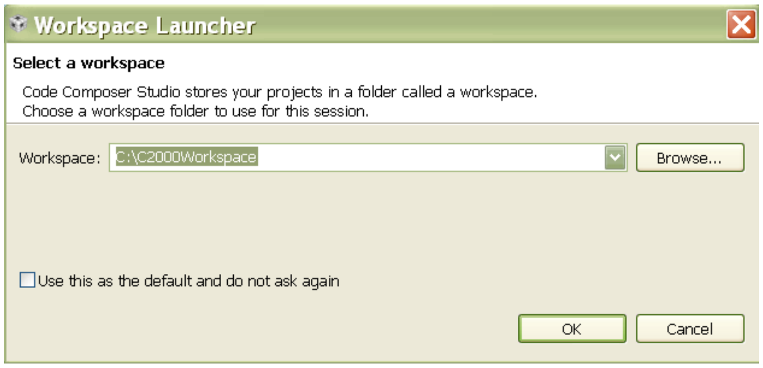
Lab 1: Guide for creating portable projects in C2000

Requisites: CCSV5.5 installed in C:\TI; ControlSUITE installed in C:\TI; C2000 PICCOLO Launchpad hardware (LAUNCHXL-F28027).

Objective: To create first project with CCS and debug. This lab will use CPU Timers and blink an LED connected to GPIO.

Start CCS IDE. When you open CCS, you will get the "Workspace Launcher" window. Type in C:\C2000Workspace



Workspace Launcher

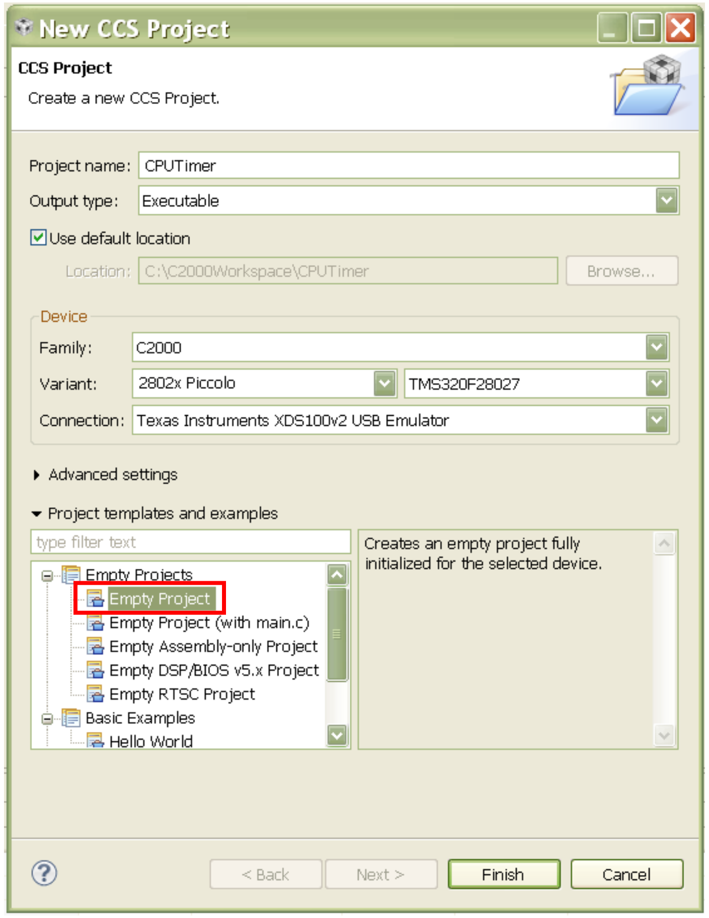
Select a workspace

Code Composer Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\C2000Workspace

☐ Use this as the default and do not ask again

- Close the resource explorer window and Click on File>New>New CCS Project to create a new project.
- Enter the project name and select the microcontroller and Connections settings as shown below:



New CCS Project

CCS Project

Create a new CCS Project.

