



# TMS570LS Microcontrollers: Using the NHET as a General Purpose I/O Pin



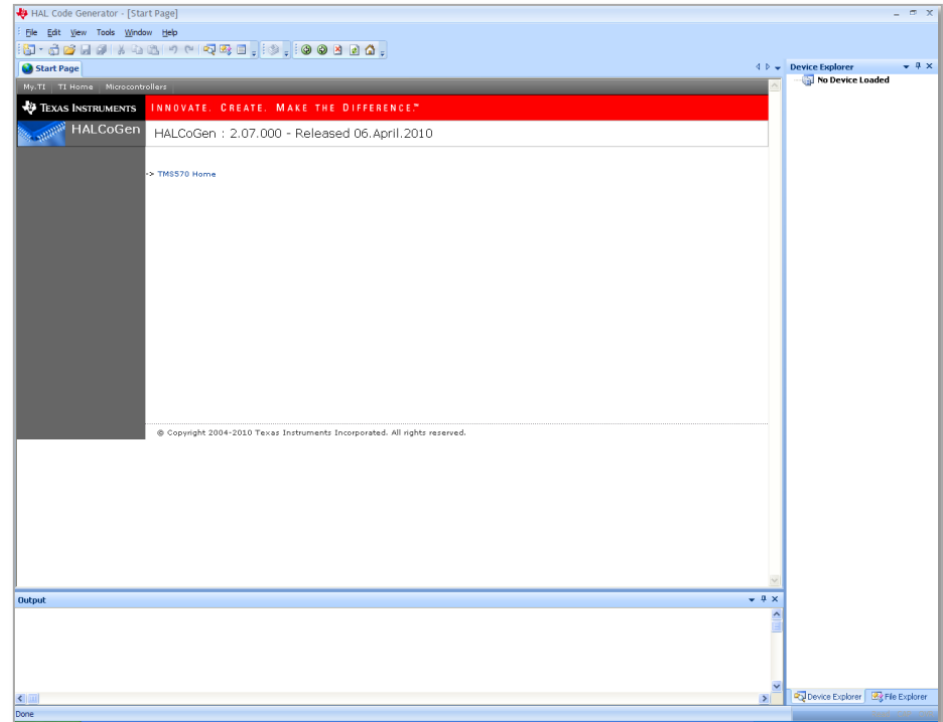
# Overview



- In this example we will:
  - Create a TMS570 HALCoGen Project
  - Generate and import code into Code Composer Studio
  - Write code to turn on the LED on NHET pin 1
  - Build, deploy and execute the code to the microcontroller
  
- Required Hardware:
  - Windows Based PC (WinXP, Vista, 7)
  - TMS570LS2x USB Development Stick or Microcontroller Development Kit
  
