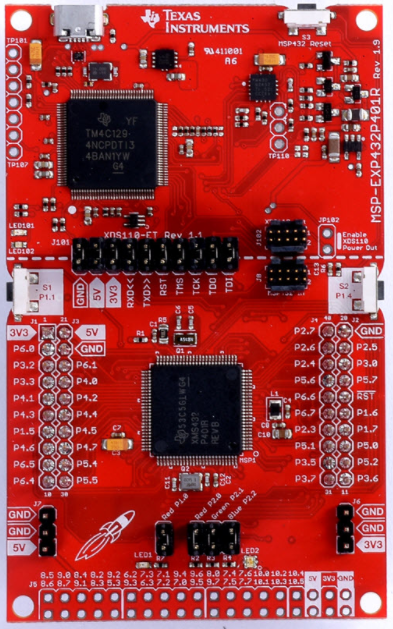
**Lab Exercise #3**

**Introduction to the MSP432 (ARM Cortex M4F) on the Texas Instruments MSP432 Launchpad Development Board**

**Objectives**

* To become familiar with the Code Composer Studio (CCS) Development System, an Eclipse based Integrated Development Environment (IDE), for creating, compiling, loading, and debugging programs to run on the MSP432 microcontroller (MCU)
* To install and run firmware on the MSP432 MCU that interfaces to external I/O devices (switches and LEDs) on the development board.
* To learn how to set breakpoints and watch memory locations on the MCU using debugging features of CCS
* To explore a Graphical User Interface (GUI) between a PC connected through USB to a MSP432 microcontroller (MCU) on a development board



**Pre-Lab Assignment** (refer to User’s Guide and use the info in the listed tables)

1. Download the MSP432P401R Launchpad User’s Guide from the MSP Documents folder on Blackboard. Study this document to become familiar with the capabilities of the board and its interfaces.
2. In your lab notebook, describe:

a. Three purposes of the TM4C129 MCU (the XDS110 debug probe) on this board

b. The default frequency of the master clock for the MSP432 and the source

c. What is connected to the MSP432 port pin P1.0? What is connected to P1.1? How can you use P1.0 to connect to an external component?

d. Why are there two voltage supply pins (5V and 3.3V) available on the Launchpad pins? What is the maximum power supply voltage for the MSP432? How is this power provided when the board is plugged into a USB port on the computer running CCS?

e. What is the maximum voltage that should be applied to MSP432 I/O pins?

The MSP432 Launchpad is a well-designed prototyping board, however, there are **a few precautions** you need to take to **avoid damaging it.** There is 5V available on the board for powering external circuits. **Do not connect 5V to any of the I/O pins on the board that are connected to the MSP432 MCU**.

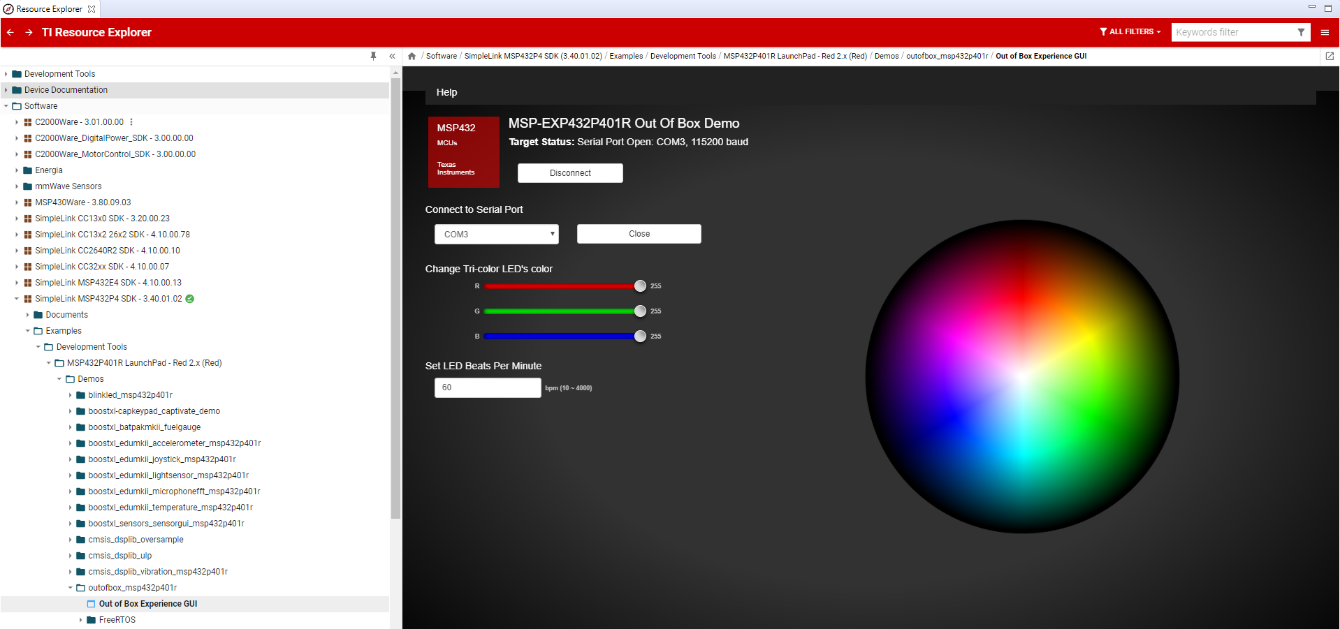
Also be sure to **limit the current output from or input to the MSP432 I/O pins** to less than 6 mA. This will be discussed in lecture, but, you should be aware that the board can be damaged if not properly wired to other devices.

