demos -> flash -> evmC6748

STEP 1:

- Install the AISgen tool from the setup available in tools subfolder of the c6sdk folder
- Open 'AISgen for D800K008' and go to File -> Load Configuration
- Browse to c6sdk_02_00_00_00\demos\flash\evmC6748 and open EVMC6748_AISGen_Config.cfg
- Choose the DSP Application File which in our case would be facedetect_evmC6748.out found at c6sdk_02_00_00_00\demos\facedetect\bin\{COMPILATION_METHOD}\evmC6748\Release
- Choose where you want to create the AIS output file, say C:\ and provide a name for that file, say facedetect_evmC6748.bin i.e. C:\facedetect_evmC6748.bin
- Now hit Generate AIS. In a few seconds, facedetect_evmC6748.bin must be created in C:\. You may now close the AISgen tool.

STEP 2:

- Connect your PC to the EVM through a UART cable.
- Turn OFF the EVM.
- Turn DIP switches 7 and 8 of S7 (the one which is closer to the power switch) to ON position while 1-6 remain in the OFF position. Ensure switches 1-8 of S2 (the other DIP switch closer to JTAG) are in the OFF position
- To ensure you have selected the UART boot mode open a serial port terminal like Tera Term, set the baud rate to 115200 and turn ON the EVM. You should see a message called BOOT ME on the terminal each time you turn on the EVM. Now close your serial port terminal to flash the image on the chip.
- Find the COM Port Number being used by your PC. It can be found by going to Device Manager -> Ports or when you run Tera Term, it shows the Serial port available for connection which is the COM Port.
- Open cmd.exe on your PC
- cd to <BIOS-C6SDK INSTALL
DIR>c6sdk_02_00_00_00\tools\OMAP-L138_FlashAndBootUtils_2_36\OMAP-L138\GNU
- If there is an image flashed already, run the following command to erase it

```
sfh_OMAP-L138.exe -erase -targetType C6748 -flashType NAND -p
<COM_Port_Number>
```

where COM_Port_Number is the serial port being used say COM5.

- Run the following command

```
sfh_OMAP-L138.exe -flash_noubl -targetType C6748 -flashType NAND -p
<COM_Port_Number> "<Ais_File_Name>"
```

where COM_Port_Number is the serial port being used say COM5 and and Ais_File_Name is the file generated by AISgen tool, in our case C:\facedetect_evmC6748.bin

- Turn DIP switches 2 and 4 to ON position while others remain in the OFF position.
- To ensure you have selected the UART boot mode open a serial port terminal like Tera Term, set the baud rate to 115200 and turn ON the EVM. You should see a message called BOOT ME on the terminal each time you turn on the EVM. Now close your serial port terminal to flash the image on the chip.
- Find the COM Port Number being used by your PC. It can be found by going to Device Manager -> Ports or when you run Tera Term, it shows the Serial port available for connection which is the COM Port.
- Open cmd.exe on your PC
- cd to <BIOS-C6SDK INSTALL
DIR>c6sdk_02_00_00_00\tools\OMAP-L138_FlashAndBootUtils_2_36\OMAP-L138\GNU
- If there is an image flashed already, run the following command to erase it

```
sfh_OMAP-L138.exe -erase -targetType C6748_LCDK -flashType NAND -p
<COM_Port_Number>
```

where COM_Port_Number is the serial port being used say COM4

- Run the following command

```
sfh_OMAP-L138.exe -flash_noubl -targetType C6748_LCDK -flashType NAND
-p <COM_Port_Number> "<Ais_File_Name>"
```

where COM_Port_Number is the serial port being used say COM4 and Ais_File_Name is the file generated by AISgen tool, in our case C:\facedetect_lcdk.bin

- Now the utility waits for a BOOT ME message from the chip. So turn ON the LCDK or if its already ON, reset it. Once the BOOT ME message is received, the utility will start flashing to the LCDK.
- Wait till the flashing is complete. The command window will look as below

```
[TYPE] Single boot image
[BOOT IMAGE] C:\facedetect_lcdk.bin
[TARGET] C6748_LCDK
[DEVICE] NAND
[NAND Block] 1

Attempting to connect to device COM4...
Press any key to end this program at any time.

[AIS Parse]: Read magic word 0x41504954.
[AIS Parse]: Waiting For BOOTME... (power on or reset target now)
[AIS Parse]: BOOTME received!
[AIS Parse]: Performing Start-Word Sync...
[AIS Parse]: Performing Ping Opcode Sync...
[AIS Parse]: Processing command 0: 0x58535901.
[AIS Parse]: Performing Opcode Sync...
[AIS Parse]: Loading section...
[AIS Parse]: Loaded 1988-Byte section to address 0x1180000.
[AIS Parse]: Processing command 1: 0x58535901.
[AIS Parse]: Performing Opcode Sync...
[AIS Parse]: Loading section...
[AIS Parse]: Loaded 1364-Byte section to address 0x11804060.
[AIS Parse]: Processing command 2: 0x58535901.
[AIS Parse]: Performing Opcode Sync...
[AIS Parse]: Loading section...
[AIS Parse]: Loaded 20-Byte section to address 0x118052F0.
[AIS Parse]: Processing command 3: 0x58535901.
[AIS Parse]: Performing Opcode Sync...
[AIS Parse]: Loading section...
[AIS Parse]: Loaded 16-Byte section to address 0x11805304.
[AIS Parse]: Processing command 4: 0x58535906.
[AIS Parse]: Performing Opcode Sync...
[AIS Parse]: Performing jump and close...
[AIS Parse]: AIS complete. Jump to address 0x11804620.
[AIS Parse]: Waiting For DONE...
[AIS Parse]: Boot completed successfully.

Waiting for SFT on the OMAP-L138...

Flashing application C:\facedetect_lcdk.bin (1775808 bytes)

100% [ _____ ]
           Image data transmitted over UART.

100% [ _____ ]
           Application programming complete

Operation completed successfully.
```