Project name: CPUTimer

Output type: Executable

☑ Use default location

Location: C:\C2000Workspace\CPUTimer

Device

Family: C2000

Variant: 2802x Piccolo TMS320F28027

Connection: Texas Instruments XDS100v2 USB Emulator

Advanced settings

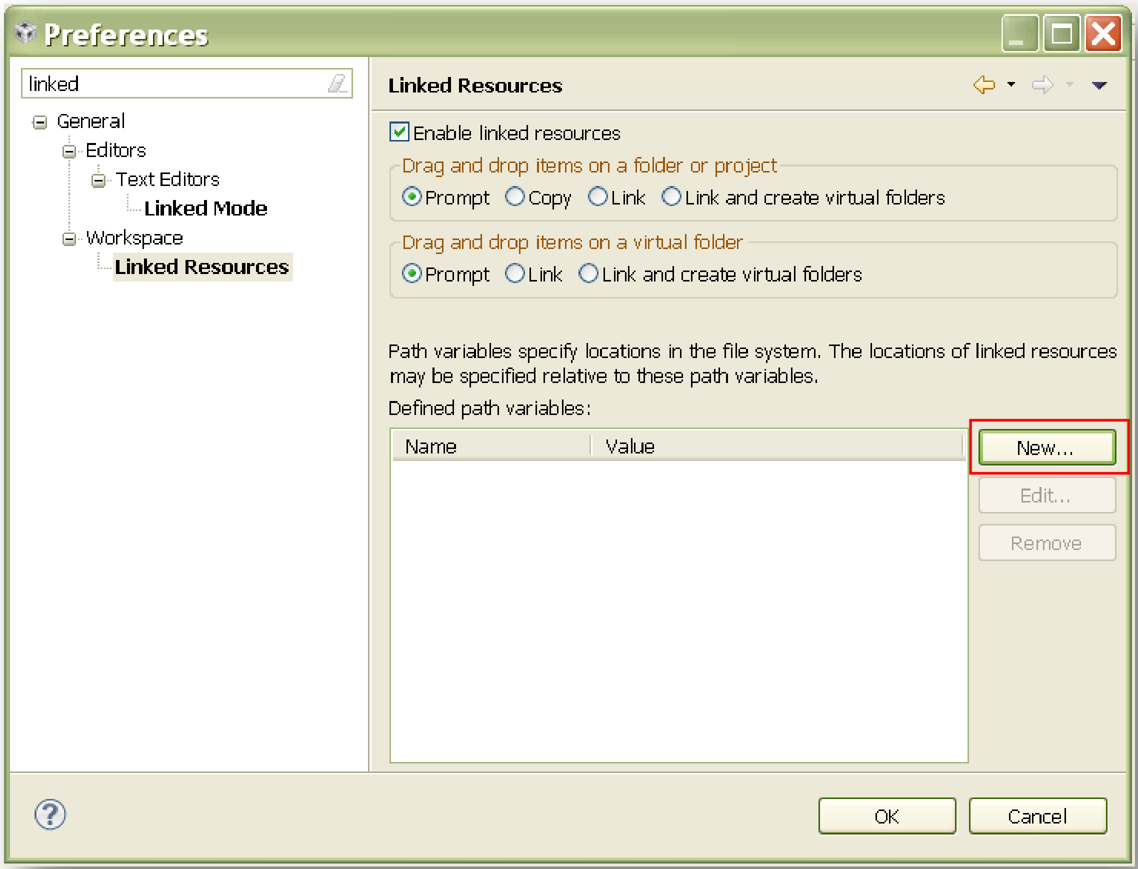
Project templates and examples

Empty Projects
- Empty Project
- Empty Project (with main.c)
- Empty Assembly-only Project
- Empty DSP/BIOS v5.x Project
- Empty RTSC Project

Basic Examples
- Hello World

Creates an empty project fully initialized for the selected device.

After creating the project, click on Window>Preferences to edit the project preferences. Type "Linked" in the search bar and select Linked resources.


Preferences

linked

- General
  - Editors
    - Text Editors
      - Linked Mode
  - Workspace
    - Linked Resources

Linked Resources

[✓] Enable linked resources

Drag and drop items on a folder or project: [✓ Prompt] Copy Link Link and create virtual folders

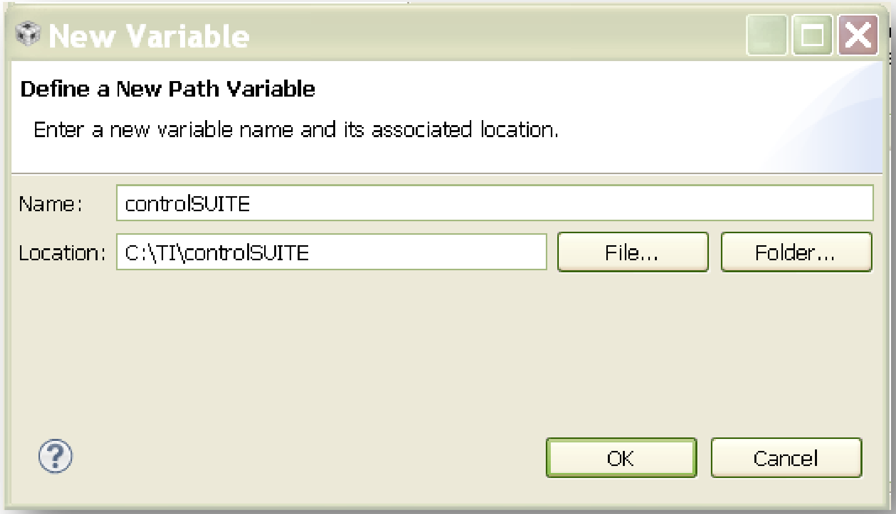
Drag and drop items on a virtual folder: [✓ Prompt] Link Link and create virtual folders

Path variables specify locations in the file system. The locations of linked resources may be specified relative to these path variables.

Defined path variables:

Name	Value

Click on "New" to add a new path variable and Name it "controlSUITE". Give the path location as C:\ti\controlSUITE
Note: In case if your controlSUITE installation path is different, enter the location of your controlSUITE install folder in the location field. Click OK when done.


