SE420 Laboratory Assignment 1
Welcome to the SE420 Lab (Where: Room 3075 ECEB)

Goals for this Lab Assignment:

1. Provide access to the lab room and its PCs
2. Go over the use of the PCs, where you will be saving your data, etc.
3. Rules for lab usage.
4. Quick introduction to the DSP controller we will be using and its development software called Code Composer Studio.
5. Introduction to TortoiseGIT and the SE420 GIT repository.

Matlab Functions Used: None

Prelab: No Prelab.

Grading Policy

- Each lab is worth 10 points. 10 labs x 10 points = 100 points.

Points	Grade
90	A
80	B
70	C

- Each lab should be completely signed off by your lab instructors before the beginning date of the next lab assignment.
- If you do not complete the lab by the beginning of the next lab, you will be given a one week extension to complete the lab without losing points. It is highly recommended that you keep up with the given due dates. This way you will not get far behind and have to rush though lab assignments to catch up.

Laboratory Exercise 1, Clone the SE420 Git Repository

- First follow the other Lab 1 document “Using the SE420 Repository” to check out the SE420 repository to your lab PC and/or your personal laptop. If you are going to be using both the lab PC and your laptop for development you will want to create the same path on both your laptop as on the lab PC. Create a folder using your NetID at the root C:\ drive and keep all your files there. It is also a good idea for you to make backups of your lab files outside of your Git repository as you progress through the semester. Making this extra backup will save you when we can’t figure out what went wrong with Git. Git is awesome and normally works great but every once in a while I have seen issues that caused students to lose versions of their files. This mainly due to us being beginner Git users.

Laboratory Exercise 2, Load your first Project into Code Composer Studio and Flash the default program to the DSP processor.

- Open Code Composer Studio 10 (CCS 10) and select the "workspace" folder in your repository. For example, if your netID was "superstdnt", you would have checked out your repository in c:\superstdnt\LabRepo. The workspace folder then to select would be c:\superstdnt\LabRepo\workspace.
- Once CCS 10 is done loading your workspace, you need to load the "labstarter" project. When you perform this load, the project is copied into your workspace. Therefore, if you rename this project, you will be able to load the "labstarter" project again if you would like to start another minimal project. I purposely located this "labstarter" project in the same folder that has many of the example projects you will be studying this semester. These examples are part of the software development stack that TI calls C2000ware. I have copied the needed parts of C2000ware into our repository so that if you accidently modify something you can easily get it back. Again now if your NetID was "superstdnt", perform the following to load your starter project. In CCS select the menu Project->Import CCS Projects. Click "Browse" and explore to "C:\superstdnt\LabRepo\C2000Ware_3_02_00_00_F28379D\device_support\f2837xd\examples\cpu1\labstarter" and finally press "Finish". Your project should then be loaded into the CCS environment. Let's then rename the project "lab1" by right clicking on the project name "labstarter". In the dialog box change the name to "lab1". Also explore into the project and find the file "labstarter_main.c". Right-Click on "labstarter_main.c" and select "Rename". Change the name to "lab1_main.c".
- Now that you have the project loaded you can build the code and download it to the LaunchPad board. Plug in your LaunchPad to the USB of your PC and then in CCS hit the green "Debug" button and select CPU1 only. This will compile the code and load it to your LaunchPad board. If you receive an error that states "Unable to launch CCS-debug-session" because an .ccxml file cannot be found, right click on your project name in the Project Explorer and select "Properties." Select "General" on the left hand side. Then in the "Project" tab find the "Connection" drop down and select "Texas Instruments XDS100v2 USB Debug Probe." "Apply and Close" and then you should be able to debug your code.
- Click the Suspend button to pause the code and the Resume button to resume code, and use both of these to prove to yourself that both the blue and red LEDs are blinking on and off.
- At home read through the main() function of your lab1_main.c file and see if you can find the function that changes the period of the timer functions. Change the period and see if the LEDs blink at a different rate.

Lab Check Off:

1. Demonstrate to your TA that you have successfully created, built and ran your first DSP project.
2. Create a Video at home to show us that you can run CCS 10 on your laptop, compile the default starter program, load and run the default starter program on your LaunchPad board. Attempt to change the period of the blink rate of the blue and red LED.