

**IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF TEXAS
DALLAS DIVISION**

LUCIO DEVELOPMENT LLC,

Plaintiff,

vs.

TEXAS INSTRUMENTS
INCORPORATED,

Defendant.

§
§
§
§
§
§
§
§
§
§
§

Case No:

PATENT CASE

COMPLAINT

Plaintiff Lucio Development LLC (“Plaintiff” or “Lucio”) files this Complaint against Texas Instruments Incorporated (“Defendant” or “TI”) for infringement of United States Patent No. 7,069,546 (hereinafter “the ‘546 Patent”).

PARTIES AND JURISDICTION

1. This is an action for patent infringement under Title 35 of the United States Code. Plaintiff is seeking injunctive relief as well as damages.

2. Jurisdiction is proper in this Court pursuant to 28 U.S.C. §§ 1331 (Federal Question) and 1338(a) (Patents) because this is a civil action for patent infringement arising under the United States patent statutes.

3. Plaintiff is a Texas limited liability company with its office address at 555 Republic Dr., Suite 200, Plano, Texas 75074.

4. On information and belief, Defendant is a Delaware corporation with its principal office at 12500 T I Blvd, Dallas, TX 75243.

5. This Court has personal jurisdiction over Defendant because Defendant has

committed, and continues to commit, acts of infringement in this District, has conducted business in this District, and/or has engaged in continuous and systematic activities in this District.

6. On information and belief, Defendant's instrumentalities that are alleged herein to infringe were and continue to be used, imported, offered for sale, and/or sold in this District.

VENUE

7. Venue is proper in this District pursuant to 28 U.S.C. §1400(b) because acts of infringement are occurring in this District and Defendant has a regular and established place of business in this District. For instance, on information and belief, Defendant has a regular and established place of business at 12500 T I Blvd, Dallas, TX 75243.

COUNT I **(INFRINGEMENT OF UNITED STATES PATENT NO. 7,069,546)**

8. Plaintiff incorporates paragraphs 1 through 7 herein by reference.

9. This cause of action arises under the patent laws of the United States and, in particular, under 35 U.S.C. §§ 271, *et seq.*

10. Plaintiff is the owner by assignment of the '546 Patent with sole rights to enforce the '546 Patent and sue infringers.

11. A copy of the '546 Patent, titled "Generic Framework for Embedded Software Development," is attached hereto as Exhibit A.

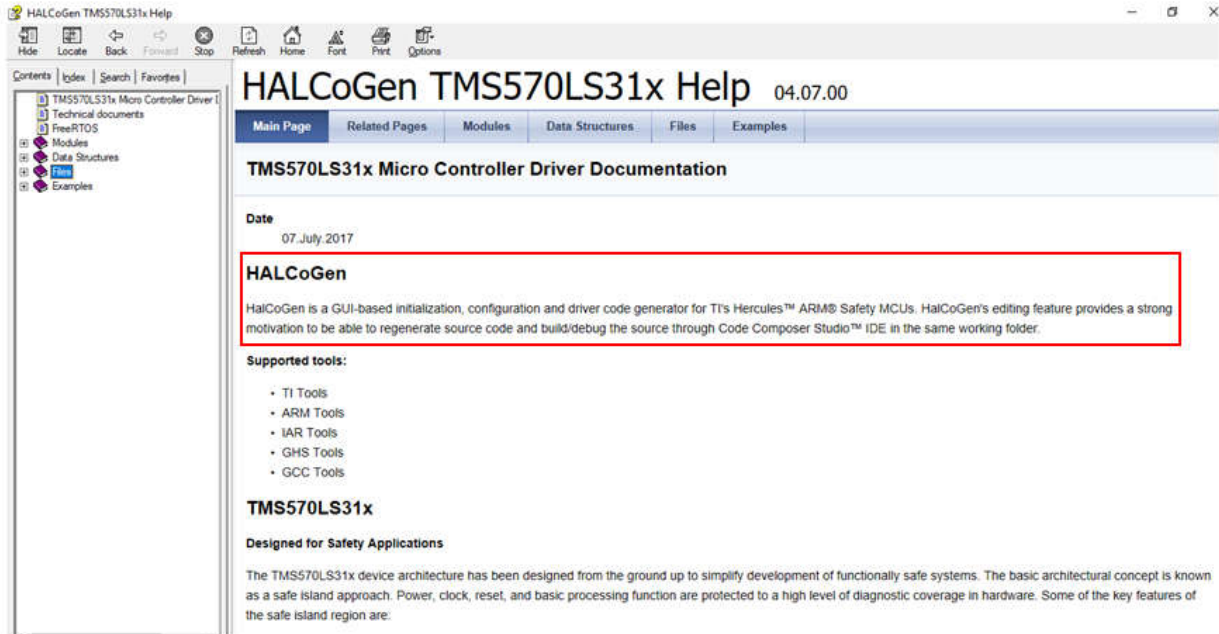
12. The '546 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