New Variable

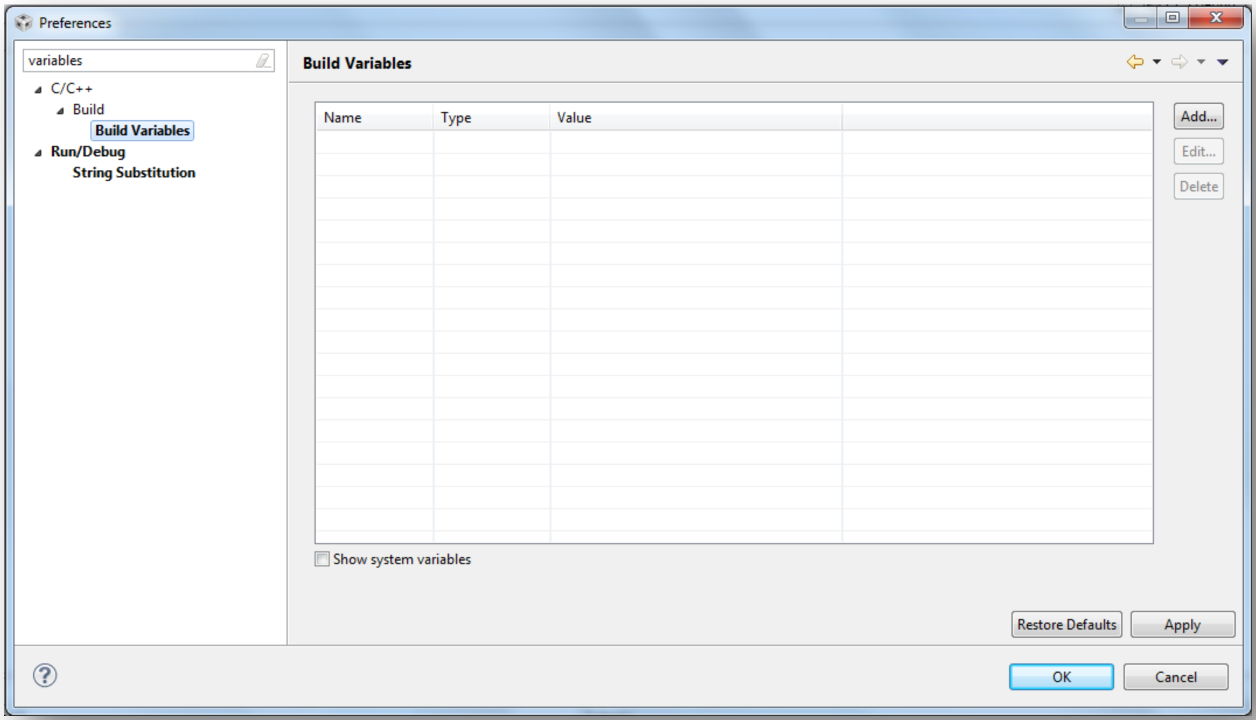
Define a New Path Variable

Enter a new variable name and its associated location.

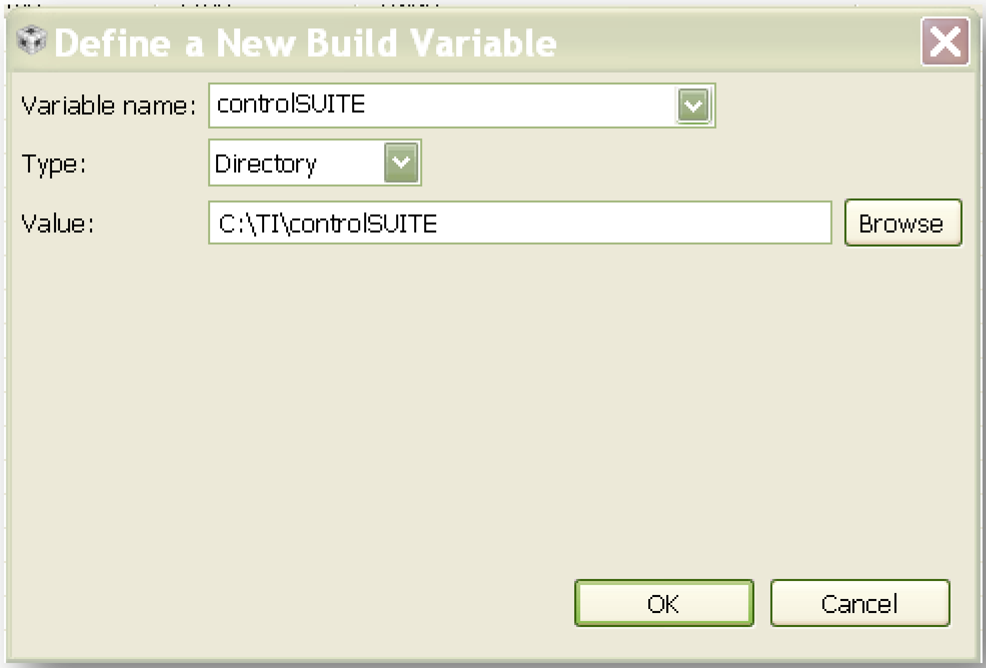
Name: controlSUITE

Location: C:\TI\controlSUITE

Go to window>Preferences again, and type "Variables" in the search field. Select "Build Variables" and click "Add" to add controlSUITE as a build variable as shown below:



Enter the variable name and path as shown below. Be sure to select the Type as "Directory". Note: In case if your controlSUITE installation path is different, enter the location of your controlSUITE install folder in the "Value" field. Click OK when done.


