Adding controlSUITE example code and drivers to MotorWare projects

Introduction

This documents how to add controlSUITE example code and peripheral drivers into a MotorWare project. We will show the procedure for F2802x devices using MotorWare proj_lab03a, but you can follow the same process for F2805x and F2806x devices and any of the MotorWare labs in any MotorWare version.

For this example we are using the MotorWare project:

\sw\solutions\instaspin_foc\boards\boostxldrv8305_revA\f28x\f2802xF\projects\ccs5\proj_lab03a

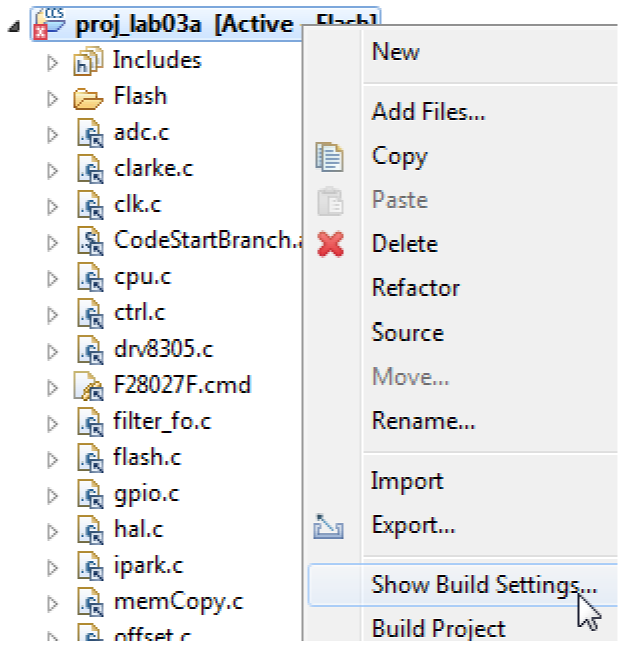
controlSUITE example code

The objective is to allow a MotorWare project to use example code from controlSUITE. For this example we will use:

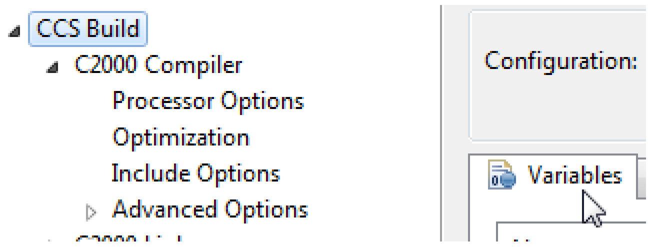
C:\ti\controlSUITE\device_support\f2802x\v230\f2802x_examples_structs\cpu_timer

Step 1. Add controlSUITE include folders to the MotorWare project

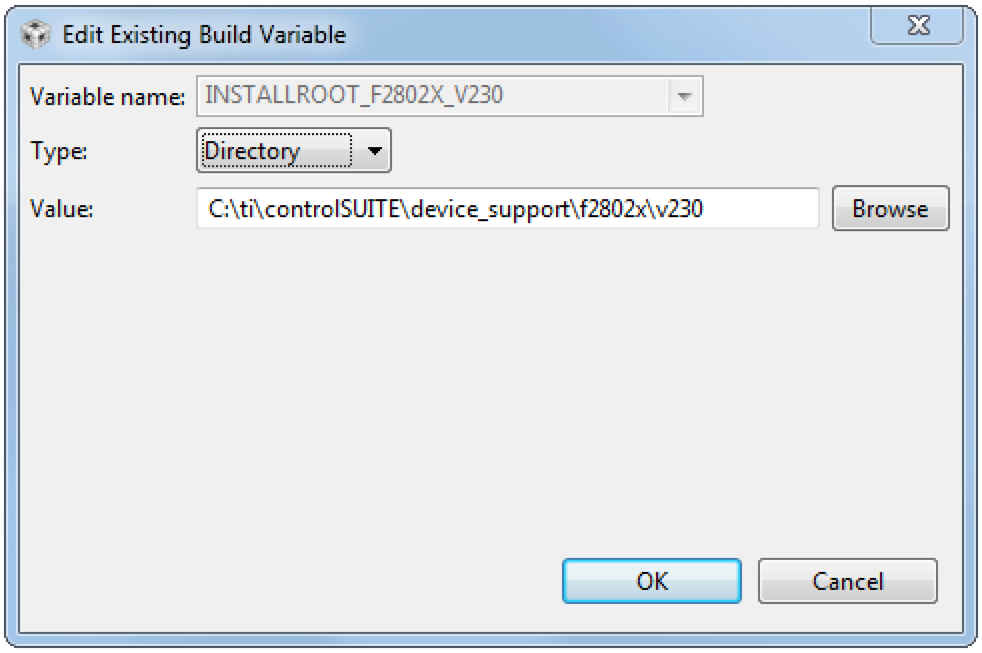
1. Right click and open properties of the project:



2. Go to variables



3. Add a new variable where the controlSUITE code for 2x is located:
   Variable name: INSTALLROOT_F2802X_V230
   Value: C:\ti\controlSUITE\device_support\f2802x\v230