13. On information and belief, Defendant has infringed and continues to infringe one or more claims, including at least Claim 1, of the '546 Patent by making, using, importing, selling, and/or offering for sale a software platform for embedded software development,

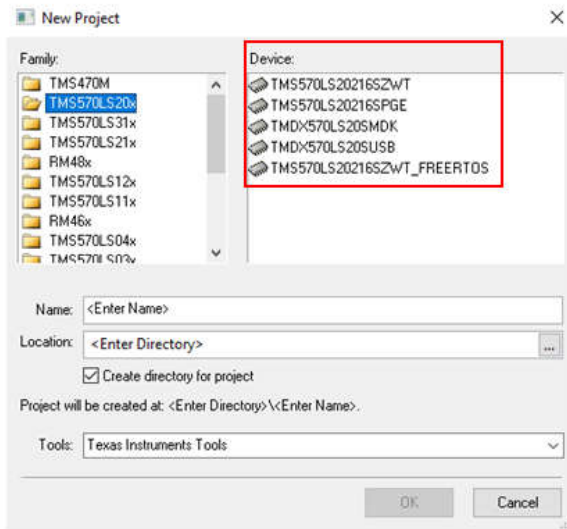
which is covered by at least Claim 1 of the '546 Patent. Defendant has infringed and continues to infringe the '546 Patent directly in violation of 35 U.S.C. § 271.

14. Defendant, sells, offers to sell, and/or uses embedded software development packages including, without limitation, HALCoGen, and any similar products (“Product”), which infringe at least Claim 1 of the '546 Patent.

15. HALCoGen is a GUI-based initialization, configuration and driver code generator for TI's Hercules™ ARM® Safety MCU families such as TMS470M, TMS570LS20x, TMS570LS31x etc.. HALCoGen provides one or more generic application handler programs, each such program comprising computer program code for performing generic application functions common to multiple types of hardware modules used in a communication system. For example, HALCoGen provides a Hardware Abstraction Layer (HAL) which includes files such as prebuilt libraries and header files (“generic application handler”) during installation. Header files include source code comprising functions and data structures, which are common and uniform across all supported HALCoGen hardware modules. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.



Source: HALCoGen software Help guide.



Source: HALCoGen software Help guide.