Define a New Build Variable

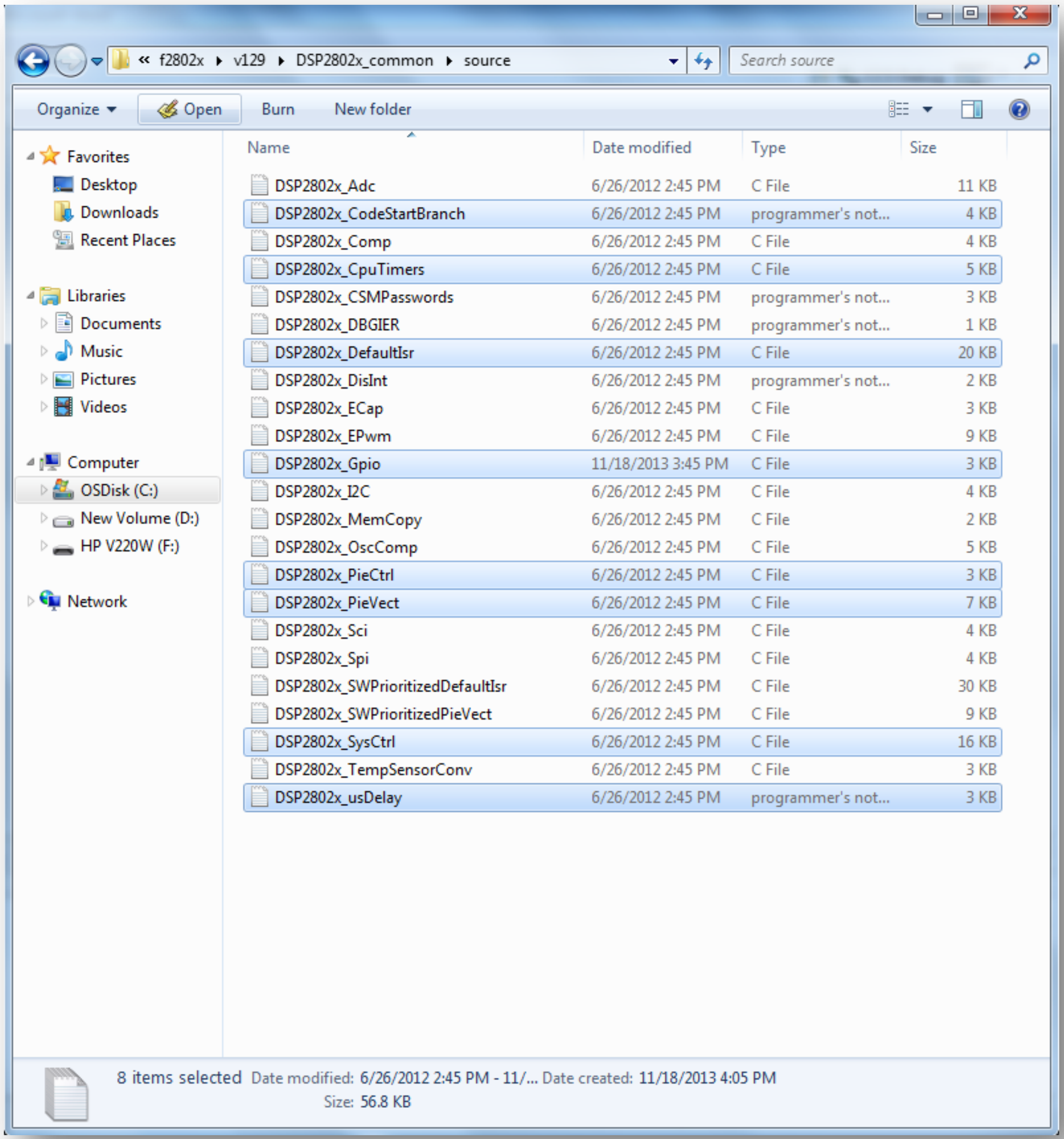
Variable name: controlSUITE

Type: Directory

Value: C:\TI\controlSUITE

Open Windows Explorer and navigate to
C:\ti\controlSUITE\device_support\f2802x\v129\DSP2802x_common\source
Select the following files and drag and drop them into the CCS project.

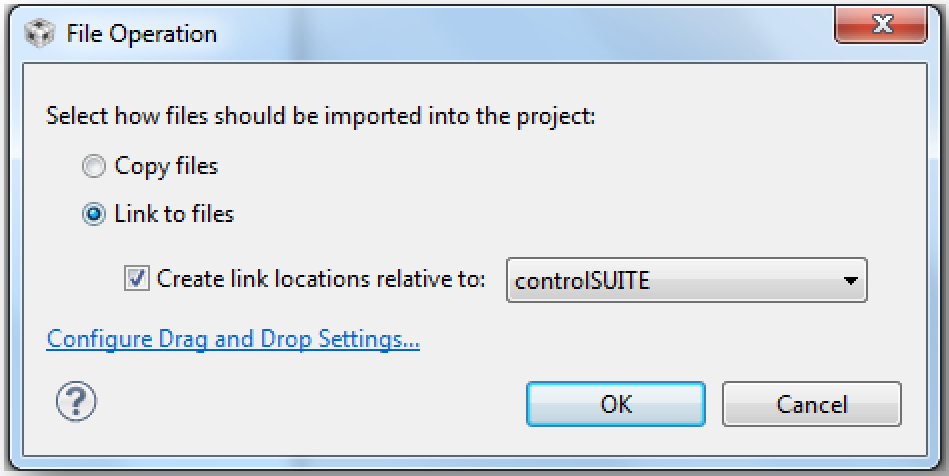
- DSP2802x_CodeStartBranch.asm
- DSP2802x _CpuTimers.c
- DSP2802x _DefaultIsr.c
- DSP2802x_Gpio.c
- DSP2802x _PieCtrl.c
- F2806x_PieVect.c
- DSP2802x _SysCtrl.c
- DSP2802x _usDelay.asm



