

ME 461 Laboratory #0

Soldering and Introduction to the Hardware and Software

Goals:

1. Learn to solder and practice by soldering the base set of components to the breakout board.
2. Start to become familiar with the TMS320F28379D LaunchPad and Code Composer Studio v10.
3. Build and run your first microcontroller program.

Exercise 1:

In this exercise, you are going to solder components to your breakout board. Your instructor will distribute the printed circuit boards, which will have some components already soldered to them. You will also receive a TMS320F28379D Launchpad. Your instructor will give a short soldering demo and show you where to find wire and other components in the lab. You will also be given the schematic files for the green breakout board you will be soldering. Follow the instructions of your instructor to solder your board

Exercise 2:

First follow one of the other Lab 0 documents “Using the ME461 Repository” to check out the ME461 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

Exercise 3:

- 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 accidentally 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\example\s\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.

Exercise 4:

- I have made updates to the “labstarter” project since last week. First to merge these changes from my repository that you forked, perform the steps in the “Using the ME461 Repository” document section “Course File Updates”. You will perform these merge steps at the github-dev.cs.illinois.edu website. Once the merge is complete, perform a pull request on your repository to get all the updates. This is a bit confusing the first time so please ask.
- Create a new labstarter project using the same steps as above. Rename this project something like “printtest”.

- Build and run this program. The LEDs should blink in the same fashion as before. Look at the code though and you will see that the function `serial_printf` is called every time the variable `UARTPrint` is equal to one. How often is `UARTPrint` getting set to one?
- To see this printed text you need to install a serial terminal on your PC. I like Tera Term in Windows. On Mac do a web search for Serial Terminal for Mac. Plug in your F28379D-Launchpad board to a USB port. We need to figure out what serial port COM number your USB serial port is using. The easiest way to find this is to run “Device Manager” in Windows and find the “Ports” item. Under ports find the COM number for the device titled “XDS100 Class USB Serial Port”. Run Tera Term and select the “Serial” item and find the XDS100 COM port in the list of COM ports. Final thing to do is change the Baud (or Bite) rate of the COM port. Still in Tera Term select the menu item “Setup” and then “Serial Port...”. Change the “Speed” to 115200 if it is not already.
- Now you are ready to “bug/debug” your code in Code Composer Studio. Download and run your code and you should see text be printed in the serial terminal.
- Also click (or give focus) to Tera Term and type some text into the terminal. The typed text will not be shown in the window but those characters are being sent to the F28379D. Notice that when you type the number of characters received increases. Look through this starter code and see if you can figure out where this number of characters received is incremented in the code. A fun thing to try is when the character ‘a’ is downloaded to the processor turn on an LED and when the character ‘b’ is received turn off that same LED. Of course to make this work nice you will need to stop blinking that LEDs in the timer functions.

Lab Checkoff:

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.
3. Either show in lab or create another home video showing that you can print text to a serial terminal on your laptop or home PC. Also show that when you type text into the serial terminal the number of characters received changes in the print line.