Application Report
SPNA121A–September 2011

How to Create A HalCoGen Based Project For CCSv4.x

ABSTRACT

HalCoGen is a GUI-based initialization, configuration and driver code generator for TI's Hercules™ ARM® Safety MCUs. HalCoGen's editing feature provides a strong motivation to be able to regenerate source code and build/debug the source through Code Composer Studio™ IDE in the same working folder.

Contents

List of Figures

1 Motivation

HalCoGen is able to maintain edits that you make to the generated source code, as long as these edits appear between comments *USER CODE BEGIN* and *USER CODE END*.

For example, by default HalCoGen generates the empty main() function (in file ‘sys_main.c’) as shown in Example 1.

Example 1. The Function main() as Initially Generated by HalCoGen

```
/* USER CODE BEGIN (2) */
/* USER CODE END */

void main(void)
{
/* USER CODE BEGIN (3) */
/* USER CODE END */
}
```

Any code that you add to main() between the comments *USER CODE BEGIN (x)* and *USER CODE END* will be maintained even if you make another pass through HalCoGen and regenerate code after having changed various settings.

You can edit sys_main.c through the Code Composer Studio IDE, and add functionality.

Example 2. User Additions to main()

```
/* USER CODE BEGIN (2) */
#include <stdio.h>             /* Correct */
#include <gio.h>               /* Correct */
/* USER CODE END */
#include <het.h>               /* Incorrect */

void main(void)
{
      hetInit();               /* Incorrect */
/* USER CODE BEGIN (3) */
      gioInit();               /* Correct */
/* USER CODE END */
}
```

Suppose you build the code, load it into a Hercules TMS570 or RM4 device, start a debugging session, and then decide that you need to change one of the PLL settings in HalCoGen.

You can go back to the HalCoGen GUI, make changes to the PLL settings, and regenerate the code. Any code additions that you made in the areas marked for user additions will be preserved through the regeneration process. As Example 3 illustrates, as long as you place the code in the correct location (between *USER CODE BEGIN* and *USER CODE END* comments); then the code changes are preserved through a cycle of regeneration.

Example 3. User Modified Code After Regeneration By HalCoGen

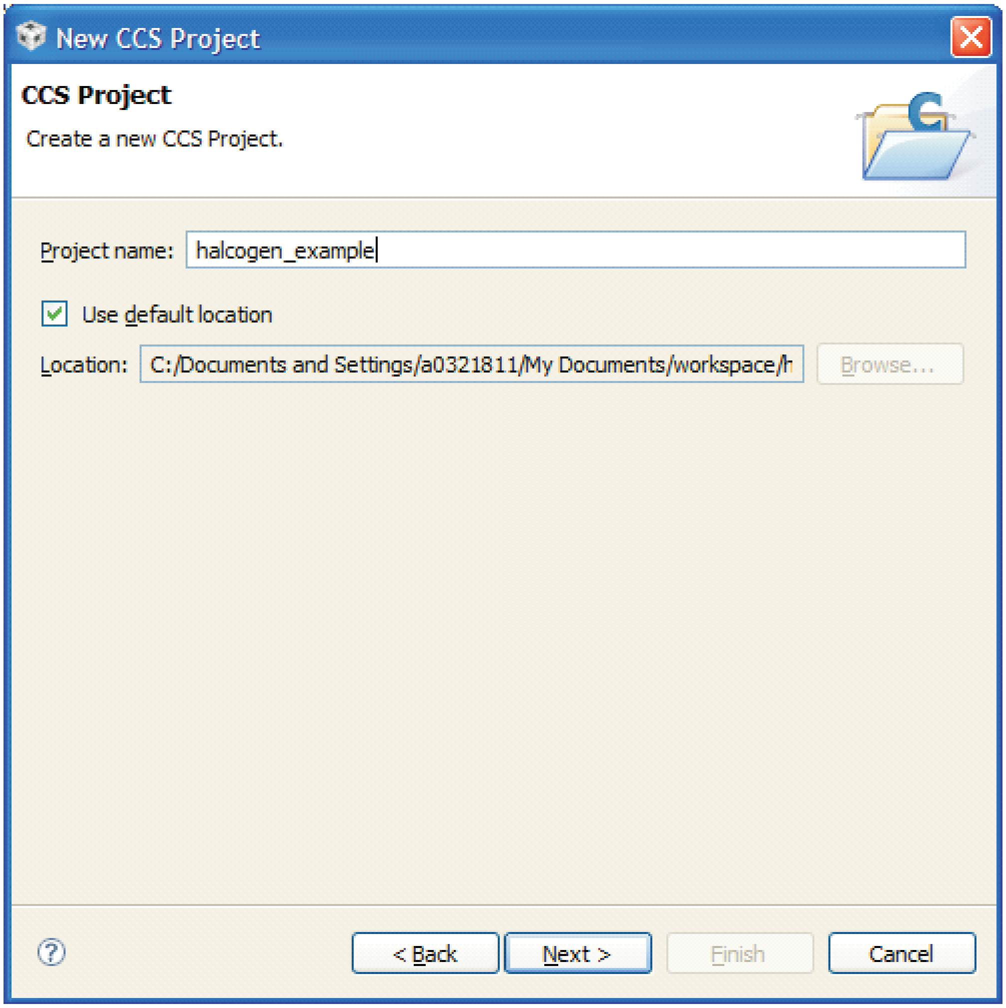
```
/* USER CODE BEGIN (2) */
#include <stdio.h>             /* Correct */
#include <gio.h>               /* Correct */
/* USER CODE END */

void main(void)
{
/* USER CODE BEGIN (3) */
      gioInit();               /* Correct */
/* USER CODE END */
}
```

