Grace for Code Composer Studio™ IDE

Getting Started Guide

Literature Number: SLAU476A
December 2012–Revised June 2014

Contents

List of Figures

Grace for Code Composer Studio™ IDE

1 Introduction

Grace provides an intuitive way to configure MSP430™ microcontroller devices. The Grace tool is a plug-in to Code Composer Studio (CCS) that allows MSP430 developers to generate peripheral setup code within minutes. Typical application use cases are provided with easy-to-follow steps for initial peripheral configuration. In addition, ready-to-use code snippets for peripheral runtime control can be copied straight into your applications.

1.1 Supported Devices

For a list of supported devices, see the Grace Quick Start Guide.

1.2 Purpose of Grace

Code generated by Grace covers only device initialization. After device initialization, the first function that is usually executed is the main function, and there is no further interaction between Grace and the application code.

1.3 Grace, MSP430Ware, and Driver Library

For MSP430FR5xx devices, a low-level library (Driver Library) is available. Grace uses the Driver Library to configure the MCU. It does this by generating source code for the FRAM devices. The source code makes calls to the MSP430 Driver Library. You can use the Driver Library in your application code. Direct register access is also possible.

The Driver Library for MSP430FR5xx devices is available as either a standalone component or as part of MSP430Ware. The easiest way to make sure that the Driver Library is installed is to use the App Center in Code Composer Studio 6 to install Grace.

The Driver Library API documentation can be found in the TI Resource Explorer. Select **TI Resource Explorer** from the **View** menu. Then select **MSP430Ware** → **Libraries** → **DriverLib** → **User's Guide** for the documentation.

MSP430F2xx and MSP430G2xx devices are not supported through the Driver Library. Instead, Grace provides support by directly accessing the peripheral registers.

2 Installing Grace

If you are using Code Composer Studio 6 (CCSv6), you will need to install Grace from the App Center. Follow these steps to install Grace:

1. From the CCSv6 menu bar, choose **View → CCS App Center**.
2. In the App Center's search field, type **Grace**.
3. Check the "Selected" box for the Grace component. Then click the **Install Software** button, which is next to the search field (see Figure 1).

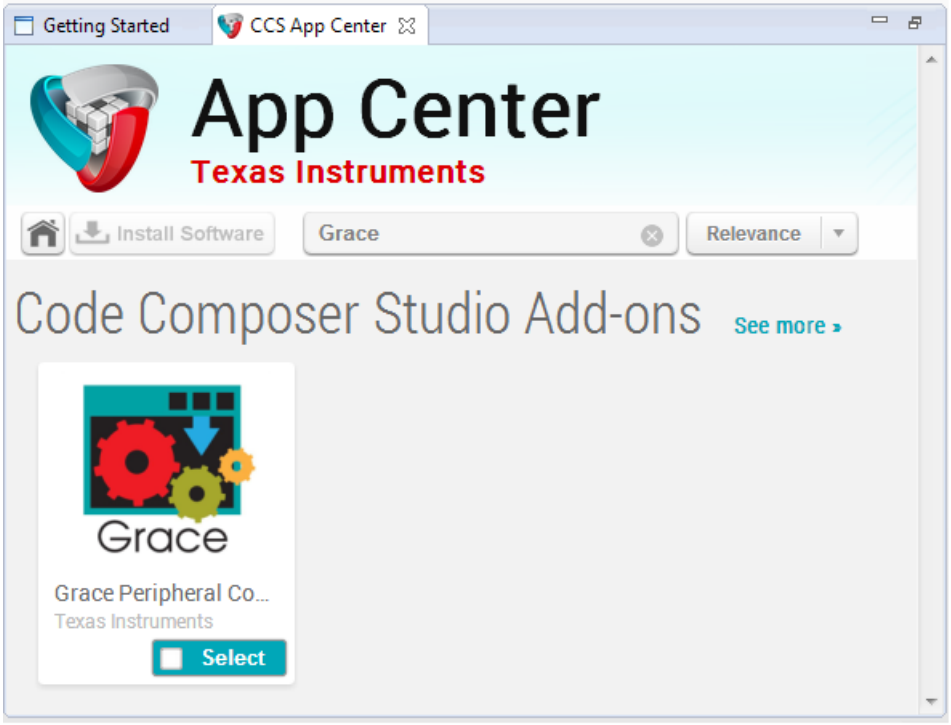

Figure 1. Grace in the CCS App Center

4. Accept the license agreement and click **OK**.
5. When the installation is complete, restart CCS.

If you are using CCSv5, Grace is included in the default installation for MSP430. If you do not have Grace installed, you can reinstall it by running the CCSv5 setup, and selecting the Grace installation option on the "Select Components" page of the installer.

3 Starting a Grace Project

When starting CCS, you select a workspace where you will store your projects. In most cases, the default setting is fine, so click **OK** (see Figure 2).

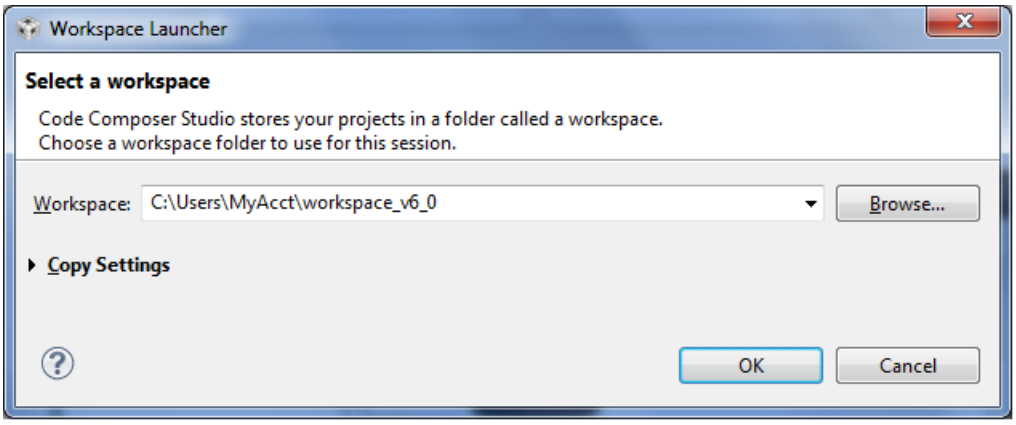

Figure 2. Workspace Launcher, Select a Workspace

When CCS has finished the startup process, start a new Grace-enabled MSP430 project.