- Required Software:
  - TMS570 HALCoGen
  - Code Composer Studio v4.x

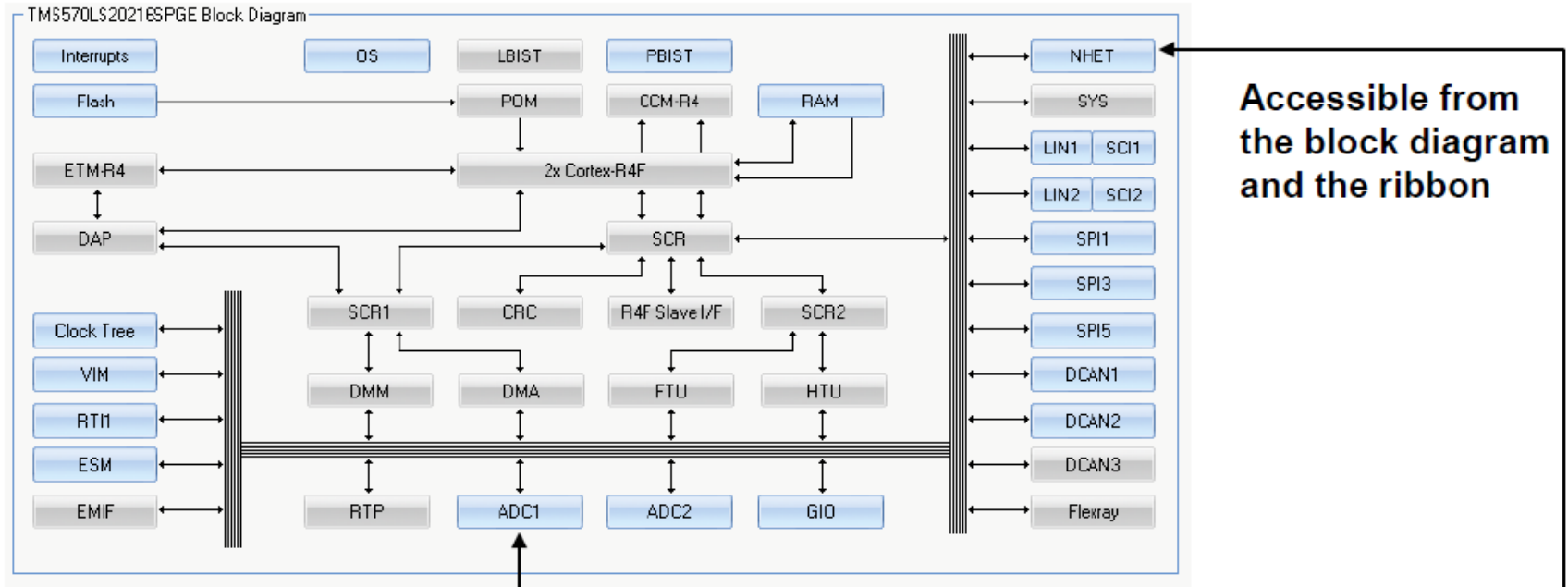
# Setting up a New HALCoGen Project

- Launch HALCoGen
  - Start → Programs → Texas Instruments → HALCoGen
- File > New > Project
- Family:
  - TMDX570
- Device:
  - TMDX570LS20USB (for USB Stick)
  - OR**
  - TMDX570LS20MDK (for MDK)
- Name: Exercise
- Location: “C:\myWorkspace”

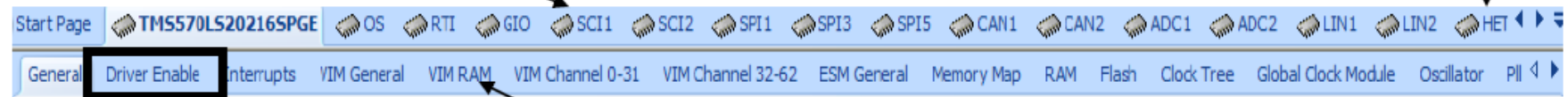


For more help with HALCoGen, see this getting started video: [LINK](#)

# The HALCoGen Interface



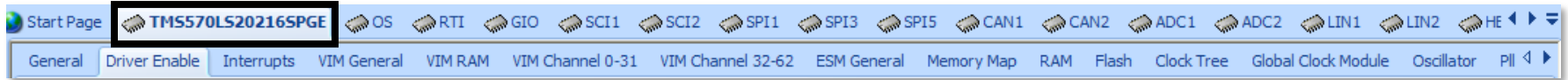
## Top Level: Peripherals on the TMS570



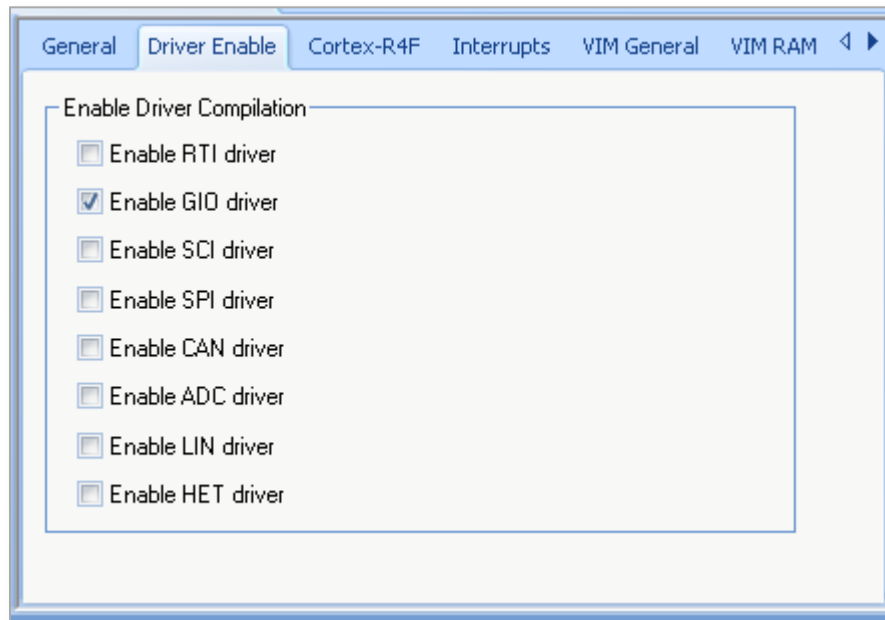
Click "Driver Enable"