Example 3 should provide motivation as to why one would want to have HalCoGen share a project directory with a Code Composer Studio project. Any updates made through HalCoGen would immediately appear in the Code Composer Studio project whenever the code is regenerated. And any edits made to the same files in the Code Composer Studio IDE will be retained through a cycle of HalCoGen regeneration as long as the code is added in areas specified for user code.

2 Creating a Code Composer Studio Project

This section explains how to configure both HalCoGen and Code Composer Studio to share the same project folder.

1. Create a project in Code Composer Studio.
2. Select File → New → CCS Project, from the Code Composer Studio menu bar, after starting the Code Composer Studio IDE. A window will open asking you where to create the Code Composer Studio project. In this case, create the project in the existing workspace, and name the project `halcogen_example` as shown in Figure 1.



New CCS Project

CCS Project

Create a new CCS Project.

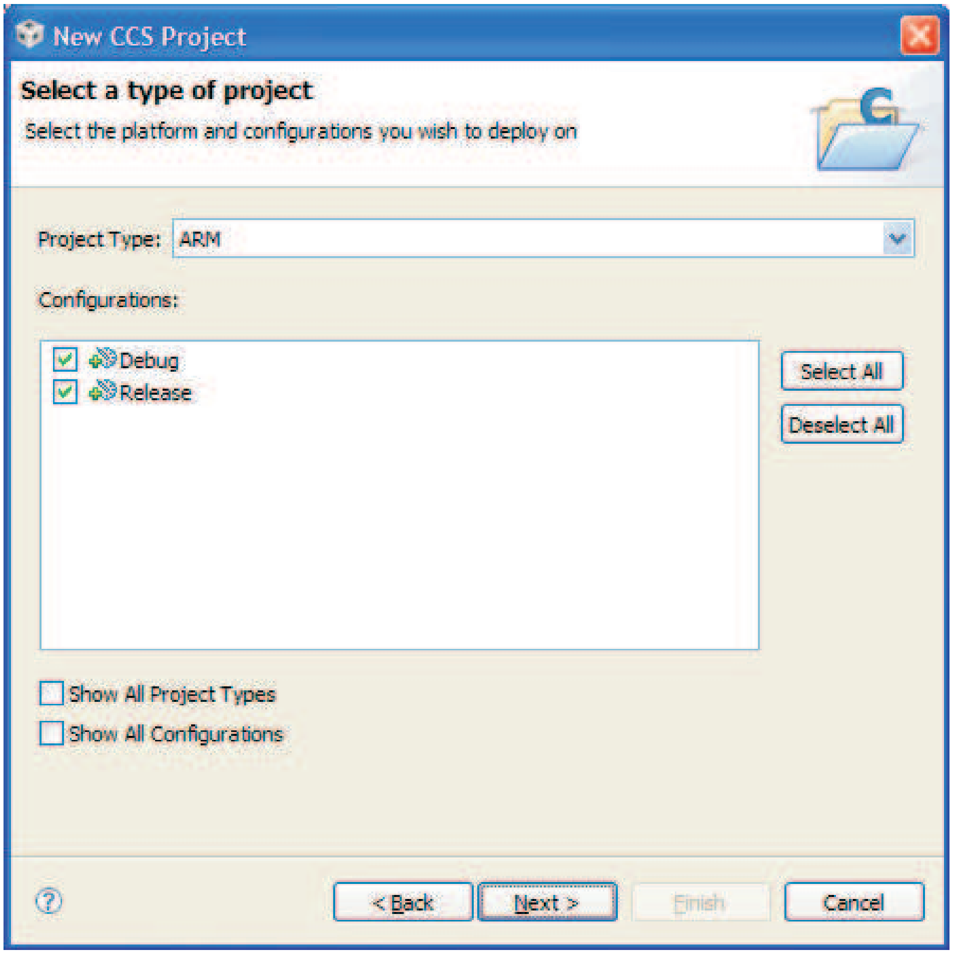
Project name: halcogen_example

☑ Use default location

Location: C:/Documents and Settings/a0321811/My Documents/workspace/h

Figure 1. Create a New Code Composer Studio Project

3. Choose ARM, as illustrated in Figure 2, when the next window asks you what type of project to create.



New CCS Project

Select a type of project

Select the platform and configurations you wish to deploy on

Project Type: ARM

Configurations:

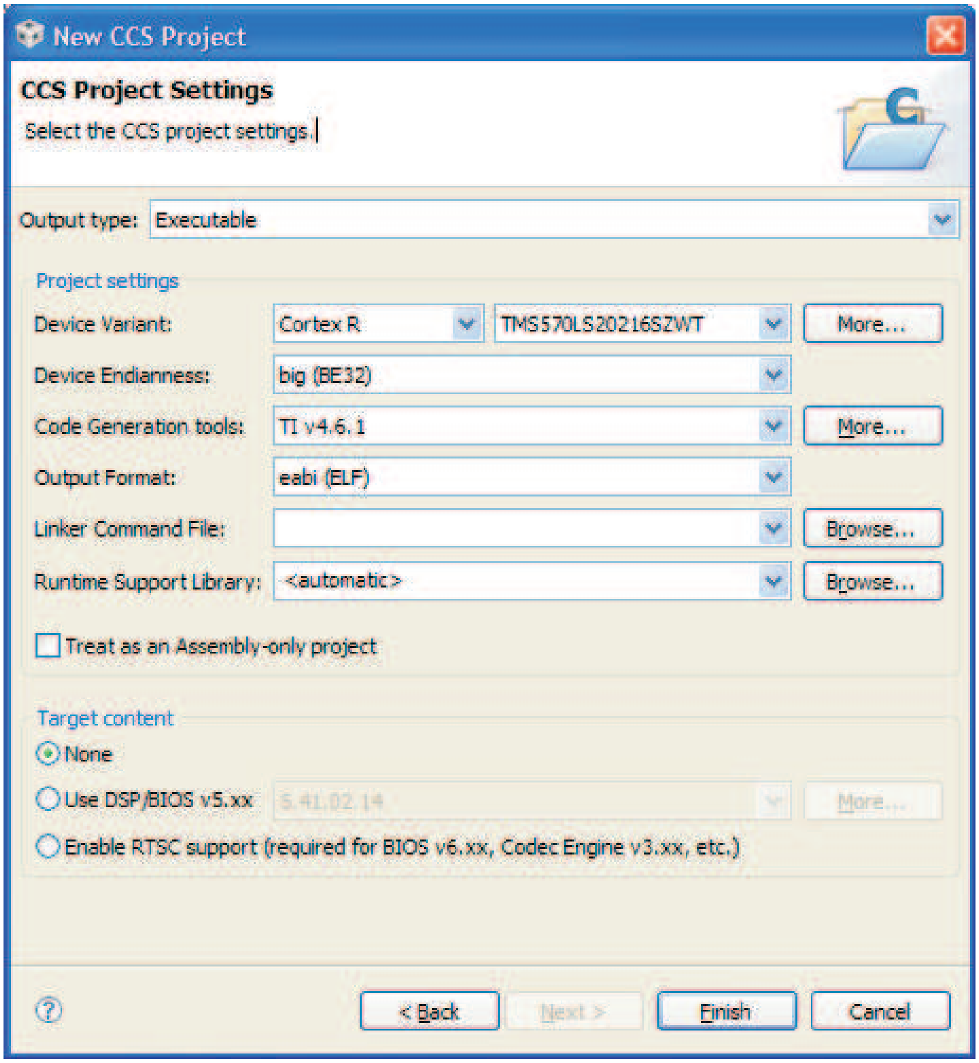
☑ Debug
☑ Release

□ Show All Project Types
□ Show All Configurations

Figure 2. Second Step Creating a New Code Composer Studio Project

4. Skip the next step that asks if there are any sub-projects to include; do not select any sub-projects.

5. Create the project using the template of the appropriate RM4, TMS470M or TMS570 device. This final step in project creation is important for HalCoGen projects. Settings for the TMS570LS20216SZWT are illustrated in Figure 3.



