

# TMS320C6455 Chip Support Library (CSL)

Version 3.00.10.02

Release Notes

06 September 2006

This release of CSL for TMS320C6455 contains peripheral programming (functional and register level) APIs for C6455 modules. The list of modules supported in this release is listed in later sections. This set of APIs provides peripheral abstraction that can be used by higher layers of software.

This release includes:

- Compiled library of supported CSL modules.
- Interrupt Controller (INTC) module
- API reference guides.
- Archived sources and build scripts.

Copyright © 2006 Texas Instruments Incorporated. All rights reserved.

Information in this document is subject to change without notice. Texas Instruments may have pending patent applications, trademarks, copyrights, or other intellectual property rights covering matter in this document. The furnishing of this documents is given for usage with Texas Instruments products only and does not give you any license to the intellectual property that might be contained within this document. Texas Instruments makes no implied or expressed warranties in this document and is not responsible for the products based from this document.

---

**TABLE OF CONTENTS**

---

1	System Requirements .....	3
2	Purpose of this Release.....	3
3	Release Contents .....	4
4	General information.....	6
5	Fixed in this release.....	13
6	Known Issues .....	20
7	Open Issues .....	22

## Release notes for TMS320C6455 CSL Version 3.00.10.02

### 1 System Requirements

- √ This version of CSL has been compiled with TI Codegen tools in Code Composer Studio version 3.3.13.2, Codegen compiler version v6.0.5, and Spectrum Digital Emulation driver for CCS 3.3.
- √ Compiler options used:
  - -mv6400+ -al -k

### 2 Purpose of this Release

This release is bug fix patch release. This also contains fixes for the bugs migrated from C6486 and C6488.

### 3 Release Contents

This release of CSL for TMS320C6455 contains functional and registers level APIs for modules listed in the table below to program the peripherals. This set of APIs provides peripheral abstraction that can be used by higher layers of software.

<b>MODULE</b>	<b>EXAMPLE PROVIDED</b>	<b>EXAMPLE TEST RESULTS ON EVM</b>	<b>REMARKS</b>
<i>DAT</i>	YES	PASS	<p>Only the register layer is provided.</p> <p>Only the register layer is provided</p> <p>Requires VDB to run this example</p> <ol style="list-style-type: none"> <li>1. Example for INTC module is available only in csl_c64xplus_intc package.</li> <li>2. INTC module is available as separate library, and should generally not be used with an embedded operating system with interrupt controller support, such as DSP/BIOS. Please refer to the API Reference Guide for more information on the INTC module.</li> <li>3. Modules using INTC should use "csl_c64xplus_intc.lib" (little endian) and "csl_c64xplus_intce.lib" (big endian).</li> </ol>
<i>DDR2</i>	YES	PASS	
<i>DEV</i>	NO	NA	
<i>EDMA</i>	YES	PASS	
<i>EMAC</i>	NO	NA	
<i>EMIFA</i>	YES	NA	
<i>GPIO</i>	YES	PASS	
<i>HPI</i>	YES	PASS	
<i>I2C</i>	YES	PASS	
<i>INTC</i>	YES	PASS	
<i>McBSP</i>	YES	PASS	<p>Only the register layer is provided.</p> <p>Only the register layer is provided.</p>
<i>MDIO</i>	NO	NA	
<i>PCI</i>	NO	NA	
<i>PLL</i>	YES	PASS	
<i>SRIO</i>	YES	PASS	
<i>TCP2</i>	YES	PASS	
<i>TIMER</i>	YES	PASS	
<i>UTOPIA2</i>	YES	NA	
<i>VCP2</i>	YES	PASS	
<i>BWMNGMT</i>	YES	PASS	

<b><i>CACHE</i></b>	YES	PASS	Only the register layer is provided.
<b><i>CFG</i></b>	YES	PASS	
<b><i>CHIP</i></b>	YES	PASS	
<b><i>ECTL</i></b>	NO	NA	
<b><i>IDMA</i></b>	YES	PASS	
<b><i>MEMPROT</i></b>	YES	PASS	
<b><i>PWRDWN</i></b>	YES	PASS	
<b><i>TSC</i></b>	YES	PASS	

#### 4 General information

##### Installation Information

This contains the API reference guide for the CSL and four directories:

	File name	Contents
API reference guide	C6455_CSL_APIREFERENCE.pdf	C6455 API Reference document
Product directory	csl_c6455	<ul style="list-style-type: none"> <li>✓ All header files.</li> <li>✓ CSL libraries in big endian and little endian mode.</li> <li>✓ Release example - Each device has its own example folders.</li> </ul>
Interrupt Controller (INTC) module directory	csl_c64xplus_intc	<ul style="list-style-type: none"> <li>✓ INTC module header files.</li> <li>✓ INTC module library file in big endian and little endian mode.</li> <li>✓ Release examples for INTC.</li> </ul> <p>This module should not be used with an embedded OS with interrupt controller support, such as DSP/BIOS. Please refer to the API Reference Guide for more information on the INTC module.</p>
C6455 Source directory	csl_c6455_src	<ul style="list-style-type: none"> <li>✓ All C6455 CSL source files.</li> <li>✓ All header files.</li> <li>✓ Build scripts to build the little endian and big endian libraries</li> </ul>
INTC Source directory	csl_c64xplus_intc_src	<ul style="list-style-type: none"> <li>✓ INTC CSL source files.</li> <li>✓ INTC CSL header files.</li> <li>✓ Build scripts to build the little endian and</li> </ul>

		big endian libraries
Release Notes	Releasenotes_C6455_v3_00_10_02.pdf	√ Releasenotes document
API Reference Document	TCI6455_CSL_APIREFERENCE.pdf	√ API Reference Guide

## Installation guidelines

The steps to be followed for installation of the release package are as follows:

1. Download the release zip file.
2. Unzip the files with command "unzip <zip file name>" at the command prompt. Alternatively, use the WinZip wizard to extract the files.

## Build guidelines

### Environmental Variable Settings Required for Building

1. Go to the directory where CCS is installed - i.e. <CCS\_INSTALLATION\_PATH>\ and run the "DosRun.bat" at command prompt.
2. Make sure the path to the gmake executable is in the DOS PATH environment variable.

If gmake path is not set, then path for 'gmake' as below

```
set path=%path%; <CCS_INSTALLATION_PATH>\bios_5_30\xdctools\
```

The steps to be followed to build the C6455 CSL library from the release are as follows:

1. At the command prompt go to the csl\_c6455\_src directory
2. Compile the code with the batch file given. Set the environment variable "CSL3X\_ROOT\_DIR = <folder path>" in your system. Different batch files for creating libraries for the devices are listed below. It will be found in following path csl\_c6455\_src\.
3. To clean the built object files invoke the corresponding batch file with the argument cleanall.

```
.\build_c6455.bat cleanall
```

Library	Batch file to be invoked	Library name
C6455 CSL	build_c6455 (little endian)	csl_c6455.lib
	build_c6455e (big endian)	csl_c6455e.lib

The steps to be followed to build the INTC CSL library from the release are as follows:

1. At the command prompt go to the csl\_c64xplus\_intc\_src directory

2. Compile the code with the batch file given. Set the environment variable "CSL3X\_ROOT\_DIR = <folder path>" in your system. Different batch files for creating libraries for the devices are listed below. It will be found in following path csl\_c64xplus\_intc\_src\  
.
3. To clean the built object files invoke the corresponding batch file with the argument cleanall.

.\build\_c64xplus\_intc.bat cleanall

Library	Batch file to be invoked	Library name
INTC CSL	build_c64xplus_intc (little endian)	csl_c64xplus_intc.lib
	build_c64xplus_intce (big endian)	csl_c64xplus_intce.lib

## Directory structure

The directory structures of TCI6482 package is as shown below:

```

Installation Package
|
|___cs1_c6455
|___cs1_c6455_src
|___cs1_c64xplus_intc
|___cs1_c64xplus_intc_src
|___Releasenotes_TCI6455_v3_00_10_02.pdf
|___TCI6455_CSL_APIREFERENCE.pdf
  
```

The directory structures of C6455 product directory package is as shown below:

```

cs1_c6455
|___example à
|           |___edma
|           |___i2c
|           |___mcbasp
|           |___ddr2
|           |___emifa
|           |___vcp2
|           |
|           |
|           |___tcp2
|___inc      CSL module Header files
|___lib
|           |___cs1_c6455.lib (little endian CSL library)
|           |___cs1_c6455e.lib (big endian CSL library)
|           |___cs1_c6454.lib (little endian CSL library)
|           |___cs1_c6454e.lib (big endian CSL library)
  
```

The directory structure of INTC library release is as shown below:

```
csl_C64xplus_intc
|
|_____ example          Example for INTC module
|
|_____ inc             INTC module Header files
|
|_____ lib
|           |_____ csl_c64xplus_intc.lib (little endian INTC CSL library)
|           |_____ csl_c64xplus_intce.lib (big endian INTC CSL library)
```