1. From the CCS menu bar, choose **File** → **New** → **CCS Project** (see Figure 3).

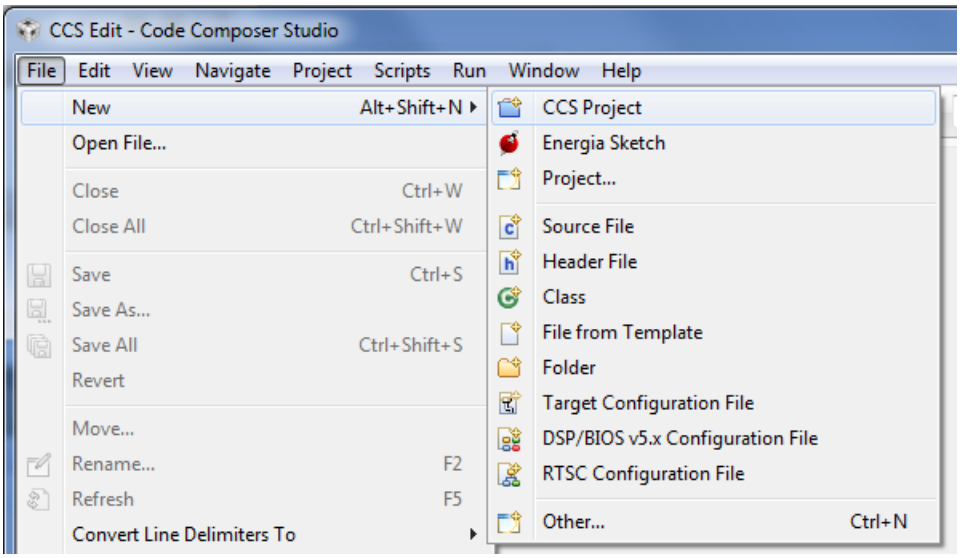

Figure 3. New CCS Project

2. In the New CCS Project wizard (see Figure 4), type part of your device name in the **Target** field, and select your MSP430 target device from the drop-down list.
3. Make sure the correct **Connection** is selected, or click **Identify** to test the selected connection.
4. Type a **Project Name** that will be used in the Project Explorer to identify this project.
5. In the "Project templates and examples" list, choose **Empty Grace (MSP430) Project**.
6. Click **Finish**. (You can click **Next** if you want to make the project use non-default software versions.)

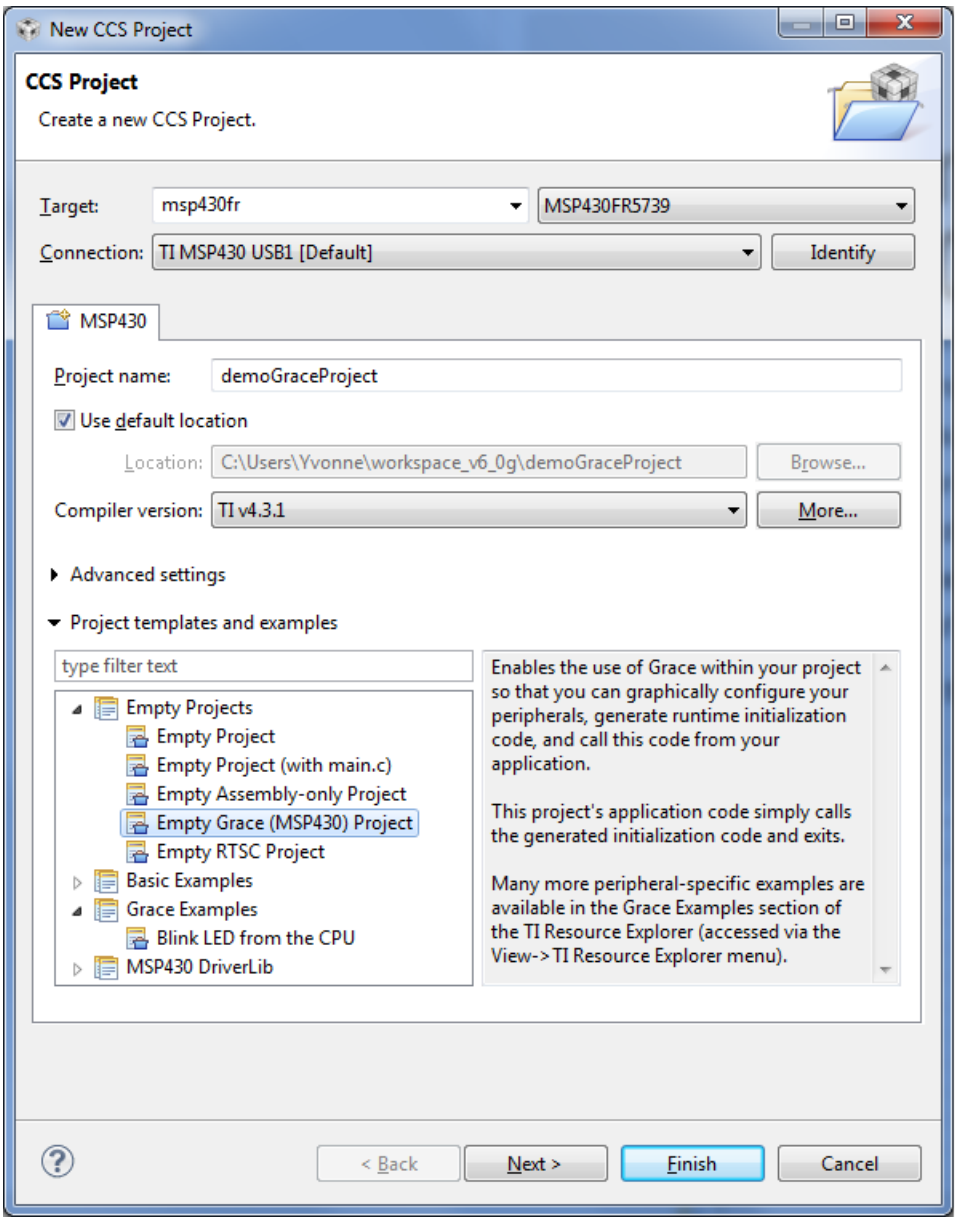

Figure 4. New CCS Project Setup

When the loading process has completed, the Grace Welcome page opens (see Figure 5). This is the main Grace page for the main.cfg configuration file in the project. This page provides useful information, including basic usage instructions and helpful links. It is a good idea to spend some time reading this page.