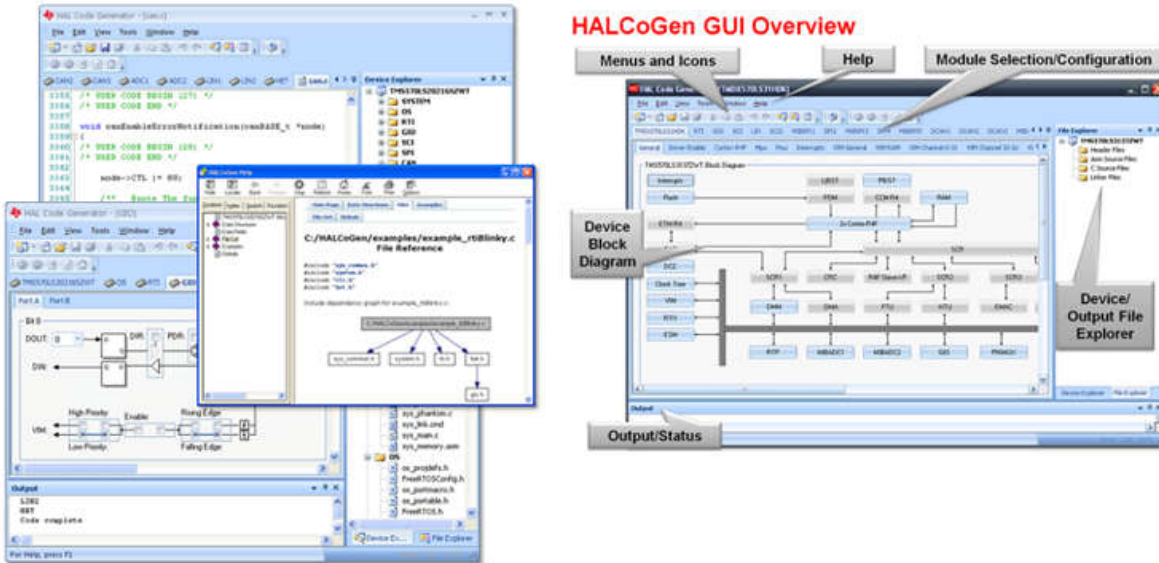
Software Maturity Table

last updated - 12-Oct-2015

Device Family	Software Quality	Device Family	Software Quality
TMS470M	Beta	TMS570LS20x	Beta
TMS570LS31x	GA	RM48x	GA
TMS570LS21x	GA	RM46x	GA
TMS570LS12x	GA	TMS570LS11x	GA
TMS570LS04x	GA	RM42x	GA
TMS570LS03x	GA	RM41x	GA
TMS570LS02x	GA		
TMS570LS09x_07x	GA	RM44x	GA
TMS570LC43x	GA	RM57Lx	GA

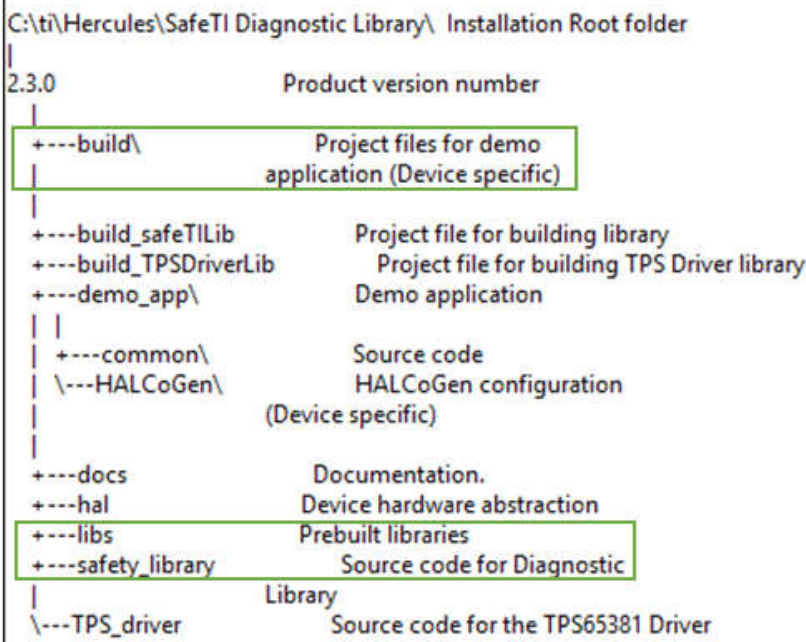
Source: <http://processors.wiki.ti.com/index.php/HALCoGen>

Hardware Abstraction Layer (HAL) is the lowest software layer. It contains software modules with direct access to MCU and is responsible for system initialization. HALCoGen allows users to generate these HAL device drivers for Hercules™ safety critical microcontrollers. Simple GUI helps in configuring and generating the microcontroller abstraction layer including the safety related functions.