## Bottom Level: Settings for each peripheral

# Configuring the Peripherals

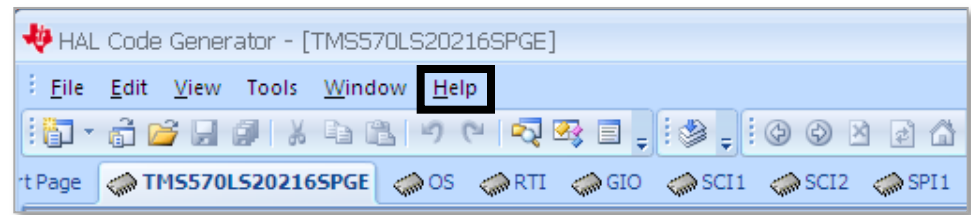


- Select the peripherals that are required for this project.
  - In this lab we need only enable the GIO driver, uncheck all other drivers



- No further changes should be made, the source code can now be generated.
  - To do this go to File → Generate Code
  - Following, the folders on the right will populate with our new files

# HALCoGen Help



**C:/HALCoGen/gio.c File Reference**

GIO Driver Implmnetation File. [More...](#)

```
#include "gio.h"
```

Include dependency graph for gio.c:

```
graph TD
    C["C:/HALCoGen/gio.c"] --> H["gio.h"]
```

Go to the source code of this file.

**Functions**

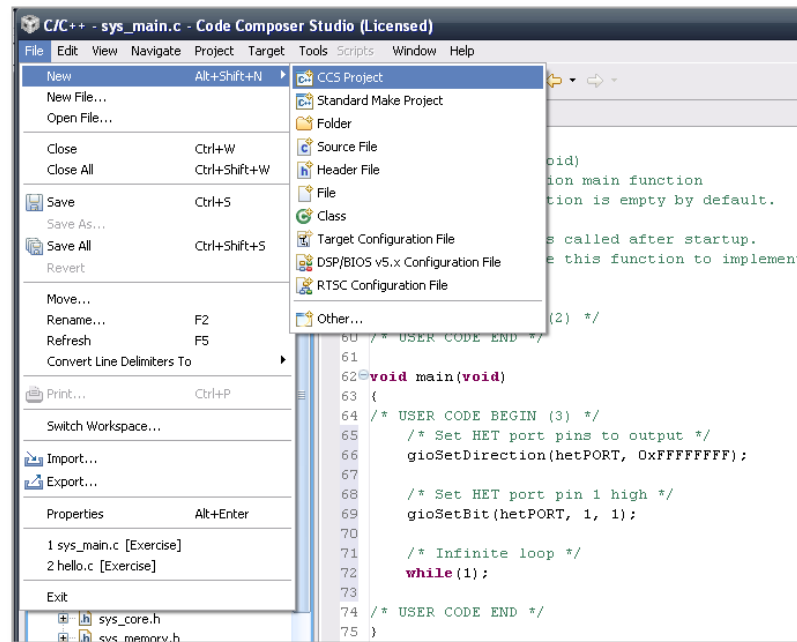
void	<b>gioInit</b> (void)	Initializes the GIO Driver.
void	<b>gioSetDirection</b> (gioPORT_t *port, unsigned dir)	Set Port Direction.
void	<b>gioSetBit</b> (gioPORT_t *port, unsigned bit, unsigned value)	Write Bit.
void	<b>gioSetPort</b> (gioPORT_t *port, unsigned value)	Write Port Value.
unsigned	<b>gioGetBit</b> (gioPORT_t *port, unsigned bit)	Read Bit.