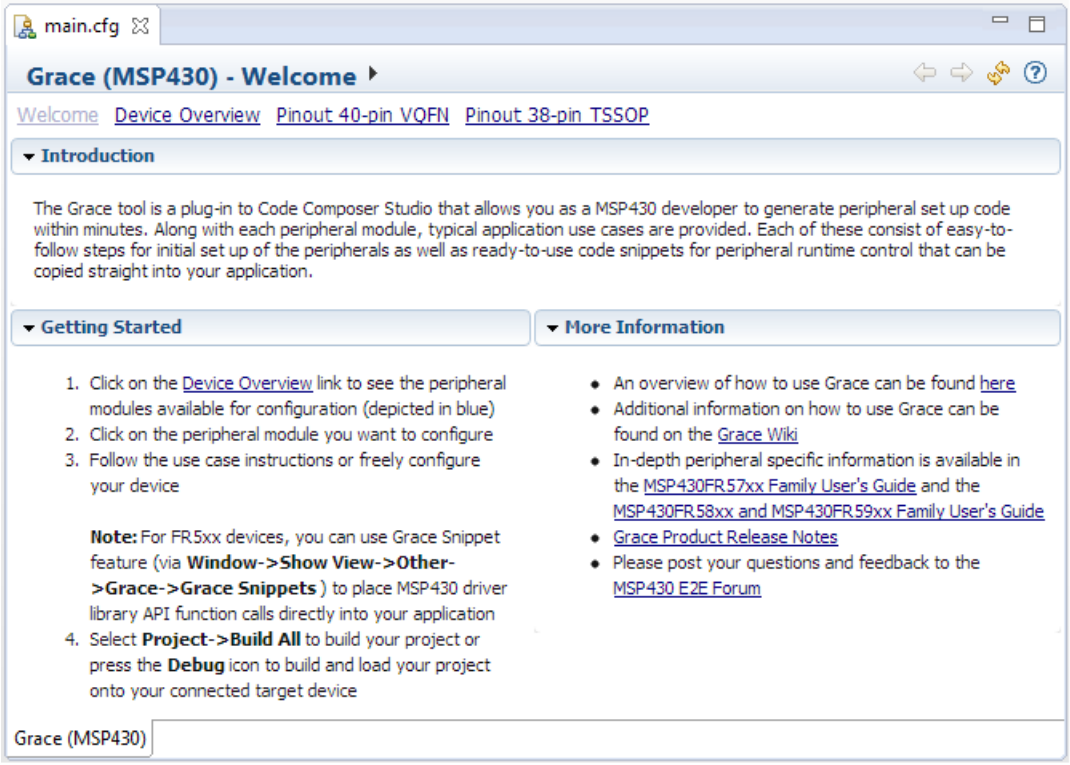

Figure 5. Grace Welcome Page

Click the "Device Overview" link near the top of the page to see an overview of your device's peripherals (see Figure 6).

Click on any peripheral shown as a blue box (for example, a Timer) to see its configuration options. You can also select a peripheral by clicking the arrow in the top bar.

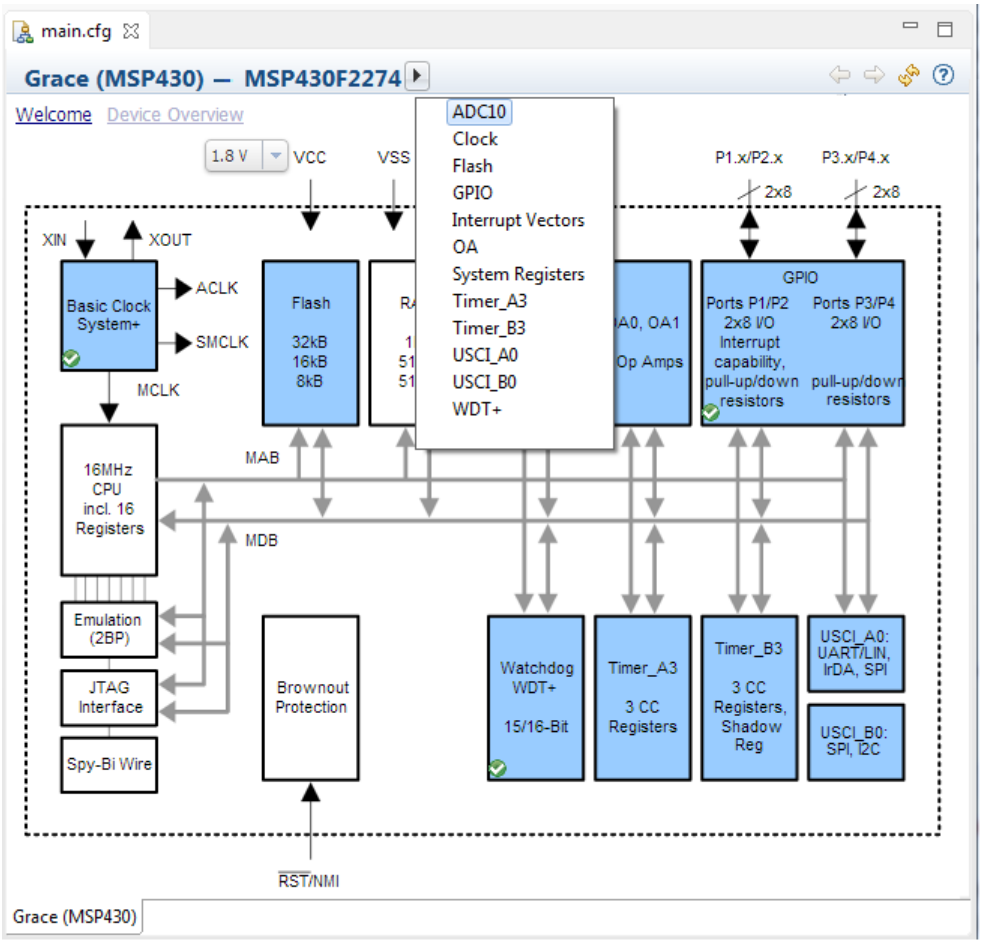

Figure 6. Device Overview

Notice the green checkmarks in some of the boxes. These checkmarks indicate that Grace is configured to generate initialization code for these peripherals. You can enable or disable use of a peripheral by right-clicking on a blue box.

Help tips are available for a number of peripherals and options. These tips are shown when the mouse cursor hovers over a configurable item on the page (see Figure 7).

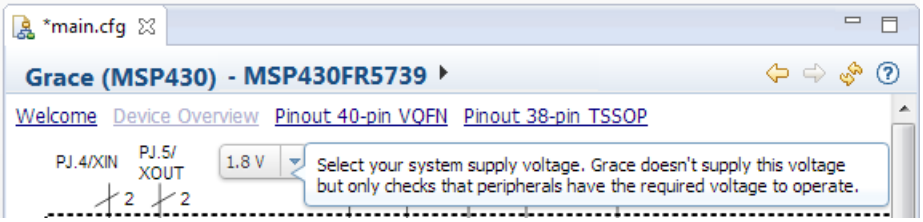

Figure 7. Screen Tips

The procedures for peripheral configuration differ somewhat depending on the device family you are using. See Section 4 for G2xx/F2xx peripheral configuration and Section 5 for FR5xx peripheral configuration.

4 Configuring G2xx and F2xx Devices

For G2xx/F2xx devices, in the Device Overview diagram, click on any peripheral shown as a blue box (for example, a Timer) to see its configuration options. You can also select a peripheral by clicking the arrow in the top bar. Notice the green checkmarks in some of the boxes. These checkmarks indicate that Grace is configured to generate initialization code for these peripherals.

4.1 Overview Page for G2xx and F2xx Devices

The Overview page for each G2xx and F2xx peripheral allows you to enable that peripheral. Check the "Enable … in my configuration" box to allow Grace to configure this peripheral. The "Introduction" section gives a short explanation of the peripheral's capabilities.

Expand a Use Case for this peripheral to see a description of the use case and sample code. The Use Cases contain instructions on how to configure the peripheral. You can use the sample code as a starting point for your own code (see Figure 8).