- Turn OFF the LCDK by unplugging the power cord and turn switches 2,3 and 4 to ON while others remain in the OFF position.

- Turn ON the LCDK and wait for around 10 seconds. If the hardware is setup correctly, you should see the demo working as expected.

How to Flash Finger Print demo on LCDK

The procedure to flash the Finger Print demo is exactly the same as for flashing Face Detect demo on LCDK, only the file names would change.

The prebuilt AISGen output file 'fingerprint_lcdk.bin' can be found at c6sdk_02_00_00_00\demos\flash\lcdk. The executable file 'fingerprint_lcdk.out' can be found at c6sdk_02_00_00_00\demos\fingerprint\bin.

Uninstalling BIOS C6SDK

The BIOS C6SDK installer installs the un-installer in c6sdk_xx_00_00_## directory. The name of the un-installer is uninstall-bios_c6sdk_x.xx.xx.xx.exe.

Note: Some packages are installed as separate packages (e.g., SYS/BIOS, DSPLIB, IMGLIB, MATHLIB and XDCTOOLS) in the system. To uninstall these packages, please run the corresponding uninstaller.

Technical Support and Product Updates

Technical Support and Forums

For technical discussions and issues, please visit

- **C6000 Single Core DSP forum:** http://e2e.ti.com/support/dsp/tms320c6000_high_performance_dsps/default.aspx ^[25]
- **BIOS Embedded Software forum:** <http://e2e.ti.com/support/embedded/f/355.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 "BIOS C6SDK", the part number (e.g. "C6748") and additionally the component (e.g. "NDK").

Frequently Asked Questions and Tips

FAQs

I am not able to flash the demos on EVM C6748

Check the SOM version of your C6748. You can do that by launching a target configuration file for C6748 and connecting to the target. Load the OMAPL1x_debug gel from pdk_C6748_2_0_0_0\gel. In the CCS Debug view, go to Scripts -> Diagnostics -> Run_All. In the output window, check the BOOTROM info section. If the Silicon Revision is 1.0, you will not be able to flash the demos.

I have a C6748 Silicon Revision 2.0. Still the flashed demos do not work

Check the ROM ID of your device by launching a target configuration file for C6748 and connecting to the target. Load the OMAPL1x_debug gel from pdk_C6748_2_0_0_0\gel. In the CCS Debug view, go to Scripts -> Diagnostics -> Run_All. In the output window, check the BOOTROM info section. If the ROM ID is not d800k006, you cannot use the prebuilt flash images. You will have to create the flash image using AISGen Tool as mentioned in the Booting and Flash section. You will also have to modify the loaded EVMC6748_AISGen_Config configuration and change the Device Type to your specific version.

While flashing the demos, I get an error saying "This application failed to open the COM port"

Close any application accessing the serial port like Tera Term or any other serial port terminal and try again.

While importing the projects in CCS, I get an error saying "Imported projects cannot be managed"

You do not have Code Generation Tools 7.3.1 installed on your system.

I have CGT 7.3.1 installed on my computer still I get the "Imported projects cannot be managed" error

If it is installed, it is not being recognised by CCS. You can cross check that by going to Window -> Preferences -> Code Composer Studio -> Build -> Compilers. See if Ti v7.3.1 is listed in the Discovered tools section. You can add the path where CGT 7.3.1 is installed in the Tool discovery path by clicking the Add button on the right.

I cannot find a particular RTSC product in the Properties window even though I have installed it

If it is installed, it is not being recognised by CCS. You can cross check that by going to Window -> Preferences -> Code Composer Studio -> RTSC -> Products. See if that particular RTSC product is listed in the Discovered tools section. You can add the path where your product is installed in the Tool discovery path by clicking the Add button on the right.

I cannot build the examples in BIOS PSP inspite of having set the paths correctly in env.mk

You may not have installed C6SDK in C:/Program Files/ Texas Instruments. Some versions of Windows have different security settings in other drives making the files generated by BIOS PSP as read only and hence not accessible by the makefile.