- Information about the files and functions that HALCoGen creates can be found in the HALCoGen 'Help' menu
- Help can be launched from the main title bar under Help → Help Topics

# Setting up Code Composer Studio 4



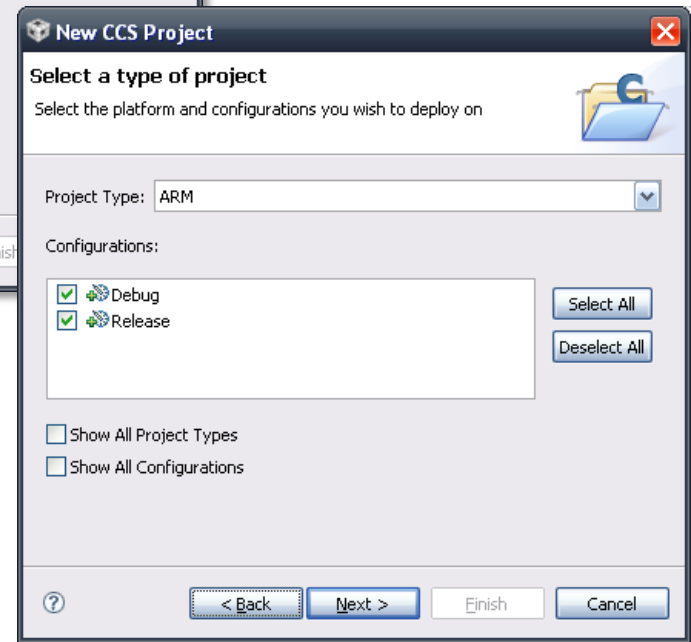
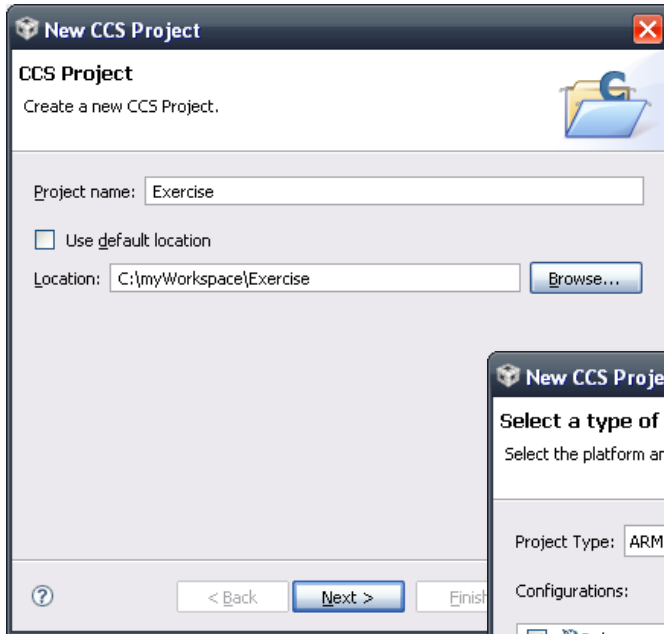
- Launch Code Composer Studio v4.x (CCS)
  - Start → Programs → Texas Instruments → Code Composer Studio v4 → Code Composer Studio v4
- When it launches, CCS will ask you to select a workspace, we will chose “C:\myWorkspace”
- Once it loads, go to File → New → CCS Project



For more help with Code Composer Studio, see this getting started video: [LINK](#)