Edit Existing Build Variable

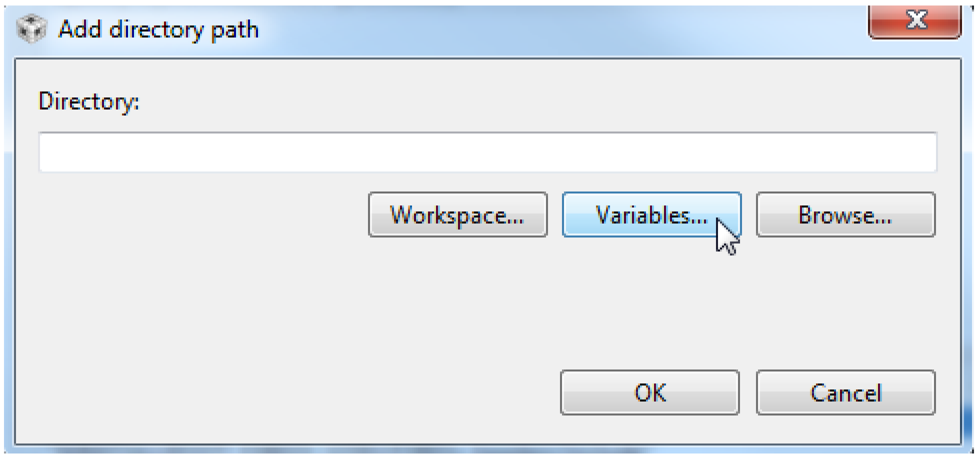
Variable name: INSTALLROOT_F2802X_V230

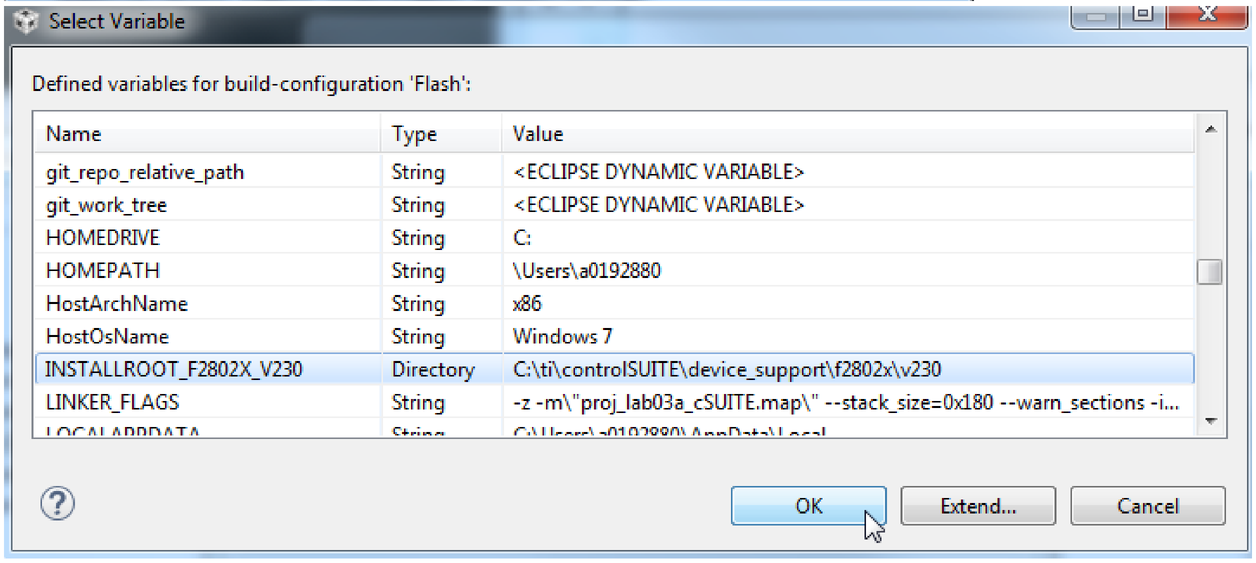
Type: Directory

Value: C:\ti\controlSUITE\device_support\f2802x\v230

4. Add two new include paths
   "${INSTALLROOT_F2802X_V230}/F2802x_common/include"
   "${INSTALLROOT_F2802X _V230}/F2802x_headers/include"

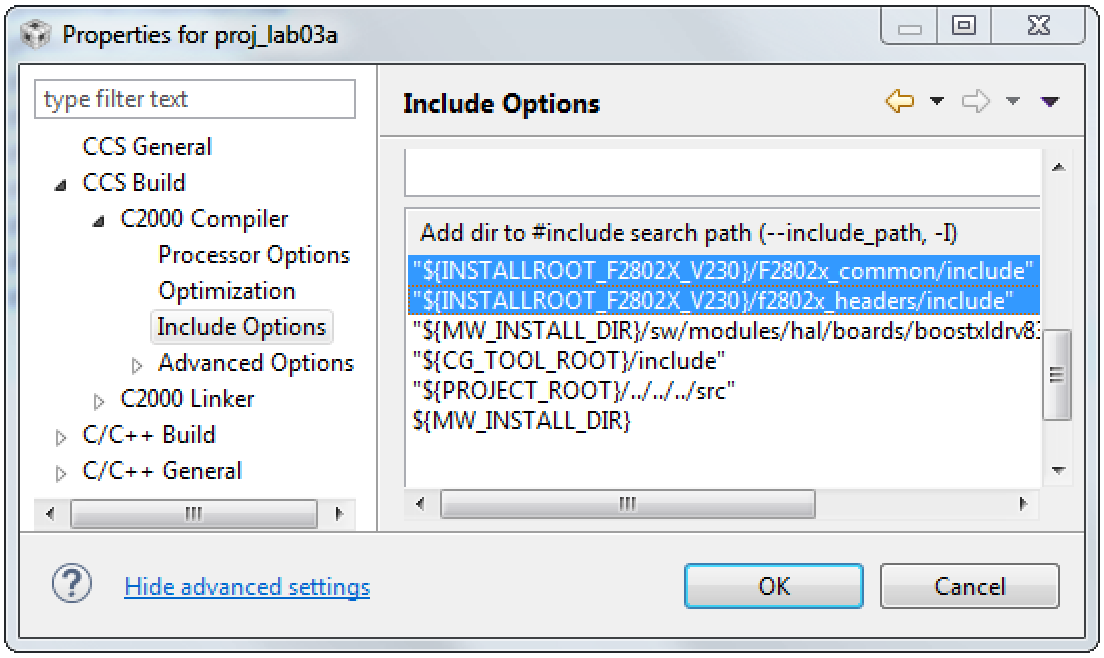
   Do not copy and paste the above text, CCS has an issue with this working consistently. Instead use the procedure below that uses the “Variables” button to select INSTALLROOT_F2802X _V230 from the displayed list, and then complete the path by entering the required text as shown above.




Select Variable

Defined variables for build-configuration 'Flash':

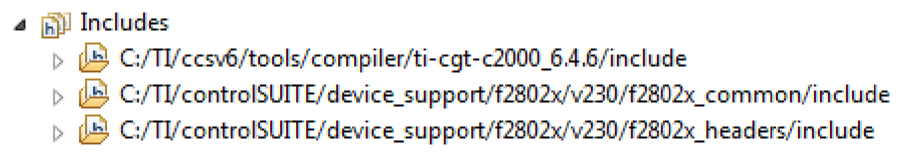
Name	Type	Value
git_repo_relative_path	String	<ECLIPSE DYNAMIC VARIABLE>
git_work_tree	String	<ECLIPSE DYNAMIC VARIABLE>
HOMEDRIVE	String	C:
HOMEPATH	String	\Users\a0192880
HostArchName	String	x86
HostOsName	String	Windows 7
INSTALLROOT_F2802X_V230	Directory	C:\ti\controlSUITE\device_support\f2802x\v230
LINKER_FLAGS	String	-z -m\"proj_lab03a_cSUITE.map\" --stack_size=0x180 --warn_sections -i...
LOCALAPPDATA	String	C:\Users\a0192880\AppData\Local


Properties for proj_lab03a

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

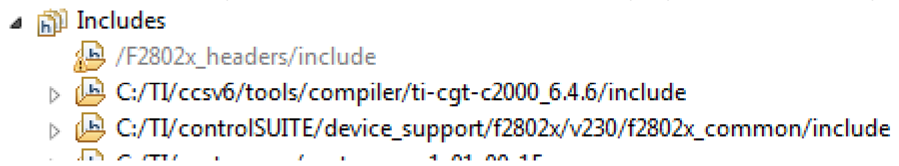
```
"${INSTALLROOT_F2802X_V230}/F2802x_common/include"
"${INSTALLROOT_F2802X_V230}/f2802x_headers/include"
"${MW_INSTALL_DIR}/sw/modules/hal/boards/boostxldrv8:
"${CG_TOOL_ROOT}/include"
"${PROJECT_ROOT}/../../src"
${MW_INSTALL_DIR}
```