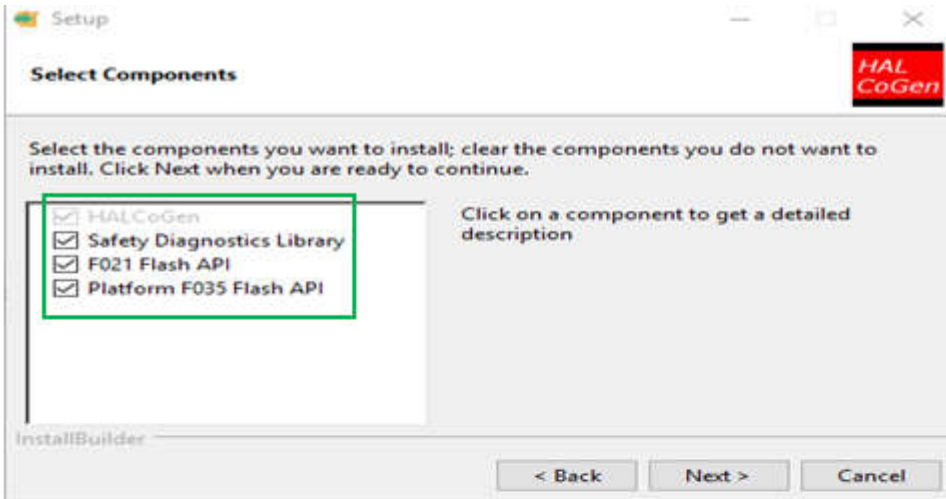


Source: [HALCoGen software Help guide](#).

The installer for this software installs by default to the C:\ti\Hercules folder. The folder structure is as shown below:



Source: [HALCoGen software Help guide](#).



Source: HALCoGen software Help guide.