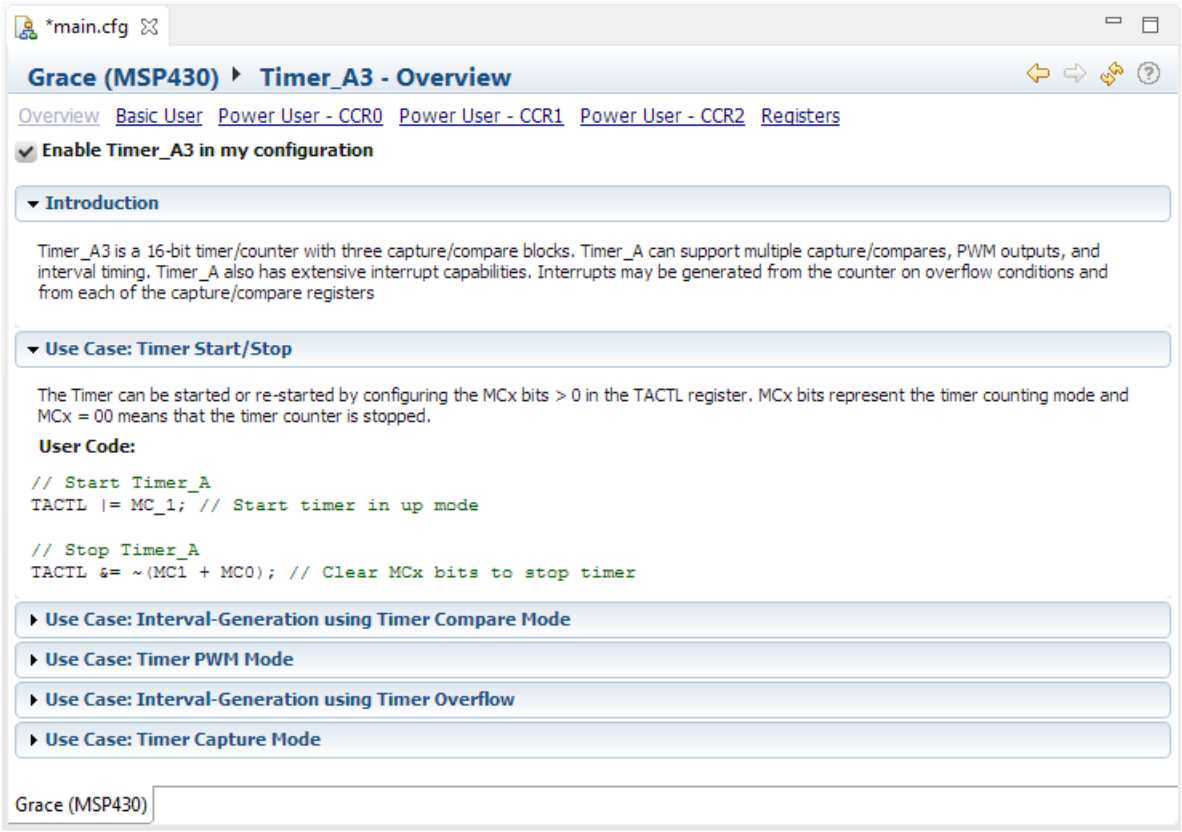

Figure 8. G2xx/F2xx Timer_A3 Use Cases

To copy the User Code, click and drag your mouse over the code. The code may not be highlighted, but you can press Ctrl+C to copy the code to your clipboard. Paste the code into your source file and remove any extra text that was copied as needed.

4.2 Basic and Power User Views for G2xx and F2xx Devices

After you enable a G2xx/F2xx peripheral, links to "Basic User" and "Power User" views are shown near the top of the page.

The "Basic User" view (see Figure 9) shows the most important configuration options and can be reached via the link at the top. If you need additional options, use a "Power User" view for a more complex setup. For most use cases, the Basic User view is sufficient.

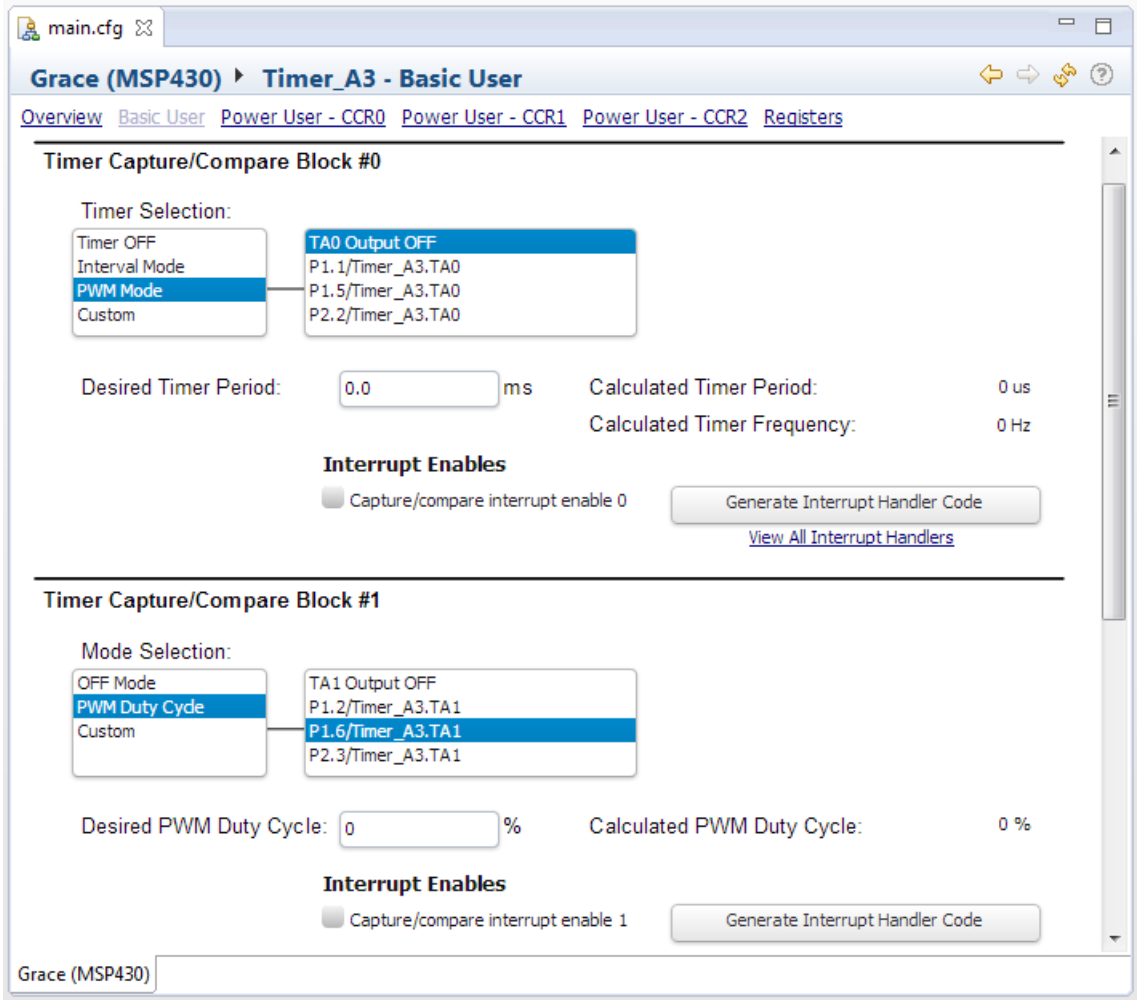

Figure 9. G2xx/F2xx Basic User Mode

The settings made in one view are automatically reflected in the other views. The advanced settings that are available in the Power User view are hidden in the Basic User view. You can switch between the Basic View and Power User Views when setting options.