Verify that the paths were added correctly by expanding the Includes directory in the CCS project. The newly added include paths should be displayed as:



```
Includes
  C:/TI/ccsv6/tools/compiler/ti-cgt-c2000_6.4.6/include
  C:/TI/controlSUITE/device_support/f2802x/v230/f2802x_common/include
  C:/TI/controlSUITE/device_support/f2802x/v230/f2802x_headers/include
```

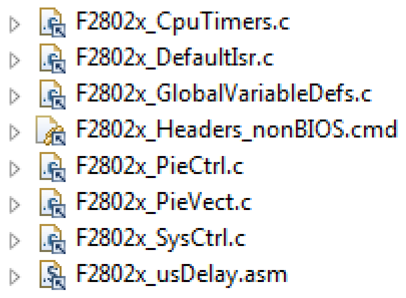
When the include path is not visible to CCS it will display the unresolved path greyed-out:



```
Includes
  /F2802x_headers/include
  C:/TI/ccsv6/tools/compiler/ti-cgt-c2000_6.4.6/include
  C:/TI/controlSUITE/device_support/f2802x/v230/f2802x_common/include
```

Step 2. Add the appropriate source files to your project

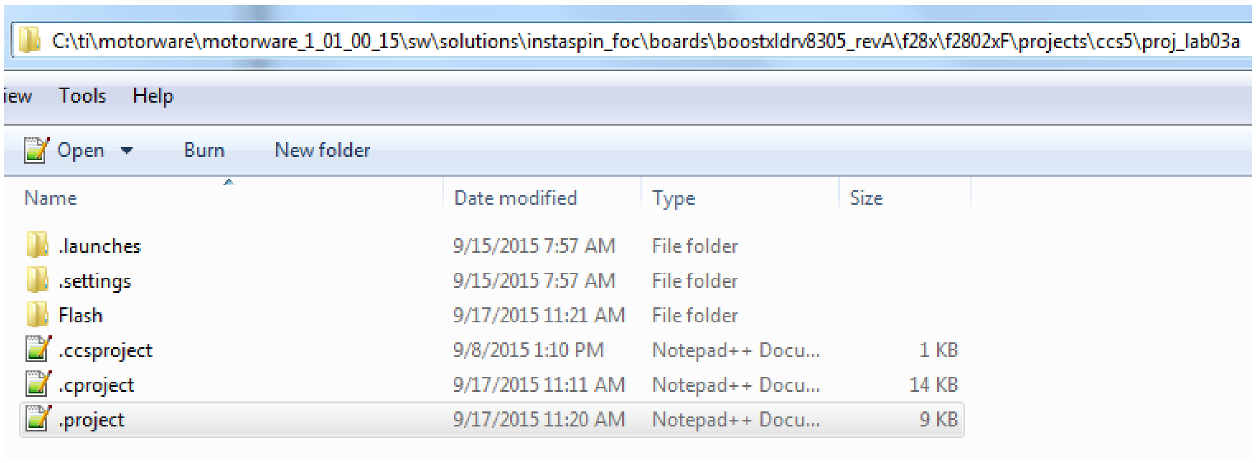
In this example, we will add the following files to project lab 3a. Wait, do not add these files from within CCS using the “Add Files to Project” method. Instead, let’s use an easier method as shown in the next steps.



```
F2802x_CpuTimers.c
F2802x_DefaultIsr.c
F2802x_GlobalVariableDefs.c
F2802x_Headers_nonBIOS.cmd
F2802x_PieCtrl.c
F2802x_PieVect.c
F2802x_SysCtrl.c
F2802x_usDelay.asm
```

In order to easily link in those files to project 3a, open the .project file of project lab 3a located here:

C:\ti\motorware\motorware_1_01_00_15\sw\solutions\instaspin_foc\boards\boostxldrv8305_revA\f28x\f2802xF\projects\ccs5\proj_lab03a



C:\ti\motorware\motorware_1_01_00_15\sw\solutions\instaspin_foc\boards\boostxldrv8305_revA\f28x\f2802xF\projects\ccs5\proj_lab03a

Name	Date modified	Type	Size
.launches	9/15/2015 7:57 AM	File folder	
.settings	9/15/2015 7:57 AM	File folder	
Flash	9/17/2015 11:21 AM	File folder	
.ccsproject	9/8/2015 1:10 PM	Notepad++ Docu...	1 KB
.cproject	9/17/2015 11:11 AM	Notepad++ Docu...	14 KB
.project	9/17/2015 11:20 AM	Notepad++ Docu...	9 KB

And then link the files as shown here:

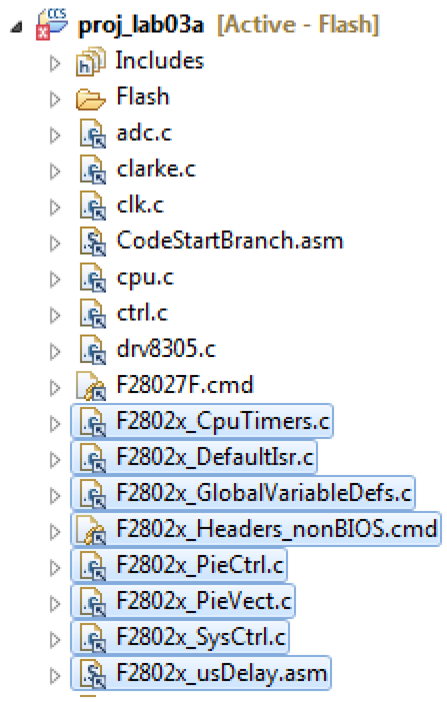
```
	<link>
		<name>F2802x_GlobalVariableDefs.c</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/f2802x_headers/source/F2802x_GlobalVariableDefs.c</locationURI>
	</link>
	<link>
		<name>F2802x_Headers_nonBIOS.cmd</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/f2802x_headers/cmd/F2802x_Headers_nonBIOS.cmd</locationURI>
	</link>
	<link>
		<name>F2802x_SysCtrl.c</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/F2802x_common/source/F2802x_SysCtrl.c</locationURI>
	</link>
	<link>
		<name>F2802x_usDelay.asm</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/F2802x_common/source/F2802x_usDelay.asm</locationURI>
	</link>
	<link>
		<name>F2802x_PieCtrl.c</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/F2802x_common/source/F2802x_PieCtrl.c</locationURI>
	</link>
	<link>
		<name>F2802x_PieVect.c</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/F2802x_common/source/F2802x_PieVect.c</locationURI>
	</link>
	<link>
		<name>F2802x_CpuTimers.c</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/F2802x_common/source/F2802x_CpuTimers.c</locationURI>
	</link>
	<link>
		<name>F2802x_DefaultIsr.c</name>
		<type>1</type>
		<locationURI>INSTALLROOT_F2802X_V230/F2802x_common/source/F2802x_DefaultIsr.c</locationURI>
	</link>
```