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

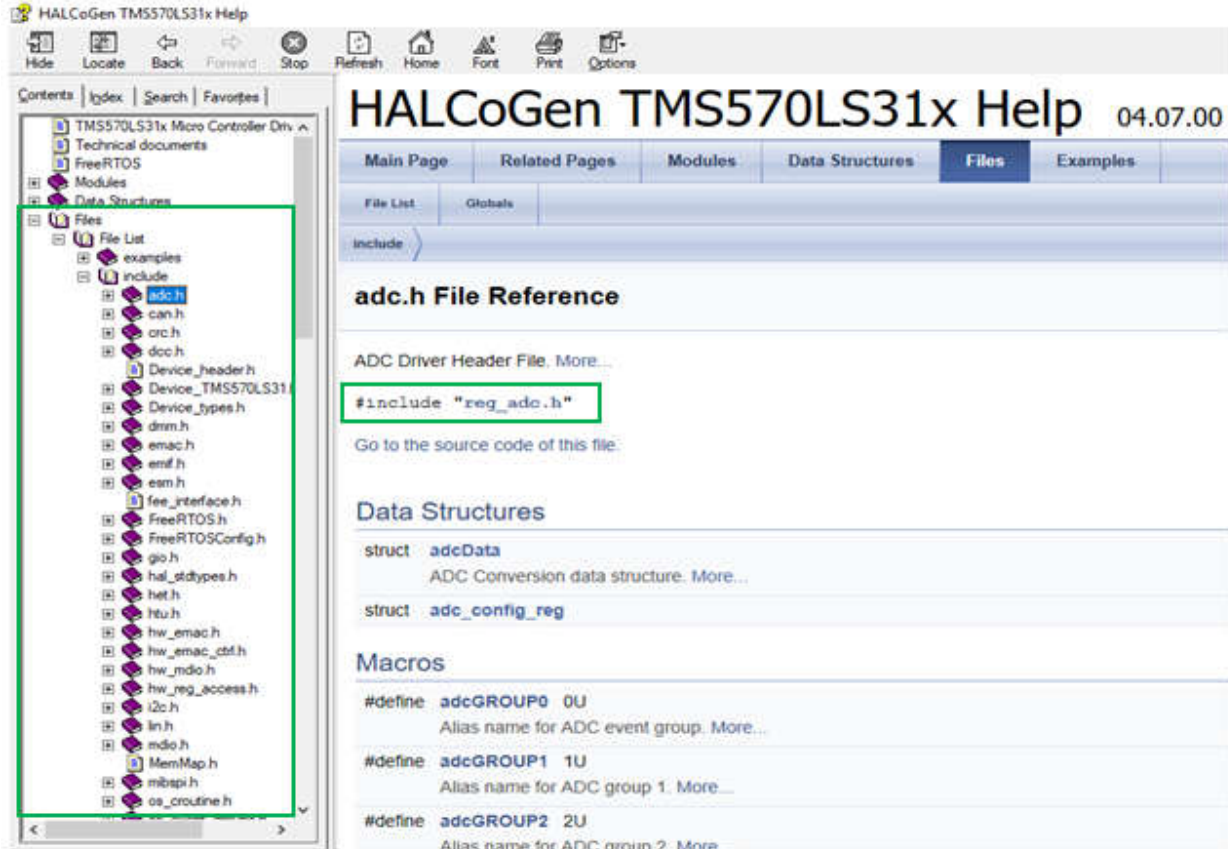
```
/* USER CODE BEGIN (2) */
/* USER CODE END */

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.

Source: <http://www.ti.com/lit/an/spna121b/spna121b.pdf>, page 1.



Source: HALCoGen software Help guide.

16. HALCoGen generates specific application handler code to associate the generic application functions with specific functions of a device driver for at least one of the types of the hardware modules. For example, in addition to the generic drivers and HAL, HALCoGen SDK also includes specific microcontroller handler code that is specific to the application and specific to particular microcontroller families. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

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.

Source: <http://www.ti.com/lit/an/spna121b/spna121b.pdf>, page 1.

The installer for this software installs by default to the C:\ti\Hercules folder. The folder structure is as shown below:

```
C:\ti\Hercules\SafeTI Diagnostic Library\ Installation Root folder
|
2.3.0          Product version number
|
+---build\     Project files for demo
|              application (Device specific)
|
+---build_safeTILib   Project file for building library
+---build_TPSDriverLib Project file for building TPS Driver library
+---demo_app\       Demo application
|
| +---common\      Source code
| |---HALCoGen\   HALCoGen configuration
| |              (Device specific)
|
+---docs         Documentation.
+---hal          Device hardware abstraction
+---libs         Prebuilt libraries
+---safety_library Source code for Diagnostic
|              Library
|
\---TPS_driver   Source code for the TPS65381 Driver
```

Source: HALCoGen software.

17. HALCoGen generates specific application handler code and defines a specific element in the specific code to be handled by one of the generic application functions for that hardware module. For example, HALCoGen generates system-specific application handler

code by defining specific elements such as functions and data structures (“specific element”) corresponding to specific hardware modules (such as TMS570LS31x USB Development Stick or TMS570LS31x Hercules Development Kit) that extend or otherwise connect the system-specific application handler code and data structures made available by the generic application handler code of the HALCoGen SDK. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Overview

- **In this example we will:**
 - Create a new TMS570LS31x HALCoGen Project
 - Configure HALCoGen to generate
 - A basic PWM with a period of 1 second and a duty cycle of 75%
 - A basic PWM with a period of 0.5 seconds and a duty cycle of 50%
 - Use the PWM to toggle the NHET1[0] and NHET1[17] LEDs on the development board
 - Generate and Import code into Code Composer Studio
 - Build and Deploy our code to the microcontroller
- **Required Hardware:**
 - Windows Based PC (WinXP, Vista, 7)
 - [TMS570LS31x USB Development Stick](#) or [TMS570LS31x Hercules Development Kit](#)
- **Required Software:**
 - [HALCoGen v2.11 or higher](#)
 - [Code Composer Studio v4.x](#)



Source: http://processors.wiki.ti.com/images/d/d6/TMS570LS31x_N2HET_PWM_Example.pdf, page 2

The screenshot displays the HALCoGen TMS570LS31x Help application. The title bar reads "HALCoGen TMS570LS31x Help" and the version is "04.07.00". The interface includes a navigation menu on the left with categories like "Contents", "Index", "Search", and "Favorites". The main content area is titled "HALCoGen TMS570LS31x Help" and features a navigation bar with tabs for "Main Page", "Related Pages", "Modules", "Data Structures", "Files", and "Examples". The "Data Structures" tab is active, showing a list of data structures with brief descriptions. A green box highlights the "Data Structures" section header and the list of structures.

Data Structures

Here are the data structures with brief descriptions:

Data Structure	Description
A_BLOCK_LINK	
acmp_format	
ACMP_INSTRUCTION	
acnt_format	
ACNT_INSTRUCTION	
adc_config_reg	
ADC_format	
ADC_INSTRUCTION	
adcBase	ADC Register Frame Definition
adcData	ADC Conversion data structure
ADCNST_format	
ADCNST_INSTRUCTION	
ADD_format	
ADD_INSTRUCTION	
ADM32_format	
ADM32_INSTRUCTION	

Source: HALCoGen software Help guide.

The screenshot shows the HALCoGen TMS570LS31x Help application. The title bar reads "HALCoGen TMS570LS31x Help" and the version is "04.07.00". The interface includes a navigation pane on the left with a tree view of the help content, including sections for "Files" and "File List". The main content area is titled "adc.h File Reference" and contains the following information:

adc.h File Reference

ADC Driver Header File. More...

```
#include "reg_adc.b"
```

Go to the source code of this file.

Data Structures