The "Registers" view shows the register settings that result from your current configuration.

4.3 GPIO Configuration for G2xx and F2xx Devices

GPIO configuration should be the last step when configuring a device (see Figure 10). IOs that are used for peripheral functions should first be configured in the peripheral view as described in Section 4.

For G2xx and F2xx devices, click the blue box for the GPIO in the Device Overview diagram. Choose the link at the top of the GPIO page for your device package to see the correct GPIO view. The GPIO view shows a diagram of the current configuration (see Figure 10).

Next to each pin is a drop-down menu that allows you to configure the pin as an input, output, or a specific peripheral function. Note that configuring a pin manually to a peripheral function does not activate the peripheral or configure it to use the pin.

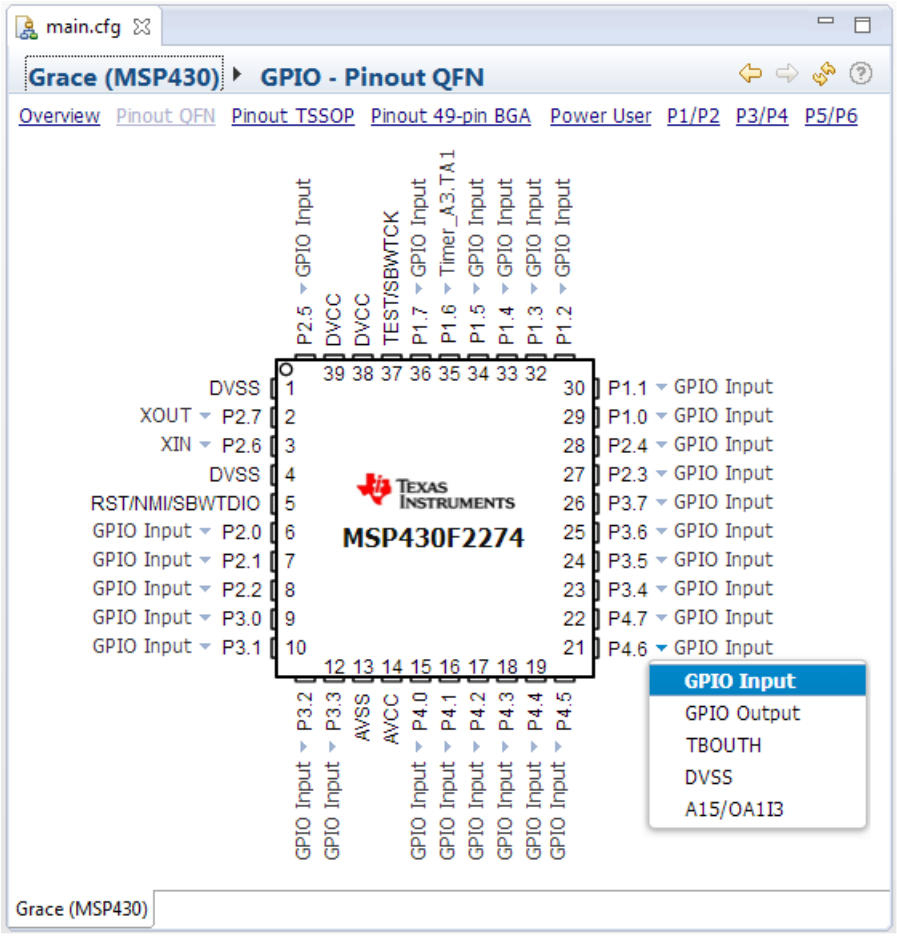

Figure 10. G2xx/F2xx GPIO Configuration, Pinout View

The GPIO's Power User view lets you make additional IO settings (see Figure 11). Switching from one view to the other does not change the configuration. Settings that are made in the Power User view might not be visible in the package views, but they are still present.

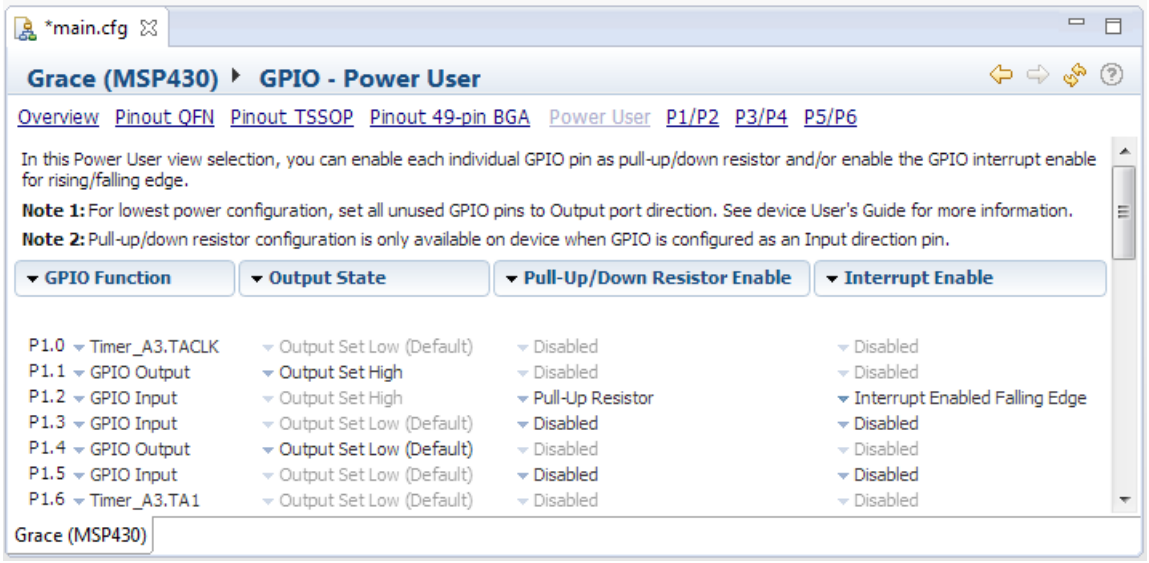

Figure 11. G2xx/F2xx GPIO Configuration, Power User View

5 Configuring FR5xx Devices

For FR5xx devices, when you select a peripheral in the Device Overview, one or more use cases for that peripheral are shown. Check the "Configure ... Peripheral" box at the top to allow Grace to configure this peripheral.

Click on a use case to enable it and show the available configuration options (see Figure 12).