Make sure the .project reflects the added variable, if not, added to the bottom of .project

```
	<variable>
		<name>INSTALLROOT_F2802X_V230</name>
```

```
                    <value>file:/C:/ti/controlSUITE/device_support/f2802x/v230</value>
            </variable>
```

After linking all those files, you will see how they populate in your project window:



proj_lab03a [Active - Flash]
- Includes
- Flash
- adc.c
- clarke.c
- clk.c
- CodeStartBranch.asm
- cpu.c
- ctrl.c
- drv8305.c
- F28027F.cmd
- F2802x_CpuTimers.c
- F2802x_DefaultIsr.c
- F2802x_GlobalVariableDefs.c
- F2802x_Headers_nonBIOS.cmd
- F2802x_PieCtrl.c
- F2802x_PieVect.c
- F2802x_SysCtrl.c
- F2802x_usDelay.asm

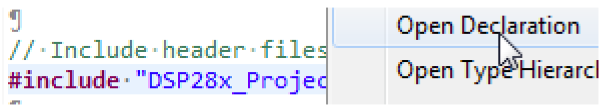
Step 3. Add F2802x Peripheral Header Files

This allows the use of bit fields and register names declared in controlSUITE in a MotorWare project. To do this, add this include file into the MotorWare project being modified (in this example, proj_lab03a.c)

```
#include "DSP28x_Project.h"
```

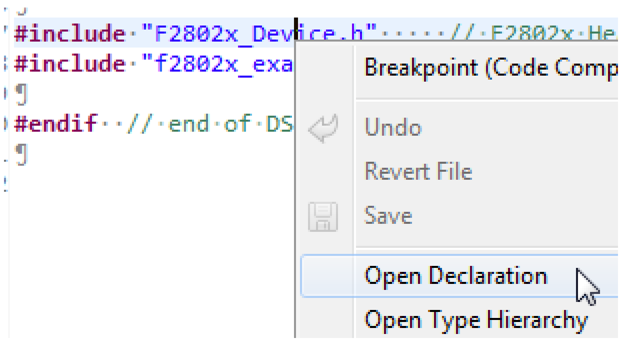
Step 4. Modify F2802x_Device.h to select the target you are building for

To do this, right click on DSP28x_Project.h and click “Open Declaration”



```
// Include header files
#include "DSP28x_Projec
```

Then right click on F2802x_Device.h and click “Open Declaration”


```
#include "F2802x_Device.h"     // F2802x He
#include "f2802x_exa
¶
#endif  // end of DS
¶
```

and type TARGET on the device you are building for, in this example, the following was selected as the target:

```
#define    DSP28_28027PT    TARGET
```

Step 5. Add code from controlSUITE to MotorWare project

1. As a quick example, we will just copy the three interrupts declared in Example_2802xCpuTimer.c of controlSUITE at the very end of proj_lab03a.c

```
__interrupt void cpu_timer0_isr(void)
{
   CpuTimer0.InterruptCount++;

   // Acknowledge this interrupt to receive more interrupts from group 1
   PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

__interrupt void cpu_timer1_isr(void)
{
   CpuTimer1.InterruptCount++;
   // The CPU acknowledges the interrupt.
   EDIS;
}

__interrupt void cpu_timer2_isr(void)
{  EALLOW;
   CpuTimer2.InterruptCount++;
   // The CPU acknowledges the interrupt.
   EDIS;
}
```

2. Add the interrupt prototypes at the beginning of proj_lab03a.c

```
__interrupt void cpu_timer0_isr(void);
__interrupt void cpu_timer1_isr(void);
__interrupt void cpu_timer2_isr(void);
```

3. Add the contents of main() function from Example_2802xCpuTimer.c into the beginning of main() of proj_lab03a.c

```
// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the f2802x_SysCtrl.c file.
   InitSysCtrl();

// Step 2. Initialize GPIO:
// This example function is found in the f2802x_Gpio.c file and
// illustrates how to set the GPIO to it's default state.
// InitGpio();  // Skipped for this example

// Step 3. Clear all interrupts and initialize PIE vector table:
// Disable CPU interrupts
   DINT;

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the f2802x_PieCtrl.c file.
   InitPieCtrl();

// Disable CPU interrupts and clear all CPU interrupt flags:
   IER = 0x0000;
   IFR = 0x0000;

// Initialize the PIE vector table with pointers to the shell Interrupt
// Service Routines (ISR).
// This will populate the entire table, even if the interrupt
// is not used in this example.  This is useful for debug purposes.
// The shell ISR routines are found in f2802x_DefaultIsr.c.
// This function is found in f2802x_PieVect.c.
   InitPieVectTable();

// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.
   EALLOW;  // This is needed to write to EALLOW protected registers
   PieVectTable.TINT0 = &cpu_timer0_isr;
   PieVectTable.TINT1 = &cpu_timer1_isr;
   PieVectTable.TINT2 = &cpu_timer2_isr;
   EDIS;    // This is needed to disable write to EALLOW protected registers

// Step 4. Initialize the Device Peripheral. This function can be
//         found in f2802x_CpuTimers.c
   InitCpuTimers();   // For this example, only initialize the Cpu Timers

#if (CPU_FRQ_60MHZ)
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 60MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 60, 1000000);
   ConfigCpuTimer(&CpuTimer1, 60, 1000000);
   ConfigCpuTimer(&CpuTimer2, 60, 1000000);
#endif
#if (CPU_FRQ_50MHZ)
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 50MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 50, 1000000);
   ConfigCpuTimer(&CpuTimer1, 50, 1000000);
   ConfigCpuTimer(&CpuTimer2, 50, 1000000);
#endif
#if (CPU_FRQ_40MHZ)
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 40MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 40, 1000000);
   ConfigCpuTimer(&CpuTimer1, 40, 1000000);
   ConfigCpuTimer(&CpuTimer2, 40, 1000000);
#endif
// To ensure precise timing, use write-only instructions to write to the entire
// register. Therefore, if any of the configuration bits are changed in
// ConfigCpuTimer and InitCpuTimers (in f2802x_CpuTimers.h), the
// below settings must also be updated.
```

```
   CpuTimer0Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0
   CpuTimer1Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0
   CpuTimer2Regs.TCR.all = 0x4001; // Use write-only instruction to set TSS bit = 0

// Step 5. User specific code, enable interrupts:

// Enable CPU int1 which is connected to CPU-Timer 0, CPU int13
// which is connected to CPU-Timer 1, and CPU int 14, which is connected
// to CPU-Timer 2:
   IER |= M_INT1;
   IER |= M_INT13;
   IER |= M_INT14;

// Enable TINT0 in the PIE: Group 1 interrupt 7
   PieCtrlRegs.PIEIER1.bit.INTx7 = 1;

// Enable global Interrupts and higher priority real-time debug events:
   EINT;   // Enable Global interrupt INTM
   ERTM;   // Enable Global realtime interrupt DBGM

// Step 6. IDLE loop. Just sit and loop forever (optional):
    for(;;);
```