```
struct adcData
    ADC Conversion data structure. More...
```

```
struct adc_config_reg
```

Macros

```
#define adcGROUP0 0U
    Alias name for ADC event group. More...
```

```
#define adcGROUP1 1U
    Alias name for ADC group 1. More...
```

```
#define adcGROUP2 2U
    Alias name for ADC group 2. More...
```

Source: HALCoGen software Help guide.

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 (Source and Include) 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 4)

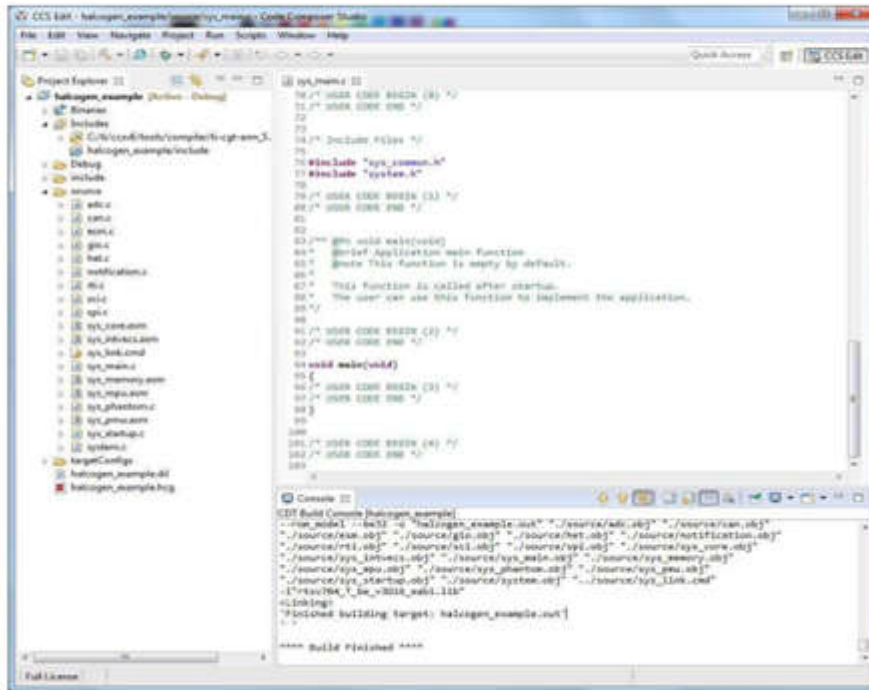


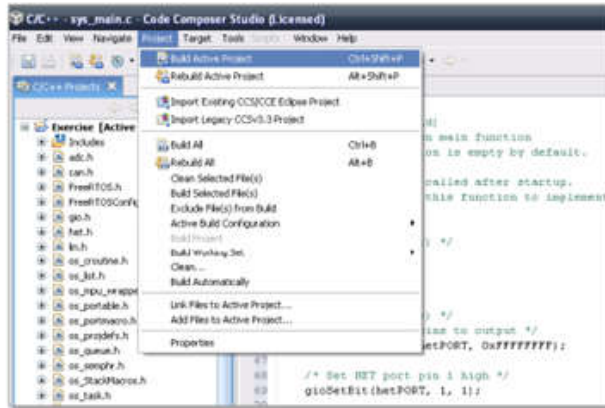
Figure 4. Code Composer Studio View of Shared Project Folder - After HALCoGen Generation Step

Source: <http://www.ti.com/lit/an/spna121b/spna121b.pdf>, page 3.

18. HALCoGen compiles the generic application handler programs together with the specific application handler code to produce machine-readable code to be executed by an embedded processor in the at least one of the types of the hardware modules. For example, when a specific application is needed for a particular hardware, the generic functions and the specific functions are compiled together to yield a machine readable code. HALCoGen SDK and/or its customers compile the generic functions and the specific functions using HALCoGen SDK and/or any other compiling SDK supported by Texas Instruments. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Compiling the Project

- The code is now complete and we are ready to build our project.
 - Go to Project → Build Active Project

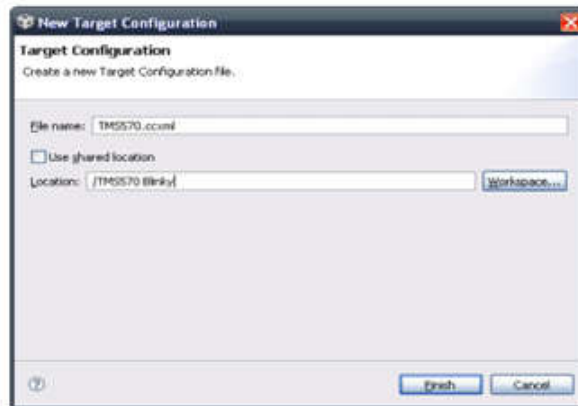


- Now that we have our .out file, we need to program the microcontrollers Flash memory.

Source:http://processors.wiki.ti.com/images/6/69/TMS570LS2x_Blinky_Example.pdf, page 15.

Creating a Target Configuration

- Before we begin, we must make a new target configuration, this tells CCS4 what device this project is designed for.
 - Target → New Target Configuration
- A new window will appear, we will make our file name "TMS570.ccxml"
- Click Finish

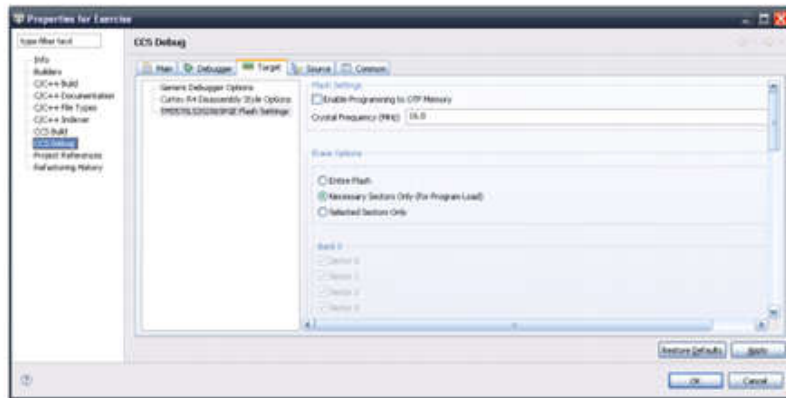


Source:http://processors.wiki.ti.com/images/6/69/TMS570LS2x_Blinky_Example.pdf, page 16.

Flash Programming Configuration

- It is possible to make the flash programming process much faster by only the necessary erasing and programming the necessary regions of flash memory.