The directory structure of C6455 CSL source directory is as shown below:

```
csl_c6455_src
|
|_____ inc ALL Header files
|
|_____ src
|           |_____ chip
|           |_____ common
|           |_____ edma
|           |_____ .
|           |_____ .
|           |_____ mcbasp
|_____ build_c6455_bc.bat (to build little endian library)
|_____ build_c6455e_bc.bat (to build big endian library)
|_____ Makefile
|_____ Makefile.inc
|_____ csl_c6455.lib (little endian CSL library)
|_____ csl_c6455e.lib (big endian CSL library)
|_____ csl_c6454.lib (little endian CSL library)
|_____ csl_c6454e.lib (big endian CSL library)
```

The directory structure of INTC CSL source directory is as shown below:

```
cs1_c64xplus_intc_src
|_____inc INTC Header files
|_____src
|           |_____intc
|           |_____common
|_____ build_c64xplus_intc.bat (to build little endian library)
|_____ build_c64xplus_intce.bat (to build big endian library)
|_____ Makefile
|_____ Makefile.inc
|_____ cs1_c64xplus_intc.lib (little endian INTC CSL library)
|_____ cs1_c64xplus_intce.lib (big endian INTC CSL library)
```

Label in clearcase

C6455\_CSL\_BASE\_REL\_03\_00\_10\_02

### Examples

The peripheral examples are designed to run on the TMS320C6455 EVM. Examples may have hardware dependencies that prevent them from running to completion on the CCS Simulator.

The example for EMIFA can be run by connecting the external memory device. The peripherals EMIFA and UTOPIA examples can be run using the VDB.

## 5 Fixed in this release

PSG00000810	Add support for C6454 device in C6455 CSL
Release Note	
This has been fixed.	

PSG00000873	Documentation Errors And Missing Data in Release Notes
Release Note	
This has been fixed.	

PSG00000956	EDMA3 symbol name changes
Release Note	
This has been fixed.	

PSG00000975	General: GetBaseAddress API for some of the modules not handling invalid pointer for base address.
Release Note	
This has been fixed.	

PSG00001009	SRIO: When the API srioHwSetupRaw() is called with valid arguments, the system hangs.
Release Note	
This has been fixed.	

PSG00001011	SRIO: When the API srioHwSetup() is called with default arguments, the system hangs.
Release Note	
This has been fixed.	

PSG00001387	DDR: No Need to Unlock PERLOCK to access PERCFG1 in Example
Release Note	
This has been fixed.	

PSG00001390	SRIO External Loopback example should be tested on EVM
Release Note	
This has been fixed.	

PSG00001472	SRIO: Documentation Errors
Release Note	
This has been fixed.	

PSG00001473	CACHE: Documentation Errors
Release Note	
This has been fixed.	

PSG00001475	PLL:CSL_pllCwControl() API for PLL instance 2 is failing
Release Note	
This has been fixed.	

PSG00001480	Chip register enums incorrectly defined
Release Note	
This has been fixed.	

PSG00001483	PWRDWN: The pwrdownOpen() API checks for invalid object pointer but does not return correct status.
Release Note	
This has been fixed.	

PSG00001488	EMAC - MDIO: EMAC and MDIO Documentation Issue
Release Note	
This has been fixed.	

PSG00001489	PWRDWN: Documentation error in the API reference Guide
Release Note	
This has been fixed.	

PSG00001493	MEMPROT: Reset values given wrongly in the register layer header files.
Release Note	
This has been fixed.	

PSG00001495	Release Notes Issues
Release Note	
This has been fixed.	

PSG00001496	Himalaya EDMA3 not up to date with changes made for Faraday CSL
Release Note	
This has been fixed.	

PSG00001507	VCP2 example and usage issues with c6455 simulator
Release Note	
This has been fixed.	

PSG00001513	CHIP: TSR read value is different from the expected value
Release Note	
This has been fixed.	

PSG00001518	CSL 03_00_10_01 Issue
Release Note	
This has been fixed.	

PSG00001540	timer lo and hi are not simulataneously enabled
Release Note	
This has been fixed.	

PSG00001543	EDMA3 References to Queue Entry Registers
Release Note	
This has been fixed.	

PSG00001578	Add McBSP-EDMA Example
Release Note	
This has been fixed.	

PSG00001692	API Reference Guide Missing Functional Layer API Summary
Release Note	
This has been fixed.	

PSG00001695	TCP2 can't supports CDMA2000 frame length more than 10000
Release Note	
This has been fixed.	