Step 6. Modify CPU definitions to avoid compile errors

Some #defines are duplicated in MotorWare and controlSUITE, and we need to add a #ifndef to avoid re-definition. In particular, we need to modify F2802x_Device.h. Originally, this is what the code looks like:

```
#define  EINT   __asm(" clrc INTM")
#define  DINT   __asm(" setc INTM")
#define  ERTM   __asm(" clrc DBGM")
#define  DRTM   __asm(" setc DBGM")
#define  EALLOW __asm(" EALLOW")
#define  EDIS   __asm(" EDIS")
#define  ESTOP0 __asm(" ESTOP0")
```

But those #defines are also in MotorWare, so we simply need to add #ifndef in front of each one of those, so the modified code looks like this:

```
#ifndef EINT
#define  EINT   __asm(" clrc INTM")
#endif

#ifndef DINT
#define  DINT   __asm(" setc INTM")
#endif

#ifndef ERTM
#define  ERTM   __asm(" clrc DBGM")
#endif

#ifndef DRTM
#define  DRTM   __asm(" setc DBGM")
#endif

#ifndef EALLOW
#define  EALLOW __asm(" EALLOW")
```

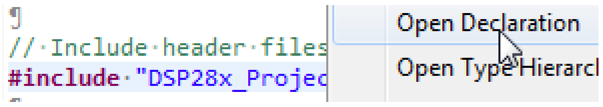
```
#endif

#ifndef EDIS
#define  EDIS   __asm(" EDIS")
#endif

#ifndef ESTOP0
#define  ESTOP0 __asm(" ESTOP0")
#endif
```

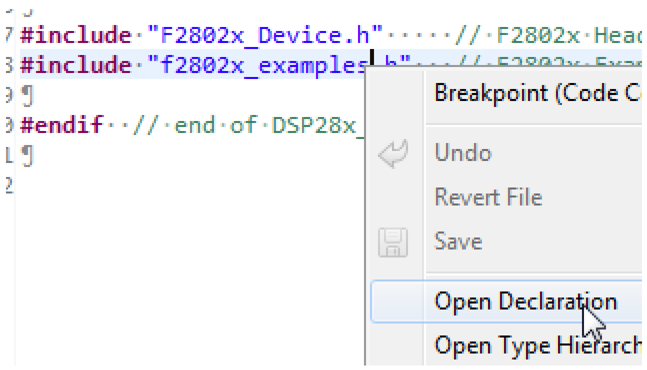
Step 7. Change declaration of conflicting data types

The combined source files have a different declaration of three variables declared by the linker. The conflicting variables can be seen as follows. First, right click and open declaration of DSP28x_Project.h



```
// Include header files
#include "DSP28x_Projec
```

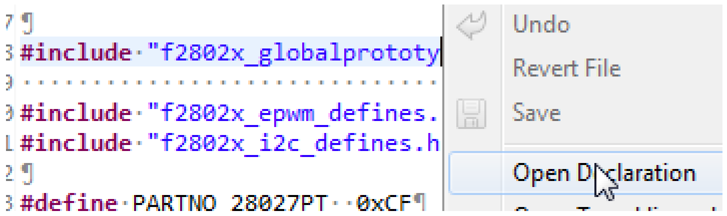
Then right click and open declaration of f2802x_examples.h:



```
#include "F2802x_Device.h"     // F2802x Heac
#include "f2802x_examples.h"

#endif  // end of DSP28x_
```

Then scroll down, look for f2802x_globalprototypes.h, right click and open declaration



```
#include "f2802x_globalprototy

#include "f2802x_epwm_defines.
#include "f2802x_i2c_defines.h

#define PARTNO_28027PT  0xCF
```

The following original code is in controlSUITE:

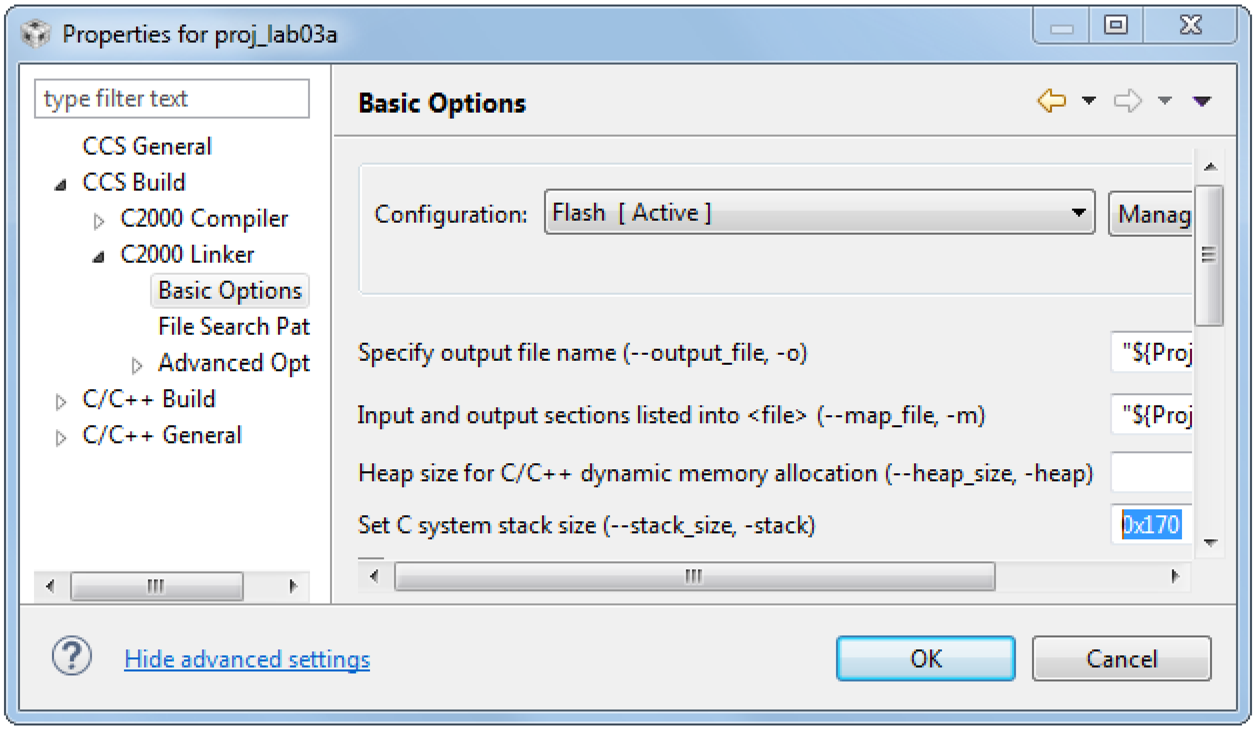
```
extern uint16_t RamfuncsLoadStart;
extern uint16_t RamfuncsLoadSize;
extern uint16_t RamfuncsRunStart;
```