# Setting up our Project

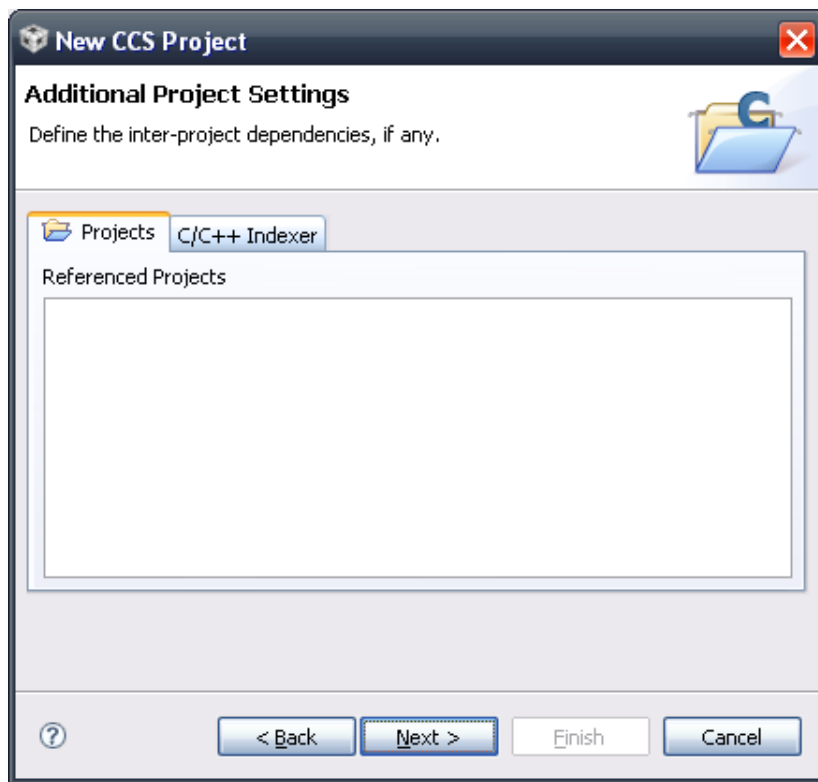
- Our project name needs to match the name of our HALCoGen Project, Exercise
- Then Click “next”
- On the next page, make sure that your project type is set to ARM and Debug and Release are both checked
- Then Click “next”





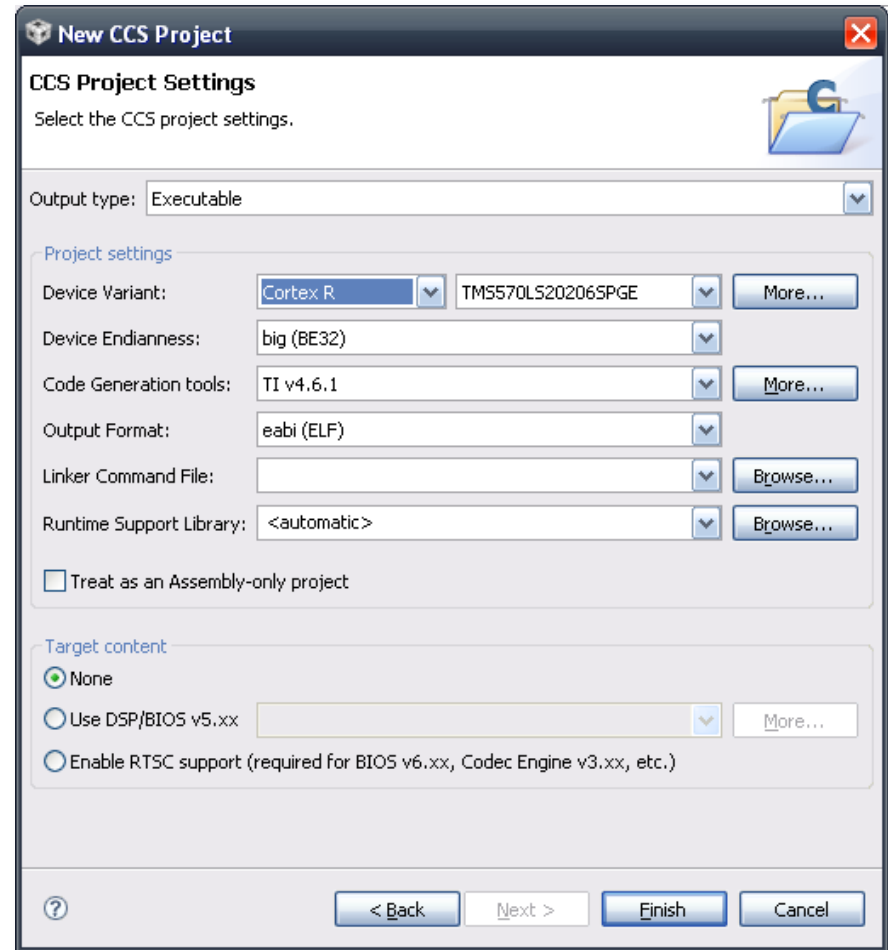
# Setting up our Project (cont.)