**Part I – Explore the capabilities of the MSP432 Launchpad and GUI interface using the Out of Box demo**

The MSP432 Launchpad board comes pre-programmed with a demo that shows off some interesting features of an embedded system interface, namely two momentary switches that control the color and flash rate of a RGB LED. While the user can interface with the demo program directly through manipulating pushbutton switches on the board (see the description in the Quick Start Guide that came in the box with the board), a graphical user interface (GUI) introduces more possibilities.

A configured version of the GUI using JavaScript that runs on your PC and controls the Out Of Box demo on the development board can be launched from CCS using the Resource Explorer. A GUI Composer tool to create a graphical interface to the Launchpad is available as an on-line tool, but, is beyond the scope of this course.

1. Open Code Composer Studio.
2. Plug the MSP432 Launchpad board into a USB connector on the computer. The first time you plug your board into the computer, a driver for the Launchpad will be configured. Monitor the progress of this driver install to ensure it runs to completion.
3. The program for the Out Of Box Experience should be already loaded on the board and running on the Launchpad when first connected to the USB port. You can change the color of the blinking LED using one of the momentary switches on the board. The other momentary switch will change the blink rate. This can be done without running the control program that interfaces the board to the PC, but, the USB cable connected to the board needs to be plugged into the PC or USB charging port to power the board.
4. A Graphical User Interface (GUI) program provides remote control of the LED color and blink rate from the connected PC. You can find this program using the Resource Explorer in the View pulldown menu of CCS (if you do not see it, you can open it from the “Getting Started” menu). Navigate in the Resource Explorer window to Software / SimpleLink MSP432P4 SDK / Examples / Development Tools / MSP432P401R Launchpad / Demos / outofbox\_msp432p401r. The GUI is shown below

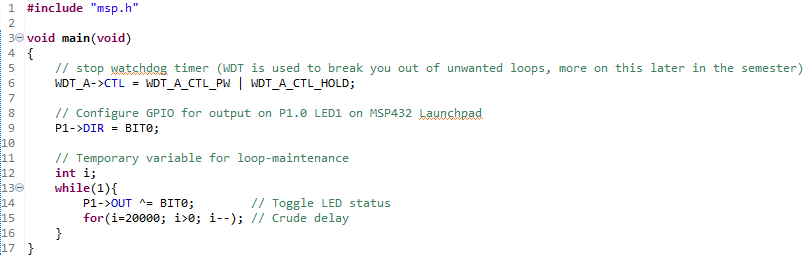


1. You can launch the GUI from a link in the **Out of Box Experience GUI** item that is immediately below the **Out of Box Experience** item (it may take a few minutes to popup a window on your screen). The user guide for the MSP432 Launchpad describes how this all works.
2. Spend a few minutes trying out the various controls on the GUI and see how they affect the color, intensity, flash rate of the LED on the development board. Move the cursor over the color wheel on the right side and click on various points to see the effect.
3. Click on the Out of Box Experience item in the Examples directory. Click on the “Import the example project into CCS” link. This will import the project into your own project directory.
4. Take a look at the source code in main.c (if it does not appear as a tab on the top window of CCS, navigate to it in the Project Explorer pane on the left side of CCS).
5. You do not need to bother with building this project right now, however, if you want to reload the demo program back onto your board after completing Part II of this exercise you should be able to restore the program from this project.

**Part II – Using the TI Code Composer Studio IDE to compile, download, and debug an application to run on the MCU**

The first step will be to create a project in CCS and include the source code. After compiling the source code, building the project, and downloading it to the MCU, we can explore features of CCS that will be useful in developing and debugging new applications for the MSP432.

1. Click on the File pull-down menu in the CCS toolbar, then New and specify a new CCS Project. A window will pop-up where you fill in a few of the details. Make sure the **Target** is set to **MSP432 Family and MSP432P401R.** Most of the defaults will be fine for now, just give the project a name and select an ***Empty Project (with main.c)***. After you click the Finish button, you can browse the project directory in the Project Explorer window on the left and see that several files have been auto-generated by the IDE. You can double click on these files to display them in the center window. These are standard include files for the TI compiler, a startup file that specifies the interrupt vector locations, a system file that initializes the processor, and a command file that gives directives to the loader for this particular target MCU.
2. The next step is to edit **main.c.** Let’s start with a simple program, one that configures an output pin on the MCU and interfaces to a LED. We’ll use the classic “blink the LED” program for this part of the exercise. Write the below code snippet into your main.c file.