 - To do so go to Project → Properties
 - In the window that appears select 'CCS Debug'
 - In the CCS Debug window select the TMS570LS20216SPGE Flash Settings option in the 'Target' tab.
 - Then select the 'Necessary Sectors Only' option in the Erase Options area, then click the 'Apply' button.

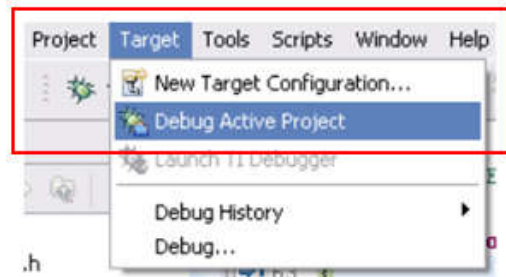
NOTE:
This option
is only
available in
CCSv4.2
and newer



Source: http://processors.wiki.ti.com/images/6/69/TMS570LS2x_Blinky_Example.pdf, page 18.

Programming the Flash

- We are now ready to program the flash.
 - Go to Target → Debug Active Project
 - A new window should appear as it programs the flash memory.
 - This may take a few moments.



Source:http://processors.wiki.ti.com/images/6/69/TMS570LS2x_Blinky_Example.pdf, page 19.

19. Defendant's actions complained of herein will continue unless Defendant is enjoined by this court.

20. Defendant's actions complained of herein are causing irreparable harm and monetary damage to Plaintiff and will continue to do so unless and until Defendant is enjoined and restrained by this Court.

21. Plaintiff is in compliance with 35 U.S.C. § 287.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff asks the Court to:

- (a) Enter judgment for Plaintiff on this Complaint on all causes of action asserted herein;
- (b) Enter an Order enjoining Defendant, its agents, officers, servants, employees, attorneys, and all persons in active concert or participation with Defendant who receive notice of the order from further infringement of United States Patent No. 7,069,546 (or, in the alternative, awarding Plaintiff a running royalty from the time of judgment going forward);
- (c) Award Plaintiff damages resulting from Defendant's infringement in accordance with 35 U.S.C. § 284;
- (d) Award Plaintiff pre-judgment and post-judgment interest and costs; and
- (e) Award Plaintiff such further relief to which the Court finds Plaintiff entitled under law or equity.

Dated: January 31, 2019

Respectfully submitted,

/s/ Jay Johnson

JAY JOHNSON

State Bar No. 24067322

D. BRADLEY KIZZIA

State Bar No. 11547550

KIZZIA JOHNSON, PLLC

1910 Pacific Ave., Suite 13000

Dallas, Texas 75201

(214) 451-0164

Fax: (214) 451-0165

jay@kjpllc.com

bkizzia@kjpllc.com

ATTORNEYS FOR PLAINTIFF

EXHIBIT A