- We are not using any referenced projects so click “next” again



# Setting up the Project (cont.)

- Lastly, set the Device Variant to “Cortex R” and TMS570LS20216SPGE
- Click “Finish”



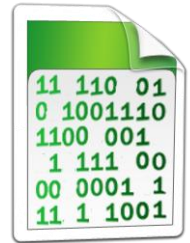
# Getting Started



- On the left hand side in the “C/C++ Projects” explorer, open “sys\_main.c”
- When ever you generate code in HALCoGen, the program overwrites user code, except specific sections marked by “USER CODE BEGIN (x)” and “USER CODE END”
  - For code placement we will be referring to the number within the User Code block

```
/* USER CODE BEGIN (0) */  
/* USER CODE END */
```

# Writing the Code



- Inside User Code 1, copy the code below.

```
/* USER CODE BEGIN (1) */  
#include "het.h"  
/* USER CODE END */
```

- Then in User Code 3, copy the code below.

```
/* USER CODE BEGIN (3) */  
/* Set HET port pins to output */  
gioSetDirection(hetPORT, 0xFFFFFFFF);  
  
/* Set HET port pin 1 high */  
gioSetBit(hetPORT, 1, 1);  
  
/* Infinite loop */  
while(1);  
/* USER CODE END */
```

# Notifications

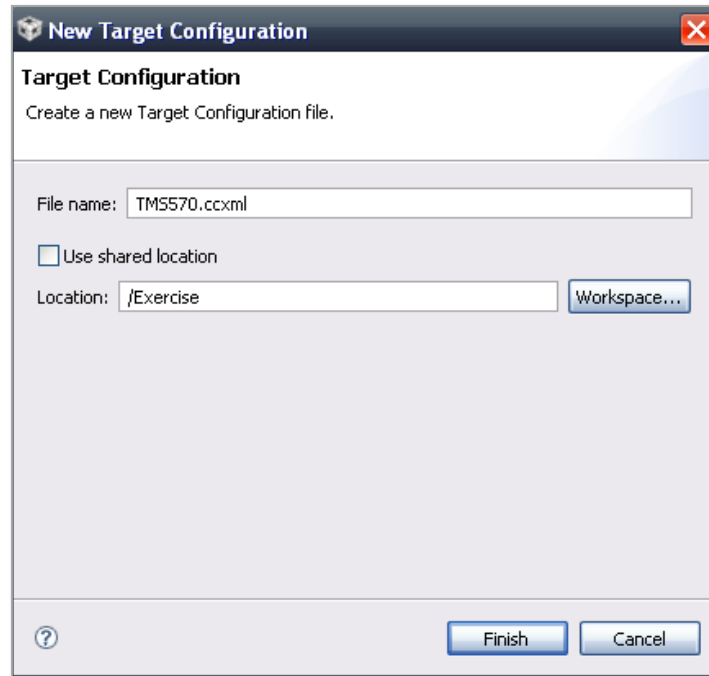
- Lastly we must insert a function that would be called if interrupts were enabled. Without these, the code will fail to build

```
/* USER CODE BEGIN (4) */  
/* GIO Notification function not used, but required by compiler */  
void gioNotification(int bit)  
{  
    return;  
}  
/* USER CODE END */
```



# 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



# Creating a Target Configuration...

- A new tab will appear with a list of emulators and devices.
  - Connection: Texas Instruments XDS100v2 USB Emulator
  - In the text box labeled “Type Filter Text”, type “TMS570”.
    - This will narrow the search down to just TMS570 devices, select TMS570LS20216SPGE
  - Click “Save” on the right