New CCS Project

CCS Project Settings

Select the CCS project settings.

Output type: Executable

Project settings

Device Variant: Cortex R TMS570LS20216SZWT

Device Endianness: big (BE32)

Code Generation tools: TI v4.6.1

Output Format: eabi (ELF)

Linker Command File:

Runtime Support Library: <automatic>

□ Treat as an Assembly-only project

Target content

● None

○ Use DSP/BIOS v5.xx 5.41.02.14

○ Enable RTSC support (required for BIOS v6.xx, Codec Engine v3.xx, etc.)

Figure 3. Final Step in Code Composer Studio Project Creation

6. Select Finish and a Code Composer Studio project will be created in the folder …/`workspace/halcogen_example`.
7. Create a HalCoGen project that will generate source files directly into this folder.

3 Creating a HalCoGen Project

Once the Code Composer Studio project has been created, it is time to create a HalCoGen project that generates code directly into the Code Composer Studio Project directory.

1. Start the HalCoGen program, which will bring up a startup screen as illustrated in Figure 4.

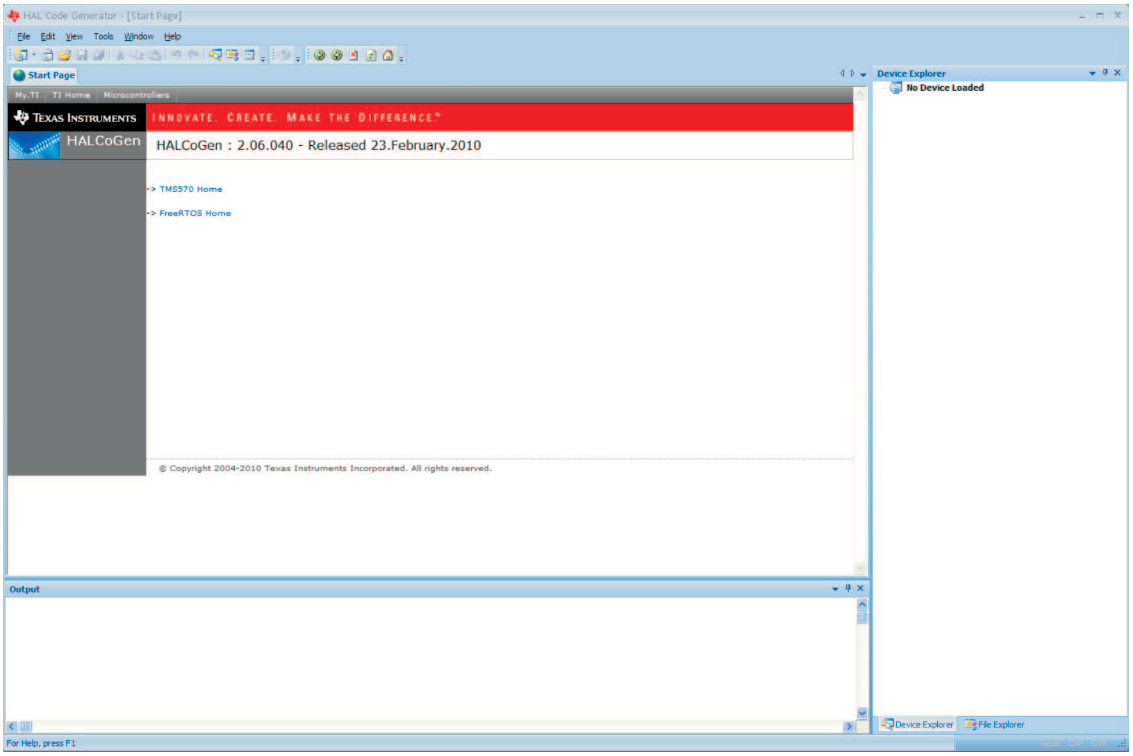

HAL Code Generator - [Start Page]
File Edit View Tools Window Help
Start Page
My.TI
TI Home
Microcontrollers
TEXAS INSTRUMENTS
INNOVATE. CREATE. MAKE THE DIFFERENCE.™
HALCoGen
HALCoGen : 2.06.040 - Released 23.February.2010
-> TMS570 Home
-> FreeRTOS Home
© Copyright 2004-2010 Texas Instruments Incorporated. All rights reserved.
Device Explorer
No Device Loaded
Output
Device Explorer
File Explorer
For Help, press F1