I have installed C6SDK in C:/Program Files/ Texas Instruments, still cannot build the BIOSPSP modules/examples

Have you set the paths in env.mk file found at `pdk_C6748_2_0_0_0\biospsp_03_00_01_00\makerules` correctly? Have you set `INTERNAL_SW_ROOT` to `C:\Program Files\Texas Instruments\pdk_C6748_2_0_0_0\biospsp_03_00_01_00`? This won't work for two reasons. You need to have forward slashes in all the paths you edit in that file. The other reason is spaces in the path. The windows command prompt does not understand spaces. The work around is to write the short hand names of the folders that can be found by the command `dir /x`. So you should set `INTERNAL_SW_ROOT` as `C:/Progra~1/Texasi~1/biospsp_03_00_01_00`. The same holds for other paths in the file.

Tips

- While building the BIOS PSP, edit the env.mk file in `pdk_C6748_2_0_0_0\biospsp_03_00_01_00\makerules` and set the `INTERNAL_SW_ROOT`, `EXTERNAL_SW_ROOT`, `UTILS_INSTALL_DIR`, `CODEGEN_PATH_DSP` and `CODEGEN_PATH_DSPELF` as explained in the FAQ. Also, make sure that `bios_PATH`, `xdc_PATH`, `ipc_PATH`, `edma_PATH` and `bios_psp_PATH` make sense as all your packages may not be installed in the same `EXTERNAL_SW_ROOT` directory.

References

- [1] http://processors.wiki.ti.com/index.php/CCSv5_Getting_Started_Guide
- [2] https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/download.htm
- [3] http://software-dl.ti.com/dsps/dsps_public_sw/c6000/web/omap1138_starterware/latest/index_FDS.html
- [4] <http://www.eclipse.org/rtscl/>
- [5] <http://www.opensource.org/licenses/bsd-license.php>
- [6] <http://www.ti.com/lit/ug/sprufp2/sprufp2.pdf>
- [7] http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/ndk/index.html
- [8] http://software-dl.ti.com/dsps/dsps_registered_sw/sdo_sb/targetcontent/psp/edma3_ild/index.html
- [9] http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/bios/index.html
- [10] http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/bios/sysbios/index.html
- [11] <http://www-s.ti.com/sc/techlit/spru523.pdf>
- [12] <http://www-s.ti.com/sc/techlit/spru524.pdf>
- [13] <http://www-s.ti.com/sc/techlit/sprufp2.pdf>

- [14] http://processors.wiki.ti.com/index.php/Network_Developers_Kit_FAQ
 - [15] http://processors.wiki.ti.com/index.php/Rebuilding_the_NDK_Core
 - [16] <http://e2e.ti.com/support/embedded/f/355.aspx>
 - [17] http://software-dl.ti.com/sdoemb/sdoemb_public_sw/dsplib/latest/index_FDS.html
 - [18] http://processors.wiki.ti.com/index.php/CCSV4_Getting_Started_Guide
 - [19] http://software-dl.ti.com/sdoemb/sdoemb_public_sw/imglib/latest/index_FDS.html
 - [20] <http://focus.ti.com/docs/toolsw/folders/print/mathlib.html>
 - [21] <http://sourceforge.net/projects/opencvlibrary/files/opencv-win/>
 - [22] http://rtsc.eclipse.org/docs-tip/Main_Page
 - [23] <http://processors.wiki.ti.com/index.php/XDS510>
 - [24] <http://www.cygwin.com/install.html>
 - [25] http://e2e.ti.com/support/dsp/tms320c6000_high_performance_dsps/default.aspx
-

Article Sources and Contributors

BIOS C6SDK 2.0 User Guide *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?oldid=132977> *Contributors:* Asheeshb, Chintangala

Image Sources, Licenses and Contributors

Image:C6748 BIOS SDK stack.png *Source:* http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:C6748_BIOS_SDK_stack.png *License:* unknown *Contributors:* Asheeshb

Image:Makedefs path.png *Source:* http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Makedefs_path.png *License:* unknown *Contributors:* Chintangala

Image:CCS1.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:CCS1.png> *License:* unknown *Contributors:* Chintangala

Image:CCS2.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:CCS2.png> *License:* unknown *Contributors:* Chintangala

Image:Fingerprint1.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Fingerprint1.png> *License:* unknown *Contributors:* Chintangala

Image:Fingerprint2.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Fingerprint2.png> *License:* unknown *Contributors:* Asheeshb, Chintangala

Image:Fingerprint3.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Fingerprint3.png> *License:* unknown *Contributors:* Asheeshb, Chintangala

Image:Flash1.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Flash1.png> *License:* unknown *Contributors:* Chintangala

Image:Flash2.png *Source:* <http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Flash2.png> *License:* unknown *Contributors:* Chintangala

File:Light_bulb_icon.png *Source:* http://ap-fpdsp-swapps.dal.design.ti.com/index.php?title=File:Light_bulb_icon.png *License:* unknown *Contributors:* DanRinkes