Name	Date modified	Type	Size
DSP2802x_Adc	6/26/2012 2:45 PM	C File	11 KB
DSP2802x_CodeStartBranch	6/26/2012 2:45 PM	programmer's not...	4 KB
DSP2802x_Comp	6/26/2012 2:45 PM	C File	4 KB
DSP2802x_CpuTimers	6/26/2012 2:45 PM	C File	5 KB
DSP2802x_CSMPasswords	6/26/2012 2:45 PM	programmer's not...	3 KB
DSP2802x_DBGIER	6/26/2012 2:45 PM	programmer's not...	1 KB
DSP2802x_DefaultIsr	6/26/2012 2:45 PM	C File	20 KB
DSP2802x_DisInt	6/26/2012 2:45 PM	programmer's not...	2 KB
DSP2802x_ECap	6/26/2012 2:45 PM	C File	3 KB
DSP2802x_EPwm	6/26/2012 2:45 PM	C File	9 KB
DSP2802x_Gpio	11/18/2013 3:45 PM	C File	3 KB
DSP2802x_I2C	6/26/2012 2:45 PM	C File	4 KB
DSP2802x_MemCopy	6/26/2012 2:45 PM	C File	2 KB
DSP2802x_OscComp	6/26/2012 2:45 PM	C File	5 KB
DSP2802x_PieCtrl	6/26/2012 2:45 PM	C File	3 KB
DSP2802x_PieVect	6/26/2012 2:45 PM	C File	7 KB
DSP2802x_Sci	6/26/2012 2:45 PM	C File	4 KB
DSP2802x_Spi	6/26/2012 2:45 PM	C File	4 KB
DSP2802x_SWPrioritizedDefaultIsr	6/26/2012 2:45 PM	C File	30 KB
DSP2802x_SWPrioritizedPieVect	6/26/2012 2:45 PM	C File	9 KB
DSP2802x_SysCtrl	6/26/2012 2:45 PM	C File	16 KB
DSP2802x_TempSensorConv	6/26/2012 2:45 PM	C File	3 KB
DSP2802x_usDelay	6/26/2012 2:45 PM	programmer's not...	3 KB

8 items selected Date modified: 6/26/2012 2:45 PM - 11/... Date created: 11/18/2013 4:05 PM
Size: 56.8 KB

When prompted, select “Link to files” as follows, and select “controlSUITE” in the drop down menu:


File Operation

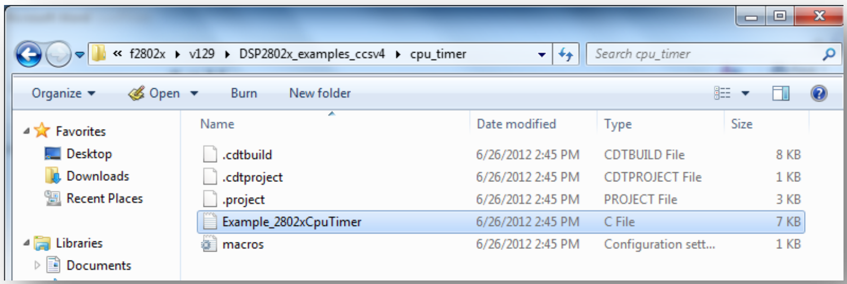
Select how files should be imported into the project:

Copy files [✓ Link to files]

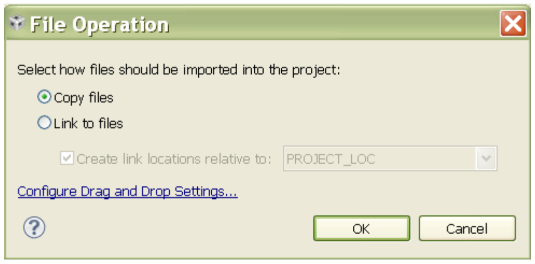
[✓] Create link locations relative to: controlSUITE

Similarly,
Go to C:\ti\controlSUITE\device_support\f2802x\v129\DSP2802x_headers\source
Add the file DSP2802x_GlobalVariableDefs.c
Go to C:\ti\controlSUITE\device_support\f2802x\v129\DSP2802x_headers\cmd
Add the file DSP2802x_Headers_nonBIOS.cmd

To add the main example C file, go to
C:\ti\controlSUITE\device_support\f2802x\v129\DSP2802x_examples_ccsv4\cpu_timer
and select the file Example_2802x_CpuTimer.c
Note: While adding this file to the project, copy it instead of linking, as we will be modifying this file during our exercise.


« f2802x ▸ v129 ▸ DSP2802x_examples_ccsv4 ▸ cpu_timer

Name	Date modified	Type	Size
.cdtbuild	6/26/2012 2:45 PM	CDTBUILD File	8 KB
.cdtproject	6/26/2012 2:45 PM	CDTPROJECT File	1 KB
.project	6/26/2012 2:45 PM	PROJECT File	3 KB
Example_2802xCpuTimer	6/26/2012 2:45 PM	C File	7 KB
macros	6/26/2012 2:45 PM	Configuration sett...	1 KB