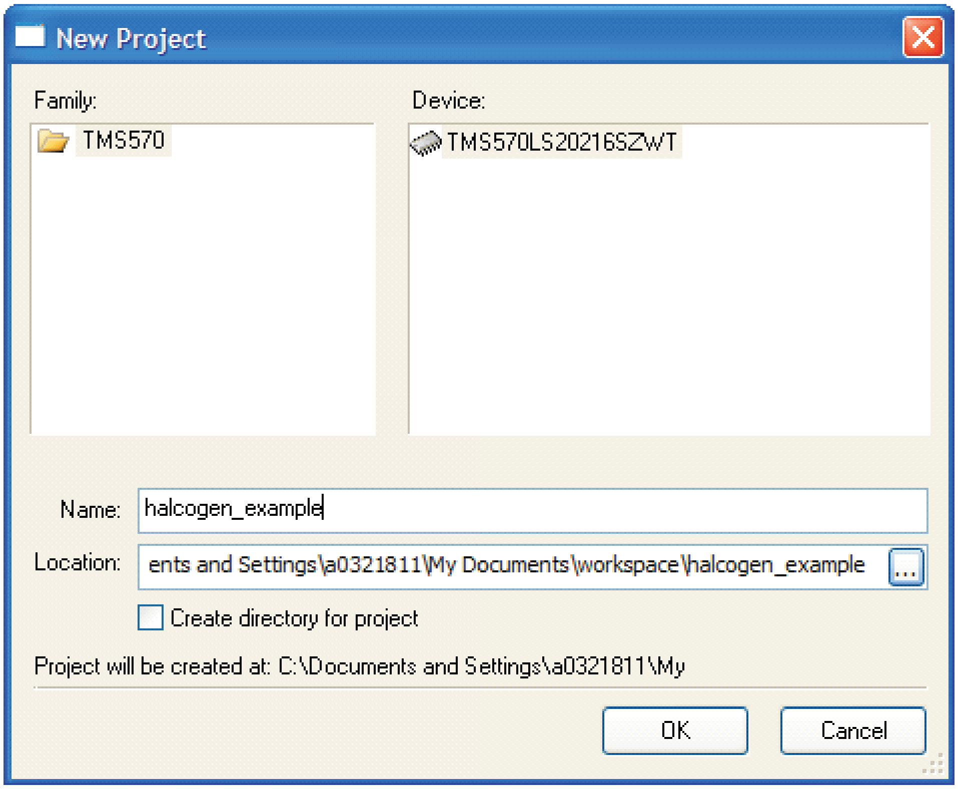
Figure 4. HalCoGen Startup With No Project

2. Select File → New → Project from the HalCoGen startup screen. A window appears asking you to select a device type, a working folder, and to select a name for the project.

   Figure 5 shows how you would create the new HalCoGen project so that code is generated directly into the Code Composer Studio project directory. In this case, the Code Composer Studio project directory was …`/workspace/halcogen_example`.

   The corresponding HalCoGen project should be created:

- With the same TMS570 device as was selected when creating the Code Composer Studio Project
- With any 'name' – although in this case the HalCoGen project name was chosen to match the Code Composer Studio Project name `halcogen_example`.
- With location matching the top level Code Composer Studio Project Folder: …`/workspace/halcogen_example`.
- With *Create directory for project* unchecked