And these variables need to be listed as shown below to avoid a conflict (add the highlighted text):

```
extern uint16_t *RamfuncsLoadStart;
extern uint16_t *RamfuncsLoadSize;
extern uint16_t *RamfuncsRunStart;
```

Step 8. Reduce the stack so program fits

Originally the MotorWare project (in this case proj_lab03a) reserves 0x180 words for the stack. You might need to reduce the reserved memory for the stack from 0x180 to 0x170



Configuration: Flash [Active]

Specify output file name (--output_file, -o) "${Proj

Input and output sections listed into <file> (--map_file, -m) "${Proj

Heap size for C/C++ dynamic memory allocation (--heap_size, -heap)

Set C system stack size (--stack_size, -stack) 0x170

Step 9. Modify CMD

CSM is defined as a MEMORY section in both F28027F.cmd and F2802x_Headers_nonBIOS.cmd, remove the definition from F28027F.cmd.

Step 10. Build and run

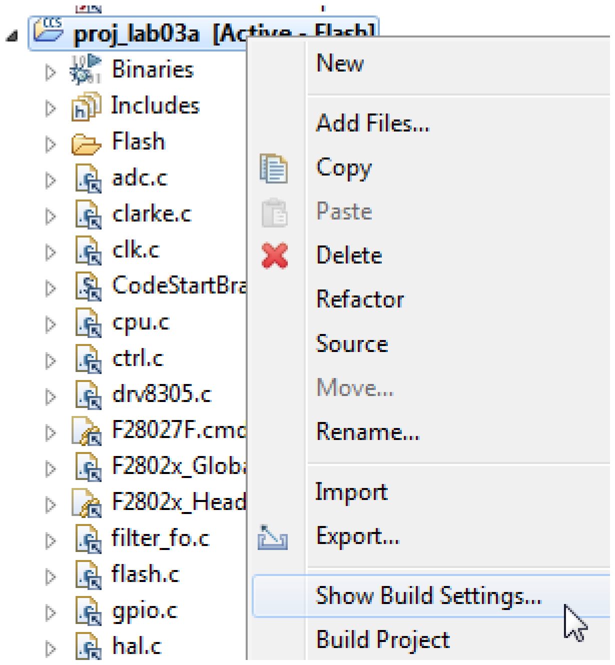
The last step is to build and run the code.

controlSUITE peripheral drivers

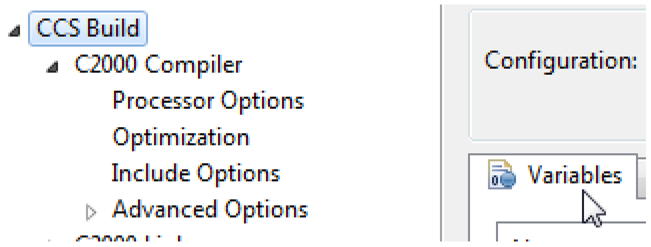
The objective is to allow a MotorWare project to use the controlSUITE peripheral drivers through the use of bit fields.

Step 1. Add controlSUITE include folders to the MotorWare project

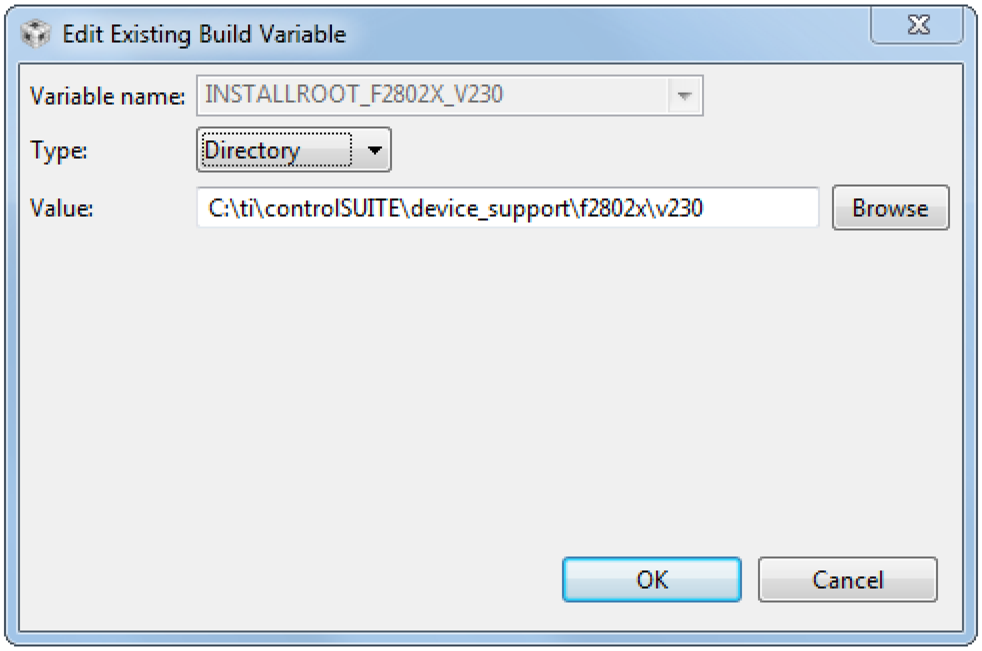
1. Right click and open properties of the project:



2. Go to variables



3. Add a new variable where the controlSUITE code for 2x is located:
   Variable name: INSTALLROOT_F2802X_V230
   Value: C:\ti\controlSUITE\device_support\f2802x\v230