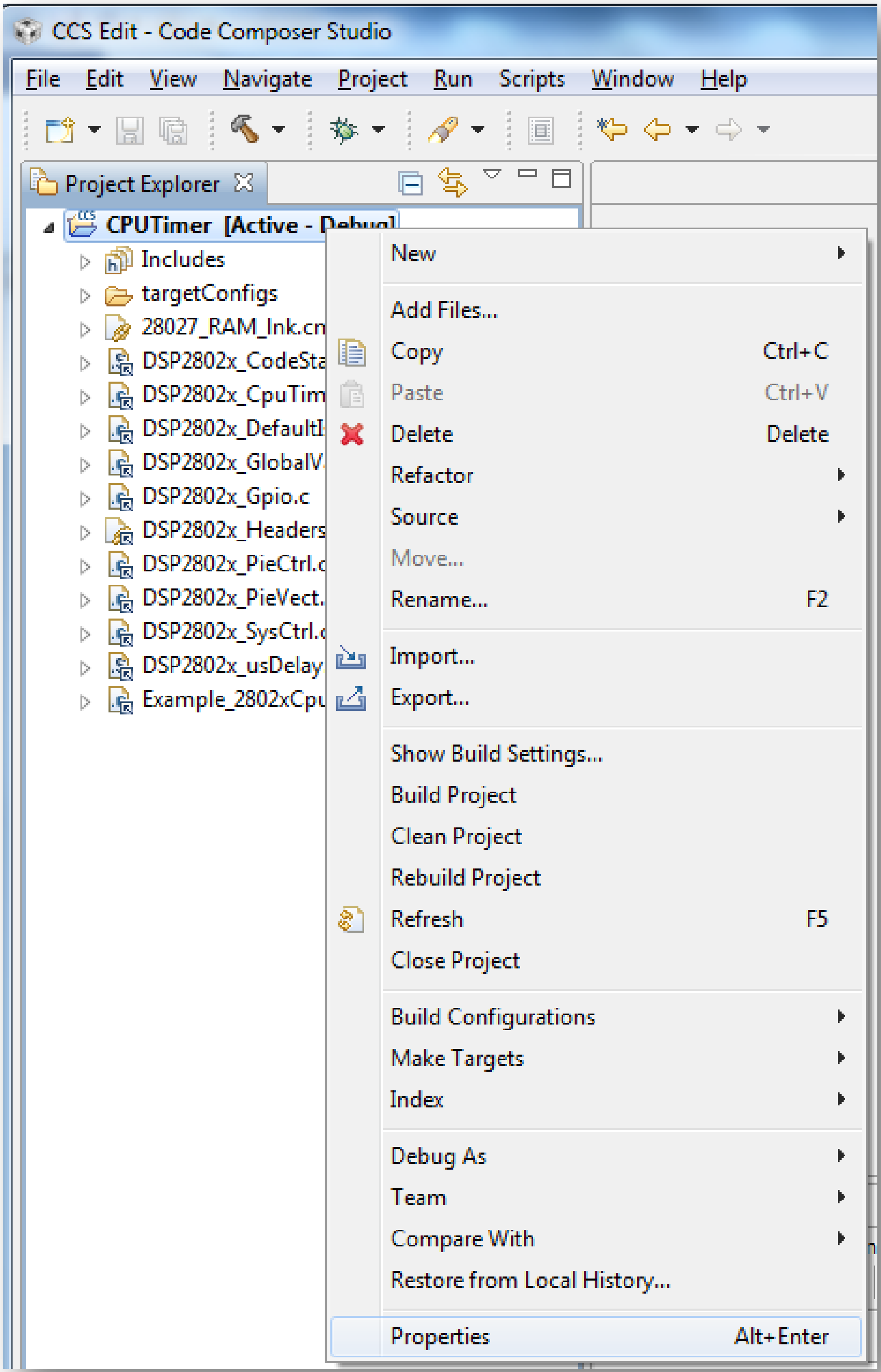

File Operation

Select how files should be imported into the project:

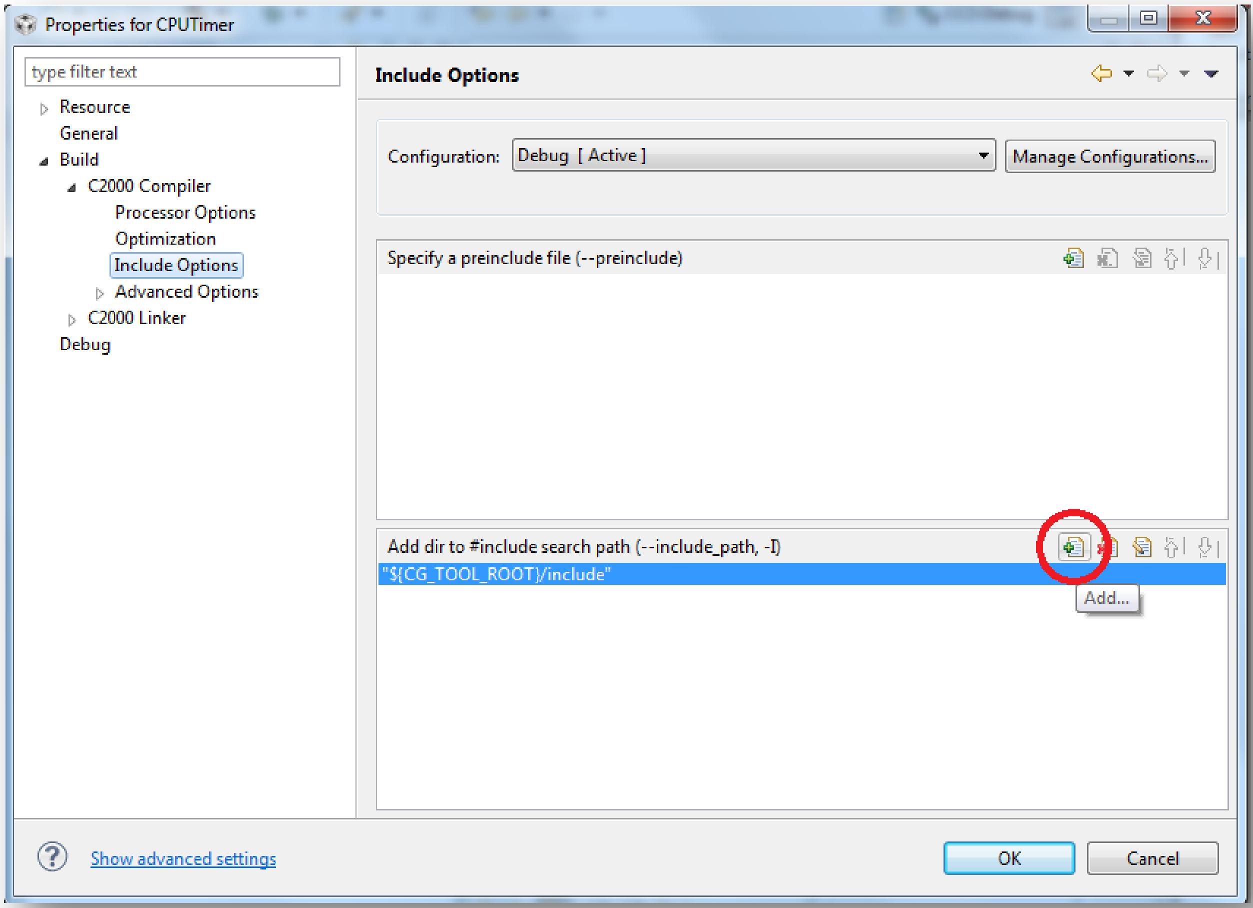
[✓ Copy files] Link to files

[✓] Create link locations relative to: PROJECT_LOC

Right click on the project and select “Properties”



Navigate to Build>C2000 Compiler>Include Options
Click the "Add" icon to add the following include options


Properties for CPUTimer

Include Options

Configuration: Debug [Active]

Specify a preinclude file (--preinclude)

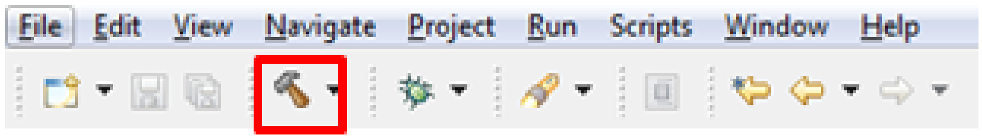
Add dir to #include search path (--include_path, -I)

"${CG_TOOL_ROOT}/include"

Add the following paths:
${controlSUITE}\device_support\f2802x\v129\DSP2802x_headers\include
${controlSUITE}\device_support\f2802x\v129\DSP2802x_common\include

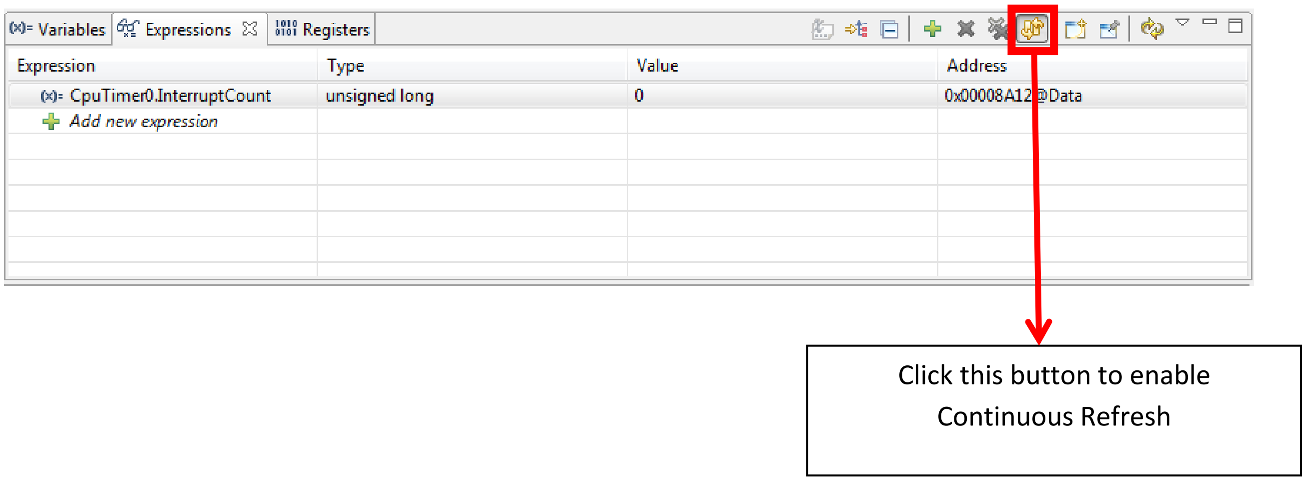
Note: ${controlSUITE} uses the build variable "controlSUITE" to search for the file path. We had defined "controlSUITE" as C:\ti\controlSUITE

We are done with creating a project!. Now let's Build and Debug the program.
Build the project by clicking on the "Build" Icon (Hammer symbol)



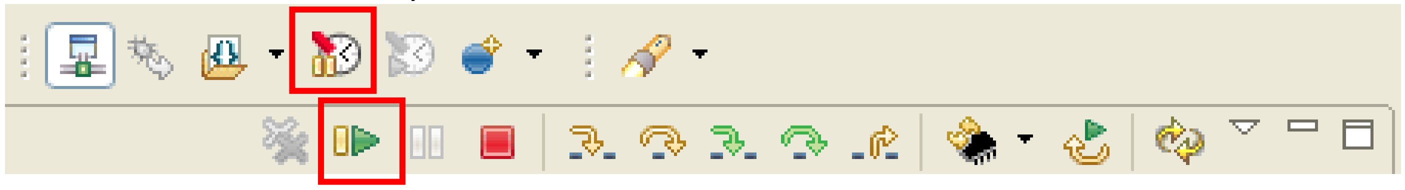
Open file “Example_2802xCpuTimer.c” and go through the code.
Click on the “Debug” button which is next to the Build button, to enter the debug session.