1. This main() function manipulates the red LED on the Launchpad via I/O port 1 bit 0 (pin labeled as P1.0) by toggling this bit between logic states every 20000 cycles of a “for loop” (a crude way to create delay using the execution time of the instructions in the loop and the CPU clock). There are far better ways to do this using a hardware timer (and the MSP432 has several fine timers), but, this will do for now.
2. Right click on the CCS project in the Project Explorer window and select Build Project from the pull-down menu (or highlight the project name and click on the hammer icon). Monitor the progress of the build in the window that pops up. View the Console window. There should be no errors in the build process (if there were, the errors would be identified in this window) and the result should be a file with an extension .out.
3. The way to download to the MSP432 on the Launchpad board is to specify Debug from the Run pull-down menu on the toolbar (or click on the ladybug icon). The Eclipse view will change to CCS Debug and new windows will appear. The Debug method is the best for loading the hex file onto the MSP432 as you can then use the debug probe to test your program. Monitor the download process in the popup window. When completed, there will be a break point in your program at the first instruction in main(). It will show in the CCS source code window with the line highlighted and a breakpoint marker in the left margin (wait for this). To start the program running, specify the Resume from the Run pull-down menu (or click on the Resume icon).

**Note:** if you are not able to connect to the MSP432 Launchpad, you may need to reprogram the firmware on your board that handles the communication with the PC. This happens sometimes when TI updates the CCS program. See the procedure described at the end of the Appendix.

1. Download and open the **Lab3PartII\_Exercises.pdf** copy of PowerPoint slides that are excerpted from a TI workshop and follow along to further explore the debug process (The code will look slightly different than yours, including line numbers, but the objective is the same).

**Part III – Creating your own program to flash of the red LED**

In this part, you will create your own program (by modifying the program from Part II) that will control the rate of flashing of the red LED. Create a new CCS Project (name it what you like) as an Empty Project, this time without main.c. Import the file main.c from the project you created in Part II. Do this with the File->Import… and select General->File system and browse the location of the source for your previous project (probably C:\Users\*username*\workspace\_v9\*projectname*). Click on the box next to the file containing main().

1. Read the source code carefully and understand how the LED flash rate is controlled in this example code, then modify your code to set the delay by a variable amount. Basically, the program sets the MCU I/O port pin (***Pin P1.0***) connected to LED as an output and then toggles the pin value (on and off) after a fixed amount of delay. You will modify the program to set a variable to produce a different delay for the time between on and off.
2. ***Demonstrate the working program to the instructor. Show that you can change the delay variable using the debug interface to change the flash rate. Halt the program using the suspend icon and change the delay parameter in the Variables window (hit enter to update the variable). Then hit the resume icon to resume execution with the new delay value. You should not be terminating the debug session and rebuilding.***

Spend some time with the board and CCS to gain experience with the platform and the development tools.

One of the nice features of the MSP432 Launchpad development board is the on-board hardware debug feature (a capability that used to require rather expensive auxiliary equipment). You should familiarize yourself with this capability so that you can use it to debug software/hardware issues that may arise with more complex future embedded designs.

**Questions:**

1. Why do you think TI decided to use an Eclipse foundation for CCS? (see <http://www.eclipse.org/ide/> ) Describe the major features of the Eclipse IDE platform. What other devices are supported by an Eclipse based IDE?
2. How is a software breakpoint established in Code Composer Studio? How does this differ from a hardware breakpoint? (see <http://processors.wiki.ti.com/index.php/How_Do_Breakpoints_Work#Hardware_vs._Software_Breakpoints> )
3. What is a watchpoint? How does it differ from a standard breakpoint?

**At The End of The Laboratory**

* Clean up your workstation.
* Make sure you clearly understand the laboratory deliverables and due date as posted on Blackboard.
* Logoff and reboot the laboratory computer (if used).

**Laboratory Deliverables**

* Your lab report should include an explanation of the laboratory objectives and an explanation of how the program was completed, including any important code snippets or function references. In addition to a description of the program’s completion, you should include screen snippets of an example run through of the program, displaying the key functionality. One primary motivation for this lab is an exploration into the debugging interface as well, your report should discuss the debugging tools you learned.
* Submit a Copy of your lab report to Blackboard and attach the source code files as separate files to the BB link. Code that fails to compile will receive no credit.
* ***Follow the report writing guidelines provided on BlackBoard.***

**Appendix**

**Note: if you are not able to connect to the Launchpad, you may need to reprogram the auxiliary processor on the board using the procedure described in the ReadMe.txt file found in the directory**

**…\ccs\_base\common\uscif\xds110 where CCS was installed (probably C:\ti\ccsv9\).**

**Open a cmd window in this directory and run:**

**xdsdfu -m**

**xdsdfu -f firmware.bin –r**

Unplug the Launchpad and replug into the USB port. Now you should be able to click on the debug icon (or Run then Debug from the menu) and the connection should be made between CCS and the board. This step will download the program into the MSP432 MCU and then halt at the first executable instruction. Press the Resume icon (or Run then Resume) to execute the program.