PSG00001707	Add specialized INTC APIs to API Reference Guide
Release Note	
This has been fixed.	

PSG00001708	Chapter 1 of API Reference Guide Needs Enhancement
Release Note	
This has been fixed.	

PSG00001741	Documentation: Clarify INTC module usage for DSP/BIOS & OS users
Release Note	
This has been fixed.	

PSG00001747	CHIP: Documentation Issue in API Reference Guide
Release Note	
This has been fixed.	

PSG00001787	CSL_SrioAddrSelect typedef enum is incorrect in csl_srio.h
Release Note	
This has been fixed.	

PSG00001823	PDC: The pdcHwControl() API fails to return a valid error when invalid parameters are passed to it
Release Note	
This has been fixed.	

PSG00001829	PDC: The API pdcGetBaseAddress() API does not return valid errors with invalid instance or invalid parameters
Release Note	
This has been fixed.	

PSG00001835	BWMNGMT: Discrepancies in the API reference guide for Tomahawk
Release Note	
This has been fixed.	

PSG00001837	Memprot: Discrepancies in the API reference guide for Memprot module
Release Note	
This has been fixed.	

PSG00001838	EDMA: Documentation Issue
Release Note	
This has been fixed.	

PSG00001839	SRIO: The srioOpen() API fails when passed with invalid parameters
-------------	--

Release Note  
This has been fixed.

PSG00001840	SRIO: GetHwSetup() API does not return valid error with invalid parameters passed to it
-------------	---

Release Note  
This has been fixed.

PSG00001841	SRIO: GetHwStatus() API does not return valid error with invalid parameters passed to it.
-------------	---

Release Note  
This has been fixed.

PSG00001842	INTC:CSL_intcExcepAllDisable(),CSL_intcExcepAllRestore(), CSL_intcExcepAllStatus() and CSL_intcExcepAllEnable() return CSL_SOK for invalid inputs
-------------	---

Release Note  
This has been fixed.

PSG00001846	Incomplete build procedure for library in release notes
-------------	---

Release Note  
This has been fixed.

PSG00001847	Known issues in CSL releases recorded in the Release Notes MUST include the MR numbers
-------------	--

Release Note  
This has been fixed.

PSG00001848	INTC: API Reference Guide Issues
-------------	----------------------------------

Release Note  
This has been fixed.

PSG00001849	EDMA: CSL_edma3ccGetModuleBaseAddr() API does not return error with invalid parameters passed to it.
Release Note	
This has been fixed.	

PSG00001850	EDMA: Documentation Issues in API Reference Guide
Release Note	
This has been fixed.	

PSG00001855	DDR2: Inconsistencies in the API reference guide for DDR2
Release Note	
This has been fixed.	

## 6 Known Issues

PSG00001309	EMAC: 100Mbps Full Duplex in Phy loopback mode, this is not working.
Release Note	
None	
Workaround (If one exists)	
None	

PSG00001358	VCP2: Offset of VCP decisions read FIFO register(VCPRDECS) is mentioned wrong in the User guide
Release Note	
None	
Workaround (If one exists)	
None	

PSG00001359	SRIO:The default values of the certain SRIO registers mentioned in the SRIO datasheet are "undefined"
Release Note	
None	
Workaround (If one exists)	
None	

PSG00001361	TIMER: Timer user guide mention timer internal clock frequency as SYSCLK2
Release Note	
None	
Workaround (If one exists)	
None	

PSG00000835	TCP2_getMaxMinErr() does not return error status
Release Note	None
Workaround (If one exists)	None

## 7 Open Issues

PSG00000914	CRCLLEN field of TCPIIC4 register in TCP Module can be set to value greater than 32
Release Note None	
Workaround (If one exists) None	

PSG00000924	For PLLDIV4 and PLLDIV5 the divider configuration is failing
Release Note None	
Workaround (If one exists) None	

PSG00000934	VCP2_emuEnable() API does not work
Release Note None	
Workaround (If one exists) None	

PSG00001184	GPIO: More example required to explain the usage of a pin configured as an input for generating interrupts or EDMA events.
Release Note None	
Workaround (If one exists) None	

PSG00001479	C6455 CSL write to PLL1 PREDIV and DDR2 BPRI O register
Release Note	
None	
Workaround (If one exists)	
None	

PSG00001484	PLL C:No enum for selecting PREDIV like CSL_PLLC_DIVSEL_PREDIV
Release Note	
None	
Workaround (If one exists)	
None	

PSG00001536	Add examples for all modules, even register-layer-only
Release Note	
None	
Workaround (If one exists)	
None	