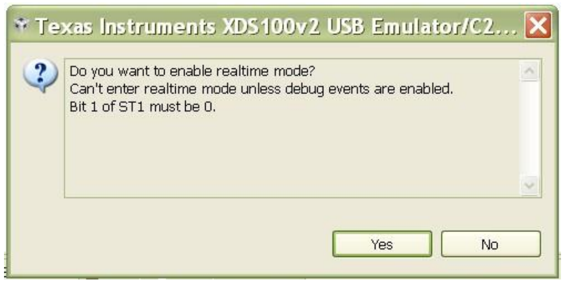
Scroll to line 178 of the file “Example_2802xCpuTimer.c” and copy the variable “CpuTimer0.InterruptCount” and Paste it in the “Expressions” window. You can also, select the variable “CpuTimer0.InterruptCount”, right click on it and select “Add Watch Expression…”



Expression	Type	Value	Address
CpuTimer0.InterruptCount	unsigned long	0	0x00008A12@Data
Add new expression			

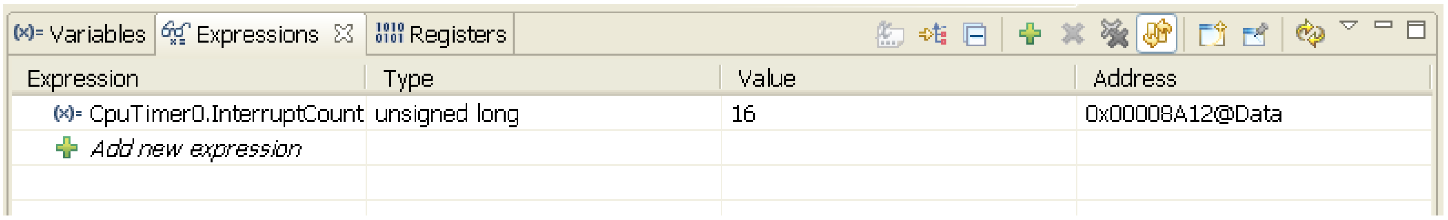
Enable “Silicon Realtime Debug mode enable” and then click on the green “Resume”(F8) button to start program execution. While enabling realtime debug, you would get a confirmation Window and you select “Yes”.





Do you want to enable realtime mode?
Can't enter realtime mode unless debug events are enabled.
Bit 1 of ST1 must be 0.

Now the program will be executing and you can see the CpuTimer0.InterruptCount getting incremented every second. To terminate, press on the Red Stop button.



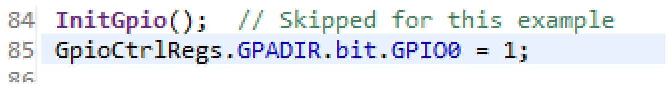
Expression	Type	Value	Address
CpuTimer0.InterruptCount	unsigned long	16	0x00008A12@Data
Add new expression			

Part 2:

Objective: Toggle the LED connected on GPIO 0 pin on every second.

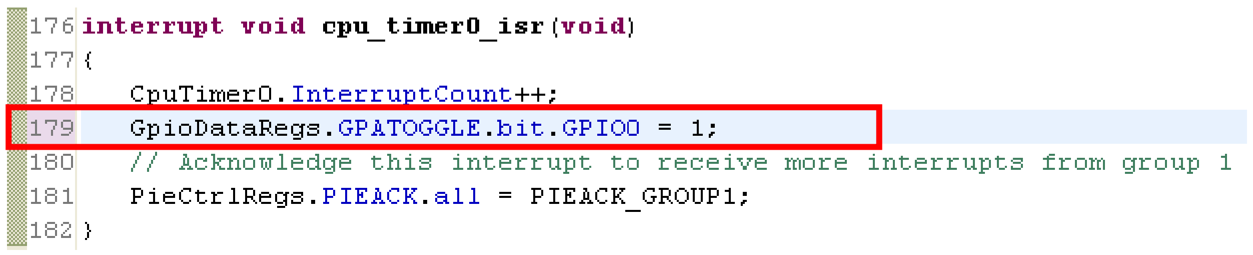
In the file “Example_2802xCpuTimer.c” In the, scroll to line 84 and uncomment the “InitGpio()” function

Write the following code to enable GPIO0 as an output pin (GpioCtrlRegs.GPADIR.bit.GPIO0=1;). Note: After entering the group name succeeded by dot (GpioCtrlRegs.), you can press “CTRL+SPACE” keys to see the list of all group members.



```
84 InitGpio();  // Skipped for this example
85 GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;
```

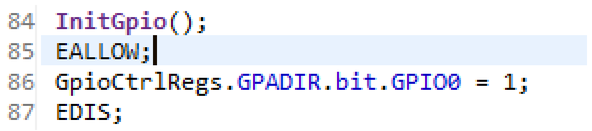
Scroll to line 179 and add the following code to toggle an LED (GpioDataRegs.GPATOGGLE.bit.GPIO0 = 1;).



```
176 interrupt void cpu_timer0_isr(void)
177 {
178    CpuTimer0.InterruptCount++;
179    GpioDataRegs.GPATOGGLE.bit.GPIO0 = 1;
180    // Acknowledge this interrupt to receive more interrupts from group 1
181    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
182 }
```

Build the program, click on debug and then run. Is the LED toggling?...

The LED will NOT toggle. Note that, the GPIO control registers are EALLOW protected. Exit the debug session and add the EALLOW and EDIS statements in the GPIO initialization code as follows:



```
84 InitGpio();
85 EALLOW;
86 GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;
87 EDIS;
```

Now rebuild the code and execute. You will see the LED at GPIO0 toggling.

Add “CpuTimer0Regs.PRD.all” variable to the expressions window. The value is 60000000. Change this value from the watch window to 20000000 and observe the LED blink rate.

Exercise: Try toggling the LEDs connected at GPIO1, GPIO2 and GPIO3 also.