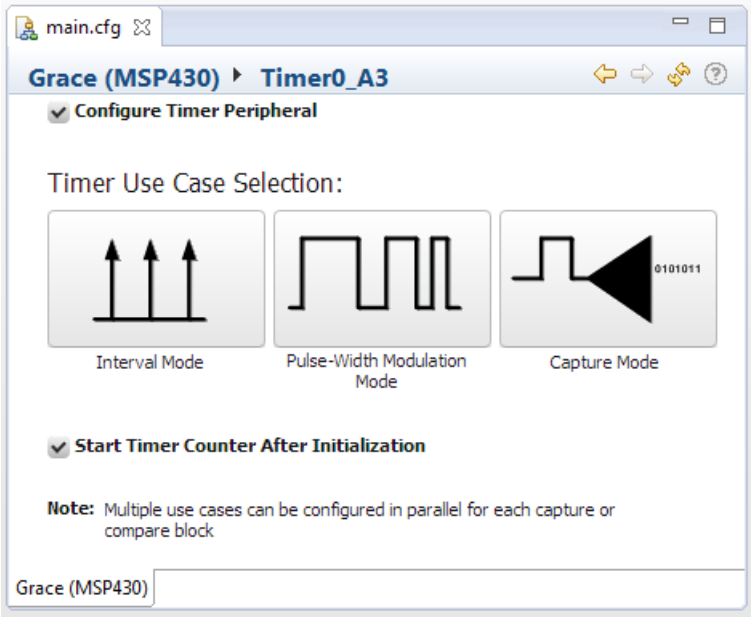

Figure 12. FR5xx Timer Peripheral Configuration

Some Use Cases have an "Advanced Settings" checkbox that displays additional options; for example, clock divider settings.

You can return to the Use Case Selection view at any time to change the peripheral's operation to a different Use Case. The timers allow several use cases to be configured simultaneously. Grace automatically checks the available timer resources and allows only valid configurations.

5.1 Pin Mux (GPIO) Configuration for FR5xx Devices

Pin mux configuration should be the last step when configuring a device (see Figure 10).

For FR5xx devices, pinout diagrams are available as links from the Device Overview page. These pinout diagrams are read-only. If a pin is configured to be used by a peripheral, a link to that peripheral configuration is provided. IOs that are used for peripheral functions should be configured in the peripheral view as described in Section 5.

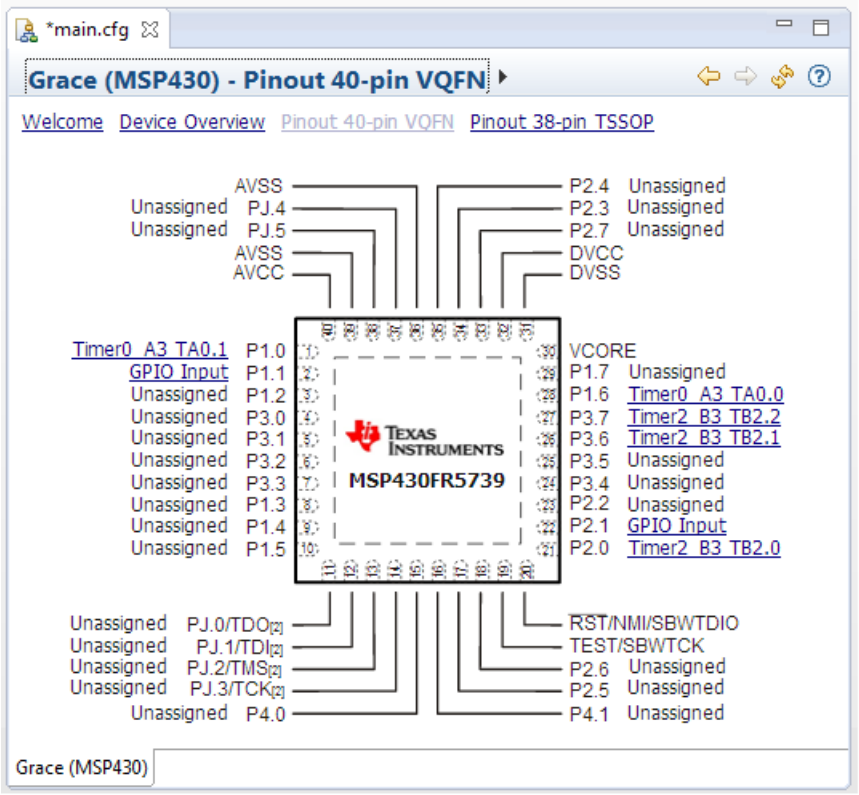

Figure 13. FR5xx Pinout View

When you select the GPIO block in the Device Overview diagram, you see the Power User view. The Power User view allows you to set pins to GPIO Input or GPIO Output.

There are no graphical Pinout views with write access to allow you to set GPIO behavior (as there are for G2xx/F2xx devices). If you want to set a pin to be used by a specific peripheral, you should make that setting in the configuration of that peripheral, not in the GPIO configuration.

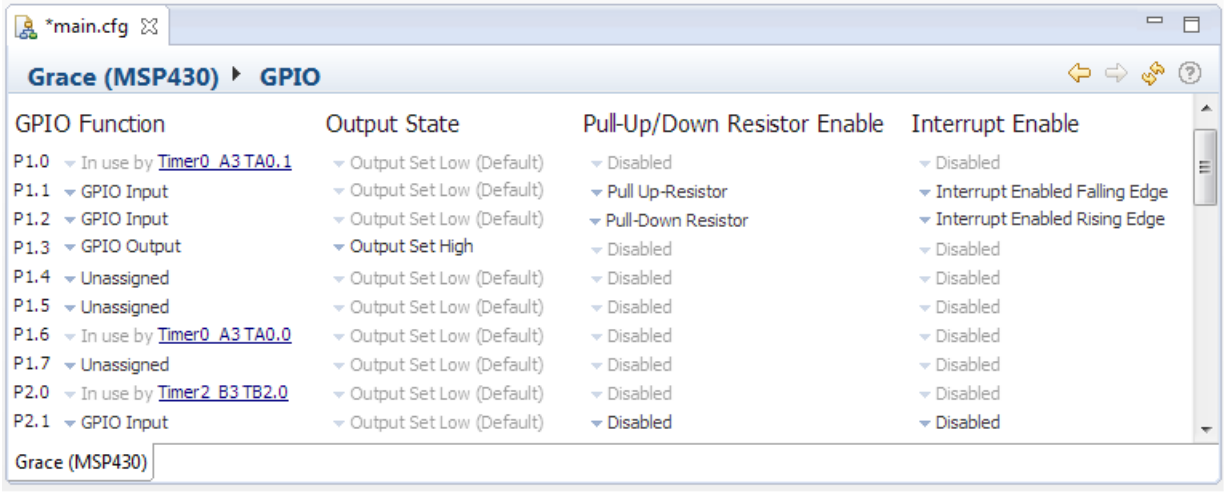

Figure 14. FR5xx GPIO Configuration, Power User View

You can configure the following settings in the Power User view:

- **GPIO Function.** Select either "GPIO Input" or "GPIO Output". By default, a pin is unassigned unless you have specified its use in the configuration for a specific peripheral.
- **Output State.** If you selected GPIO Output, you can set the Output State to "Output Set High". By default, pins have Output Set Low.
- **Pull-Up/Down Resistor Enable.** If you selected GPIO Input, you can enable either "Pull-Up Resistor" or "Pull-Down Resistor". The default is disabled. Choose a setting for this column for level-triggered interrupts. A pull-up resistor brings the voltage towards the high logic level when the peripheral connected to the pin is inactive. A pull-down resistor brings the voltage toward zero when the peripheral is inactive.
- **Interrupt Enable.** If you selected GPIO Input, you can enable either "Interrupt Enabled Falling Edge" (interrupt triggered when voltage changes from high to low) or "Interrupt Enabled Rising Edge" (interrupt triggered when voltage changes from low to high). The default is disabled. Choose a setting for this column for edge-triggered interrupts.