New Project

Family: TMS570

Device: TMS570LS20216SZWT

Name: halcogen_example

Location: ents and Settings\a0321811\My Documents\workspace\halcogen_example

□ Create directory for project

Project will be created at: C:\Documents and Settings\a0321811\My

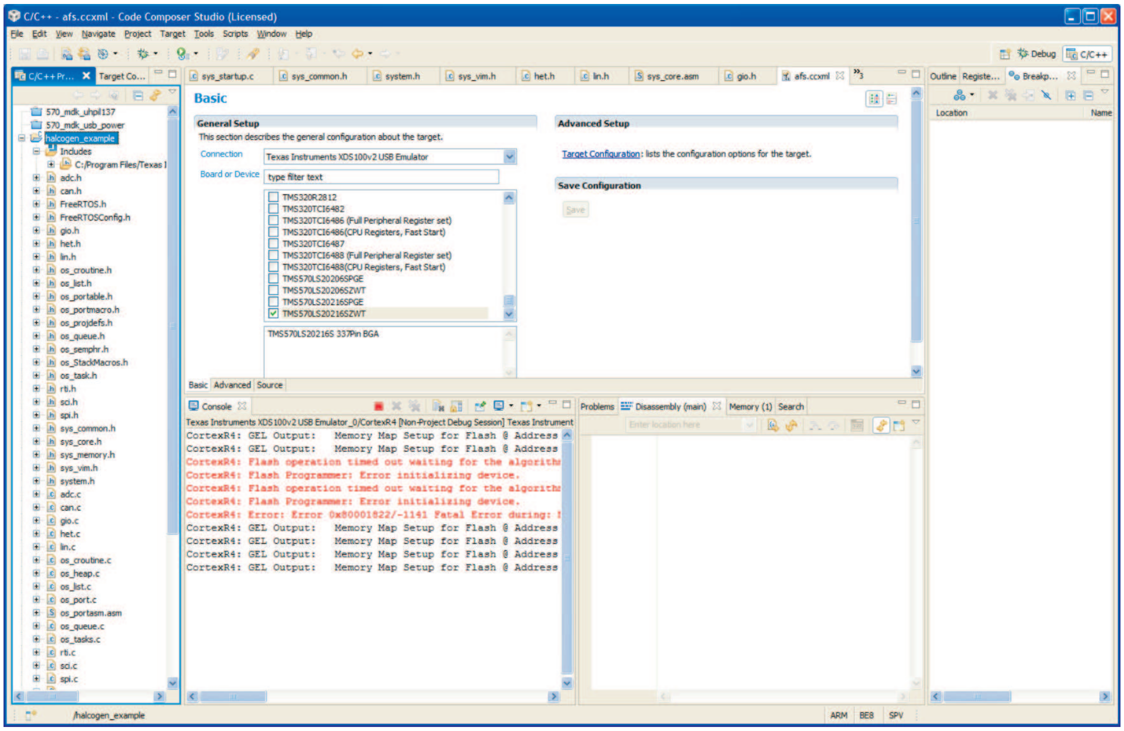
Figure 5. Creating a HalCoGen Project Sharing the Code Composer Studio Project Folder

That’s it! Now you have a pair of HalCoGen and Code Composer Studio projects that are set up to allow rapid iteration between the environments.

4 Generate the Code – Check the Code Composer Studio Project

At this point, you should be able to generate code from HalCoGen (you can try this immediately after creating the project, without changing any of the default settings.)

If everything is setup correctly, when you look at the corresponding Code Composer Studio project, you will see that all of the source files generated by HalCoGen have automatically been added to the Code Composer Studio project, as shown in the left hand pane of the Code Composer Studio project view (Figure 6)



Basic

General Setup

This section describes the general configuration about the target.

Connection: Texas Instruments XDS100v2 USB Emulator

Board or Device: type filter text

- □ TMS320R2812
- □ TMS320TCI6482
- □ TMS320TCI6486 (Full Peripheral Register set)
- □ TMS320TCI6486(CPU Registers, Fast Start)
- □ TMS320TCI6487
- □ TMS320TCI6488 (Full Peripheral Register set)
- □ TMS320TCI6488(CPU Registers, Fast Start)
- □ TMS570LS20206SPGE
- □ TMS570LS20206SZWT
- □ TMS570LS20216SPGE
- ☑ TMS570LS20216SZWT

TMS570LS20216S 337Pin BGA

Advanced Setup

Target Configuration: lists the configuration options for the target.

Save Configuration

Save

```
Texas Instruments XDS100v2 USB Emulator_0/CortexR4 [Non-Project Debug Session] Texas Instrument
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address
CortexR4: Flash operation timed out waiting for the algorith
CortexR4: Flash Programmer: Error initializing device.
CortexR4: Flash operation timed out waiting for the algorith
CortexR4: Flash Programmer: Error initializing device.
CortexR4: Error: Error 0x80001822/-1141 Fatal Error during: 
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address
CortexR4: GEL Output:   Memory Map Setup for Flash @ Address
```

Figure 6. Code Composer Studio View of Shared Project Folder - After HalCoGen Generation Step

Now you will be able to make changes to the same project through either the HalCoGen environment or through the Code Composer Studio editor. And you will be able to regenerate code without losing your edits in the Code Composer Studio environment, as long as they are placed in the designated user code areas.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connctivity	www.ti.com/wirelessconnectivity		
		TI E2E Community Home Page	e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265