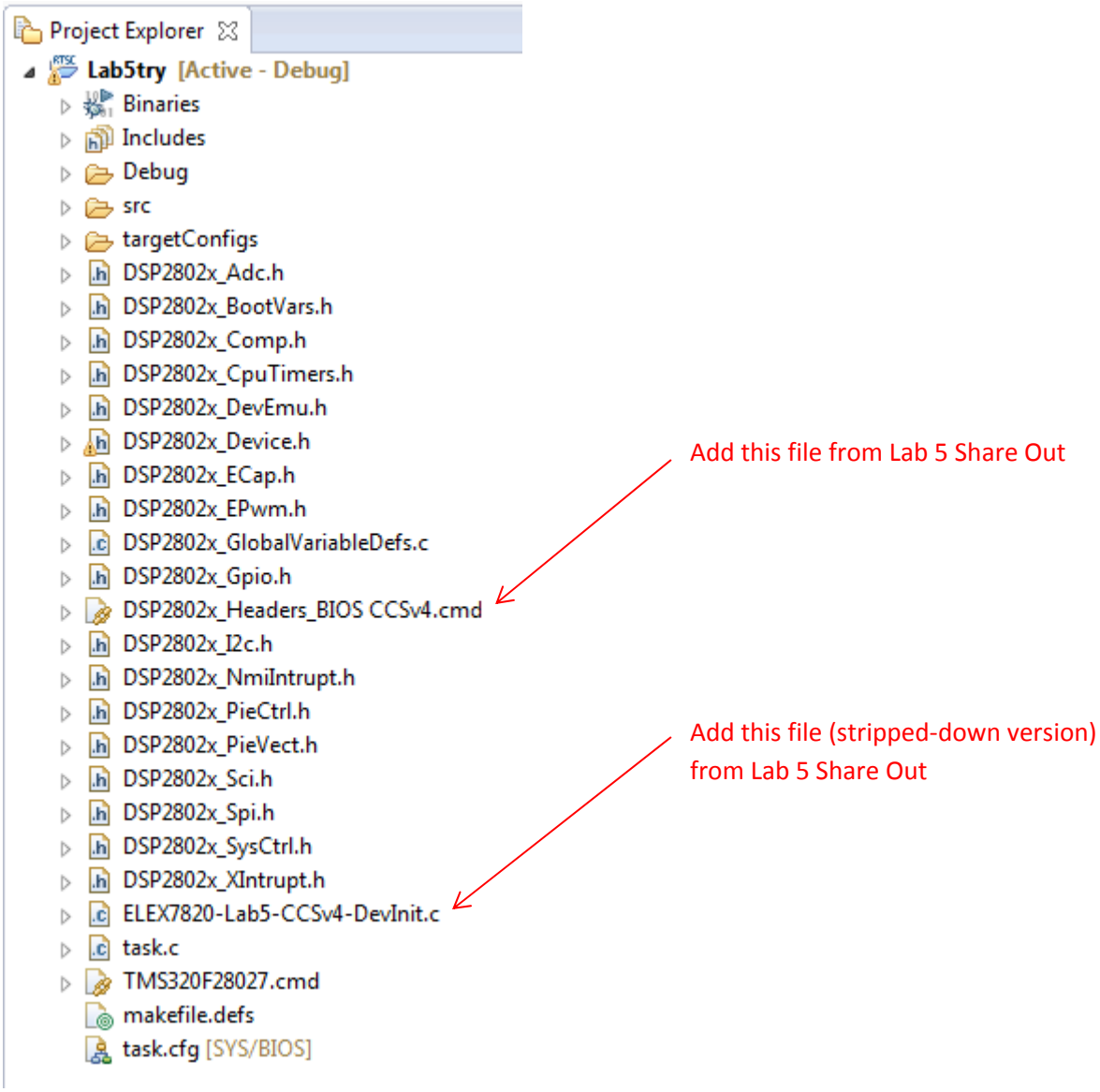
Info to get peripherals working (in CCSv4 and CCSv6, not CCSv5)

In this example code, I created a task.c project to start with and added various files from previous labs to support the use of the peripherals. To test it, I flashed an LED with a GPIO line.



Include file **DSP2802x_device.h** when you want to use a peripheral…

Also, add a function prototype for **`DeviceInit()`** and call it from main…

Also, on the lab computers, you need to add the "**`#define xdc__strict`**" line as shown – thanks to Maninder for finding this out (note that it contains a *double* underscore). Also, note that some students found they needed to "Clean" the project before this line took effect.

task.c

```
16 *
17 *  All output is routed to a log buffer which can be viewed
18 *  using the RTA "Raw Logs" viewer. After loading and running
19 *  the application, launch the Tools->RTA->Raw Logs tool to
20 *  view the logs.
21 */
22 #define xdc__strict
23 #include <xdc/std.h>
24 #include <xdc/runtime/Log.h>
25 #include <ti/sysbios/BIOS.h>
26 #include <ti/sysbios/knl/Task.h>
27 #include <ti/sysbios/knl/Semaphore.h>
28
29 #include "DSP2802x_Device.h"
30
31 void DeviceInit(void);
32
33 /* Semaphore handle defined in task.cfg */
34 extern const Semaphore_Handle mySem;
35
36 /* Counter incremented by timer interrupt */
37 volatile UInt tickCount = 0;
38
39 /*
40 *  ======== main ========
41 */
42 Int main()
43 {
44     DeviceInit();
```

Use the peripheral (in this example: GPIO line to toggle LED):

```
63 Void myTickFxn(UArg arg)
64 {
65     tickCount += 1;     /* increment the counter */
66
67     GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1; //toggle LED line
68
69     /* every 10 timer interrupts post the semaphore */
70     if ((tickCount % 10) == 0) {
71         Semaphore_post(mySem);
72     }
73 }
```