Settings that are made in the Power User view might not be visible in the package views, but they are still present.

6 Building the Project

Save any changes you have made with Grace to the *.cfg file. Click the **Build** icon (shown as a hammer) in the CCS tool bar to start the build process (see Figure 15). If a target is connected to the PC, you can start a debug session by clicking the **Debug** icon.

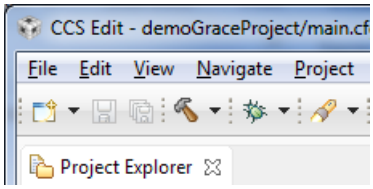

Figure 15. Build and Debug Icons

7 Using Grace Snippets

Grace allows you to insert predefined code templates called "Grace Snippets". These templates are device-specific and contain C code that helps program the device. The snippets range from simple port I/O settings to complex interrupt handler functions.

First, activate the Grace snippets by choosing **View** → **Grace Snippets** from the CCS menu bar (see Figure 16).

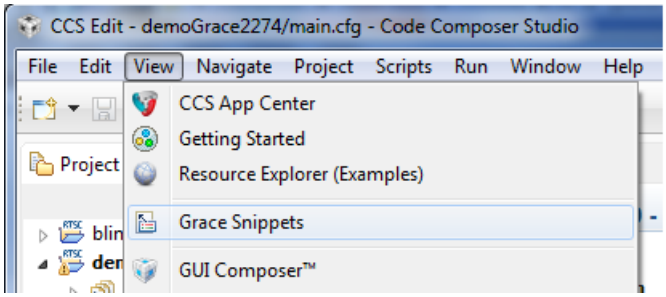

Figure 16. Activate Grace Snippets

To use Grace Snippets, you can insert the code by opening a C file in the editor view and then double-clicking on the snippet on the right side. You can also drag and drop the snippets to the cursor position. (See Figure 17.) For some snippets, you will be asked to provide values to use in the generated snippet.

For FR5xx devices, you must add the "!Header file include" Grace Snippet near the beginning of the C file for any peripheral you are using.

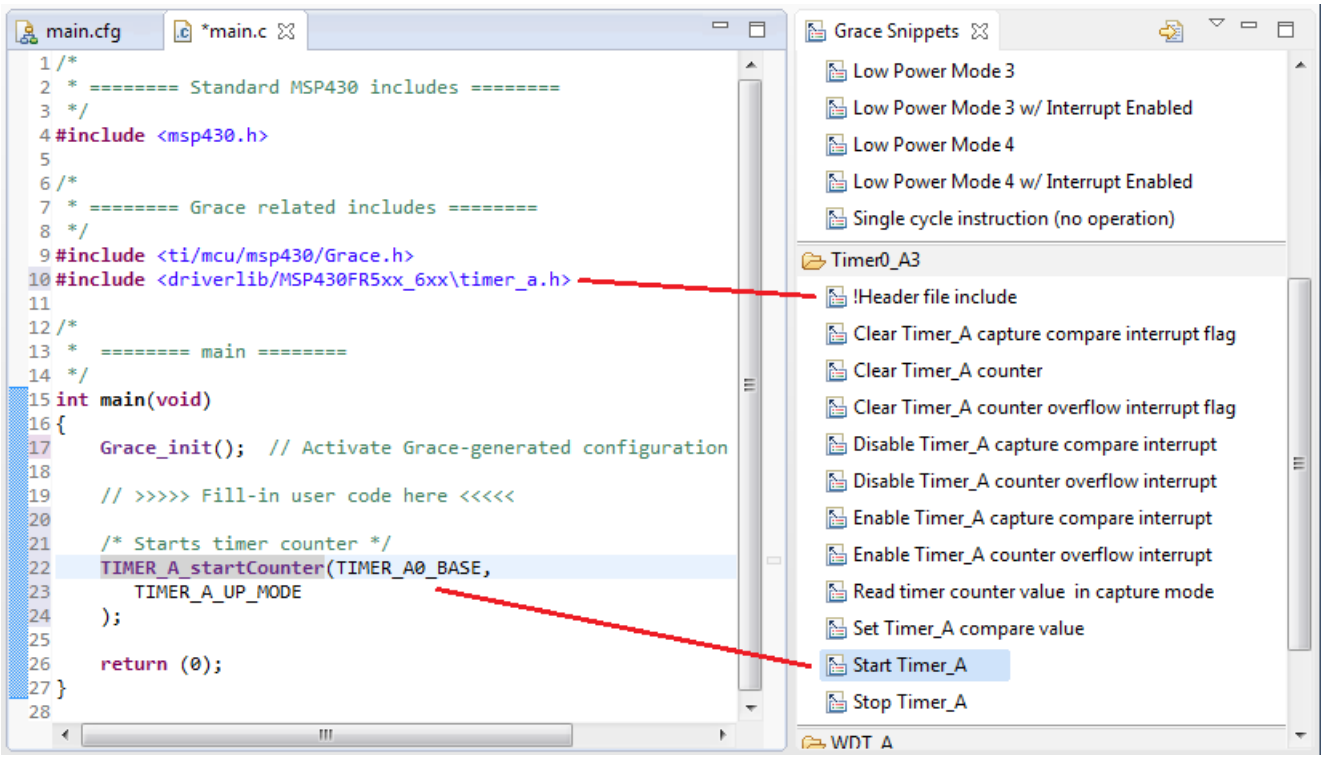

Figure 17. Header File for Grace Snippets

8 Reviewing the Generated Source Code

After building your application, the generated source can be found in the Project Explorer in the **src→grace** folder (see Figure 18).

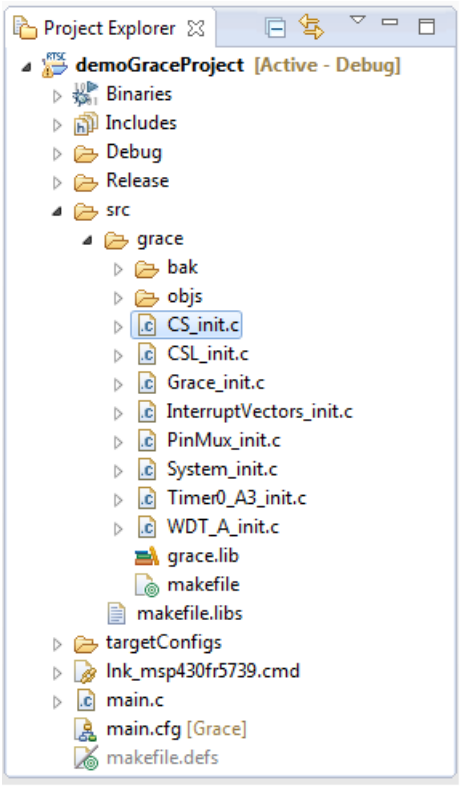

Figure 18. Project Explorer, Generated Source

The Interrupt_Vectors_init.c file is regenerated whenever you modify or save the configuration. The other initialization source files and the makefile are generated when you build the project. The makefile is used to compile and link the generated files to create grace.lib.

9 Interrupt Handling

Grace lets you manage peripheral interrupts efficiently.