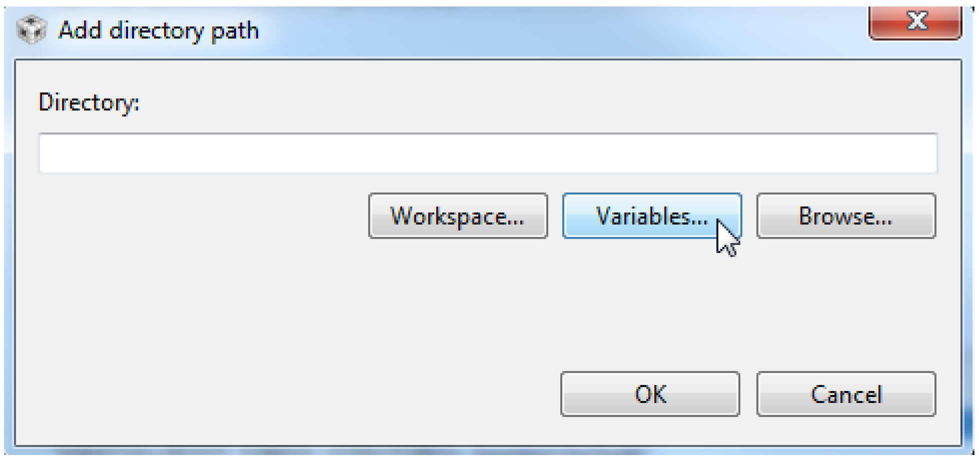
Variable name: INSTALLROOT_F2802X_V230

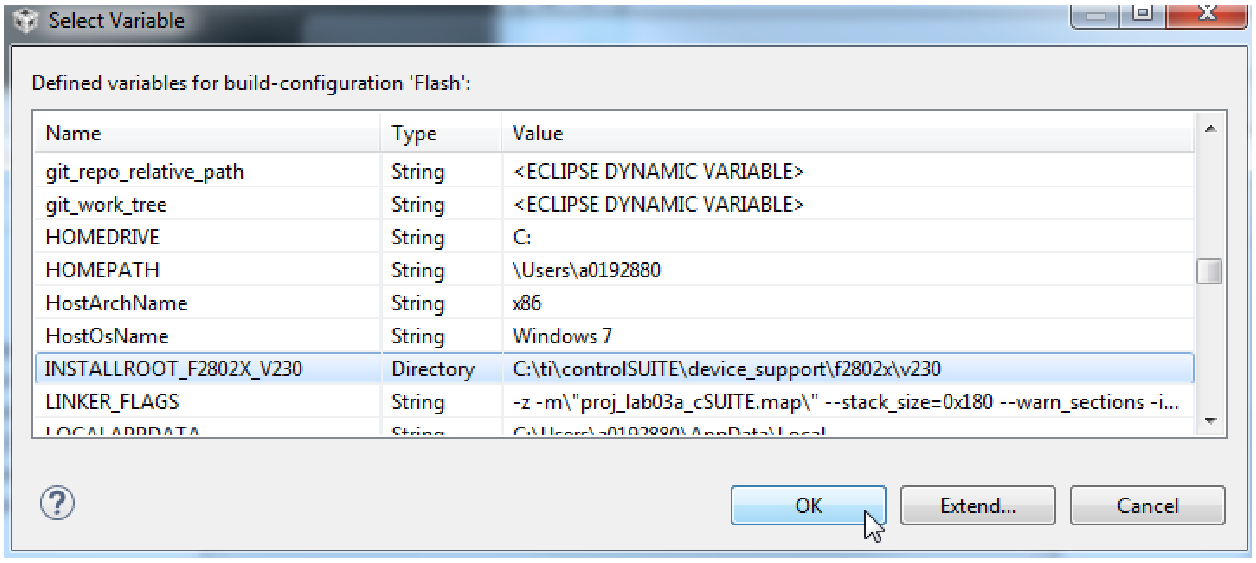
Type: Directory

Value: C:\ti\controlSUITE\device_support\f2802x\v230

4. Add this new include path:
"${INSTALLROOT_F2802X_V230}/f2802x_headers/include"

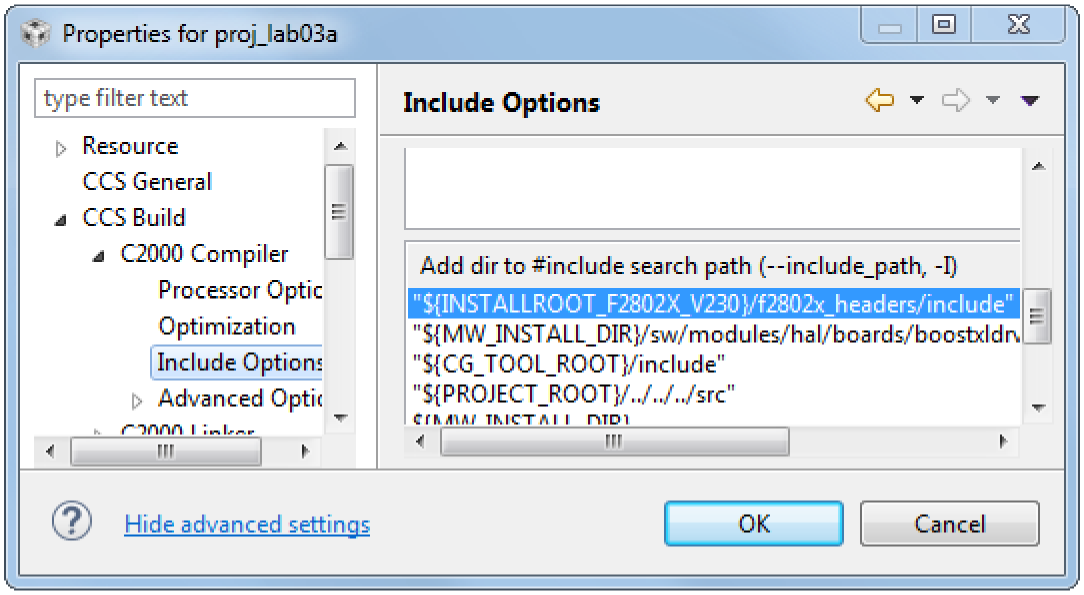
   Do not copy and paste the above text, CCS has an issue with this working consistently. Instead use the procedure below that uses the "Variables" button to select INSTALLROOT_F2802X _V230 from the displayed list, then complete the path by entering the required text as shown above.





Defined variables for build-configuration 'Flash':

Name	Type	Value
git_repo_relative_path	String	<ECLIPSE DYNAMIC VARIABLE>
git_work_tree	String	<ECLIPSE DYNAMIC VARIABLE>
HOMEDRIVE	String	C:
HOMEPATH	String	\Users\a0192880
HostArchName	String	x86
HostOsName	String	Windows 7
INSTALLROOT_F2802X_V230	Directory	C:\ti\controlSUITE\device_support\f2802x\v230
LINKER_FLAGS	String	-z -m\"proj_lab03a_cSUITE.map\" --stack_size=0x180 --warn_sections -i...