Connection: Texas Instruments XDS100v2 USB Emulator

Device: TMS570

- TMS570LS20206SPGE
- TMS570LS20206SZWT
- TMS570LS20216SPGE
- TMS570LS20216SZWT

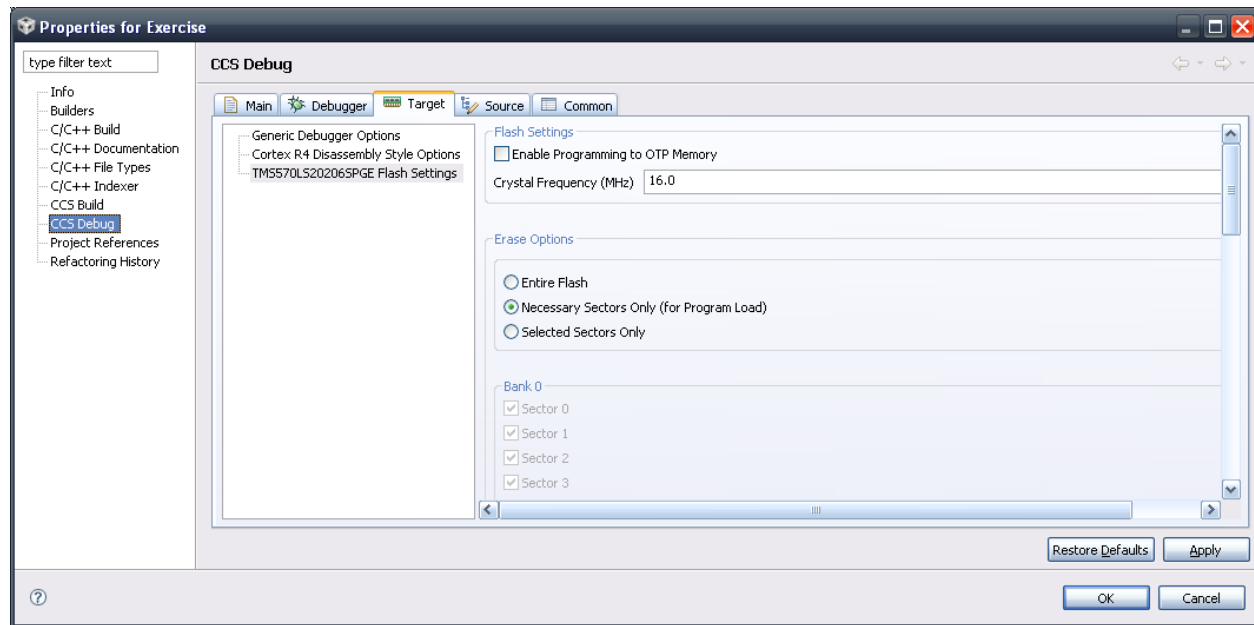


# Flash Programming Configuration

- 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.

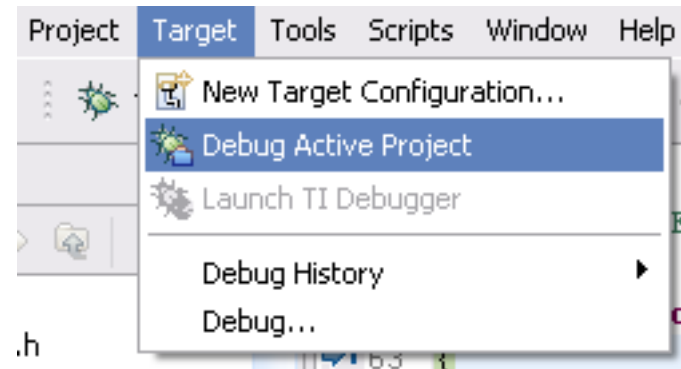
## NOTE:

This option is only available in CCSv4.2 and newer



# 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.





# For More TMS570 Information



- TMS570 Web Page: [www.ti.com/TMS570](http://www.ti.com/TMS570)
  - Data Sheets
  - Technical Reference Manual
  - Application Notes
  - Software & Tools Downloads and Updates
  - Order Evaluation and Development Kits
- TMS570 Forum:  
<http://e2e.ti.com/support/microcontrollers/tms570/default.aspx>
  - News and Announcements
  - Useful Links
  - Ask Technical Questions
  - Search for Technical Content
- TMS570 WIKI:  
<http://processors.wiki.ti.com/index.php/Category:TMS570>
  - How to guides
  - Intro Videos
  - General Information



# Thank You!

For completing this TMS570 example