You can configure interrupts in peripheral views by enabling an interrupt and clicking the associated button that says "Generate Interrupt Handler Code" (see Figure 19). These buttons toggle their text to allow you to enable or disable the generation of interrupt handler code. When a button says "Remove Interrupt Handler Code", generation of this interrupt handler code is currently enabled.

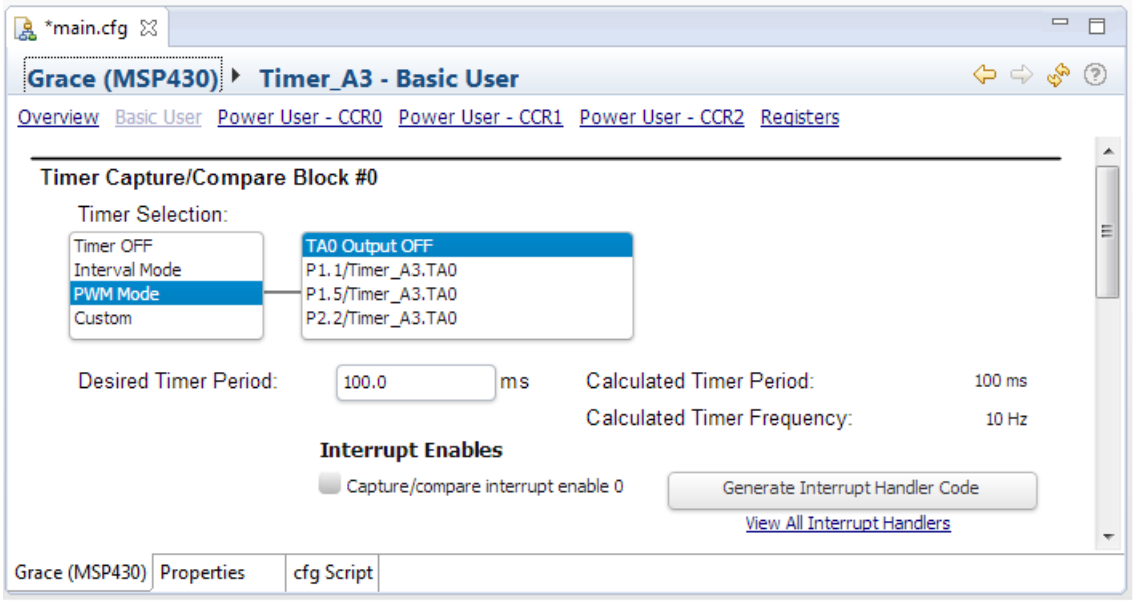

Figure 19. Timer Capture/Compare Block #0

Another way to configure interrupts in to open the Interrupt Vector List by clicking the "View All Interrupt Handlers" link in a peripheral view. Select an interrupt from the list and configure the settings for that interrupt on the right. (See Figure 20.)

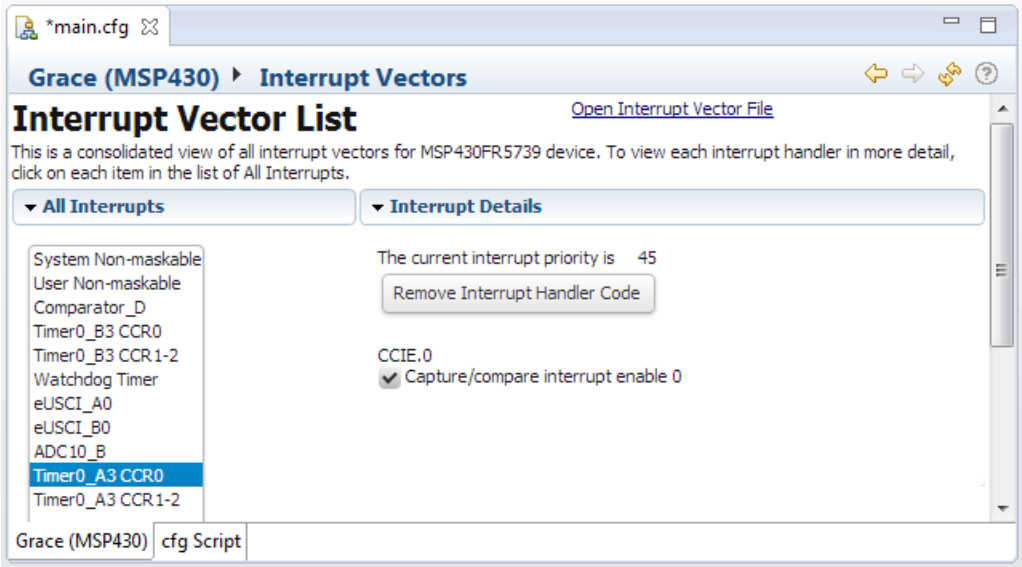

Figure 20. Interrupt Vector List

You can program the behavior of interrupts in the InterruptVectors_init.c file, which is located in the src/grace source folder of your project (see Figure 21). You can also access this file by clicking the "Open Interrupt Vector File" link at the top of the Interrupt Vector List page.

Add your own code between the "USER CODE START" and the "USER CODE END" tags in the interrupt function. Do not make any modifications outside these tags, as they will be overwritten the next time you modify or save the *.cfg file.

If you click a button to "Remove Interrupt Handler Code", any code affected is moved to the end of the file and saved inside a #if 0 / #endif block that will not be executed.

main.cfg InterruptVectors_init.c

```
1 #include <msp430.h>
2
3 /* USER CODE START (section: InterruptVectors_init_c_prologue) */
4 /* User defined includes, defines, global variables and functions */
5 /* USER CODE END (section: InterruptVectors_init_c_prologue) */
6
7 /*
8  *  ======== InterruptVectors_graceInit ========
9  */
10 void InterruptVectors_graceInit(void)
11 {
12 }
13
14 /*
15  *  ======== Timer0_A3 Interrupt Service Routine ========
16  */
17 #pragma vector=TIMER0_A0_VECTOR
18 __interrupt void TIMER0_A0_ISR_HOOK(void)
19 {
20     /* USER CODE START (section: TIMER0_A0_ISR_HOOK) */
21     /* replace this comment with your code */
22     /* USER CODE END (section: TIMER0_A0_ISR_HOOK) */
23 }
24
25
26 /*
27  * ======== Preserved user code snippets ========
28  */
29 #if 0
30 /* USER CODE START (section: PORT1_ISR_HOOK) */
31     /* replace this comment with your code */
32     /* USER CODE END (section: PORT1_ISR_HOOK) */
33 #endif
34
```

Figure 21. Interrupt Vector Code

10 Revision History

This table lists the significant changes made in recent versions of this document.

Revision	Location	Additions/Modifications/Deletions
SLAU476A	Throughout	Updated document and figures to reflect Grace v3.0, which is used with Code Composer Studio v6.
	Section 2	To use Grace with CCS v6, install Grace using the CCS App Center.
	Section 4	Combined all G2xx and F2xx configuration into this section.
	Section 5	Combined all FR5xx configuration into this section.
	Section 5.1	Added section on pin mux (GPIO) configuration of FR5xx devices.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

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

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