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

```
"${INSTALLROOT_F2802X_V230}/f2802x_headers/include"
"${MW_INSTALL_DIR}/sw/modules/hal/boards/boostxldrv
"${CG_TOOL_ROOT}/include"
"${PROJECT_ROOT}/../../../src"
```

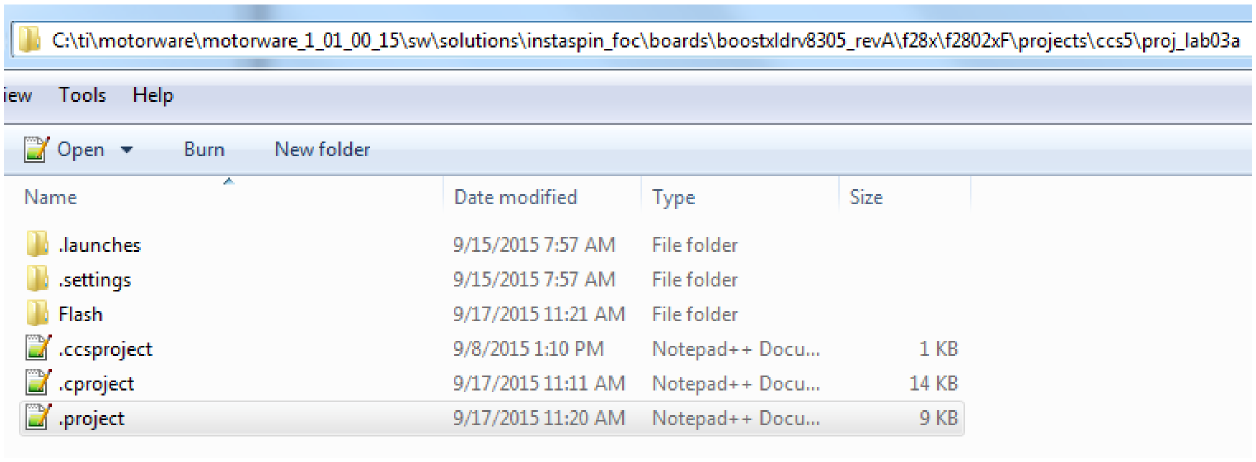
Step 2. Add the appropriate source files to your project

In this example, we will add the following files to project lab 3a. Wait, do not add these files from within CCS using the "Add Files to Project" method. Instead, let's use an easier method as shown in the next steps.

- F2802x_GlobalVariableDefs.c
- F2802x_Headers_nonBIOS.cmd

In order to easily link in those files to project 3a, open the .project file of project lab 3a located here:

C:\ti\motorware\motorware_1_01_00_15\sw\solutions\instaspin_foc\boards\boostxldrv8305_revA\f28x\f2802xF\projects\ccs5\proj_lab03a


C:\ti\motorware\motorware_1_01_00_15\sw\solutions\instaspin_foc\boards\boostxldrv8305_revA\f28x\f2802xF\projects\ccs5\proj_lab03a

Name	Date modified	Type	Size
.launches	9/15/2015 7:57 AM	File folder	
.settings	9/15/2015 7:57 AM	File folder	
Flash	9/17/2015 11:21 AM	File folder	
.ccsproject	9/8/2015 1:10 PM	Notepad++ Docu...	1 KB
.cproject	9/17/2015 11:11 AM	Notepad++ Docu...	14 KB
.project	9/17/2015 11:20 AM	Notepad++ Docu...	9 KB

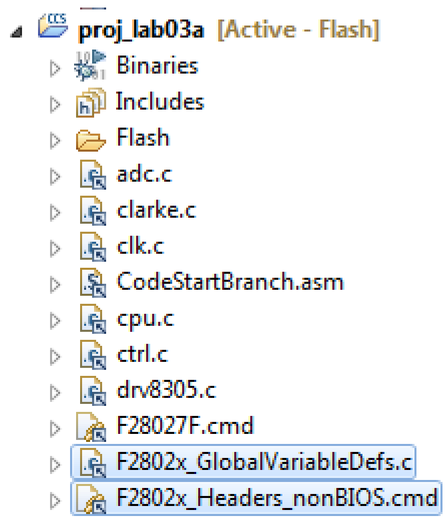
And then link the files as shown here:

```
		<link>
			<name>F2802x_GlobalVariableDefs.c</name>
			<type>1</type>
			<locationURI>INSTALLROOT_F2802X_V230/f2802x_headers/source/F2802x_GlobalVariableDefs.c</locationURI>
		</link>
		<link>
			<name>F2802x_Headers_nonBIOS.cmd</name>
			<type>1</type>
			<locationURI>INSTALLROOT_F2802X_V230/f2802x_headers/cmd/F2802x_Headers_nonBIOS.cmd</locationURI>
		</link>
```

Make sure the .project reflects the added variable, if not, add to the bottom of .project

```
		<variable>
			<name>INSTALLROOT_F2802X_V230</name>
			<value>file:/C:/ti/controlSUITE/device_support/f2802x/v230</value>
		</variable>
```

After linking all those files, you will see how they populate in your project window:



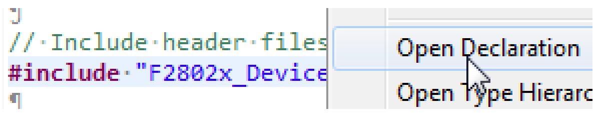
Step 3. Add F2802x Peripheral Header Files

This allows the use of bit fields and register names declared in controlSUITE in a MotorWare project. To do this, add this include file into the MotorWare project being modified (in this example, proj_lab03a.c)

```
#include "F2802x_Device.h"
```

Step 4. Modify F2802x_Device.h to select the target you are building for

To do this, right click on "F2802x_Device.h" and open declaration



```
// Include header files
#include "F2802x_Device
```

and type TARGET on the device you are building for, in this example, the following was selected as the target:

```
#define    DSP28_28027PT    TARGET
```

Step 5. Add code from controlSUITE to MotorWare project

As a quick example, we will write to some bit fields to show that the code compiles.

```
  CpuTimer0Regs.TCR.all = 0x4000;
  CpuTimer1Regs.TCR.all = 0x4000;
  CpuTimer2Regs.TCR.all = 0x4000;
```

Step 6. Modify CPU definitions to avoid compile errors

Some #defines are duplicated in MotorWare and controlSUITE, and we need to add a #ifndef to avoid re-definition. In particular, we need to modify F2802x_Device.h. Originally, this is what the code looks like:

```
#define  EINT   __asm(" clrc INTM")
#define  DINT   __asm(" setc INTM")
#define  ERTM   __asm(" clrc DBGM")
#define  DRTM   __asm(" setc DBGM")
#define  EALLOW __asm(" EALLOW")
#define  EDIS   __asm(" EDIS")
#define  ESTOP0 __asm(" ESTOP0")
```

But those #defines are also in MotorWare, so we simply need to add #ifndef in front of each one of those, so the modified code looks like this:

```
#ifndef EINT
#define  EINT   __asm(" clrc INTM")
#endif

#ifndef DINT
#define  DINT   __asm(" setc INTM")
#endif

#ifndef ERTM
#define  ERTM   __asm(" clrc DBGM")
#endif

#ifndef DRTM
#define  DRTM   __asm(" setc DBGM")
#endif

#ifndef EALLOW
#define  EALLOW __asm(" EALLOW")
#endif

#ifndef EDIS
#define  EDIS   __asm(" EDIS")
#endif

#ifndef ESTOP0
#define  ESTOP0 __asm(" ESTOP0")
#endif
```

Step 7. Build and run

The last step is to build and run the code.

Revision History

First release in MotorWare 1.01.00.16 March 2016