Background

The purpose of this experiment is to familiarize yourself with the MSP430 Microcontroller, so that later this semester you can implement maximum power point tracking algorithms in a dc–dc converter, and so that you can control an inverter to produce a sinusoidal output. You will use the Code Composer Studio tools to write programs that control the digital and timer/PWM outputs, and in a later experiment you will use the A/D converter. Finally, you will implement circuitry that causes a power MOSFET to switch at a duty cycle programmable within the MSP430.

This experiment does not require a written lab report; instead upload the completed lab report sheet (attached at the end of this document) after getting signoff from your TA.

MSP430F5172 Development Board

Your group should purchase a parts kit from the ECEE electronics store. This kit includes a development board containing a Texas Instruments MSP430F5172 25-MHz microcontroller, with on-board 3.3 V regulator, LED, and JTAG port. All parts have already been soldered onto the board.

The development board can be powered from an external dc power supply (any value from 5 V to over 15 V will work) or from the JTAG connector; a jumper J5 (which you should add) selects this. The USB ports of the lab computers are capable of supplying the required power via the JTAG connector, but when operating your Exp. 3 system you will want to derive this power from an external bench or PV cart dc power supply. You will also need to add two other jumpers: (1) a jumper J7 that connects the LED to P1.0, and (2) a jumper J10 that selects the digital I/O voltage to either 3.3 V or an externally-supplied 5 V — install the jumper to use the board 3.3 V. A schematic of the development board is available on the course website in the Experiment 2 folder.

You can obtain a MSP-FET430UIF JTAG debug interface box from your TA. These boxes plug between a USB port of your lab computer and the JTAG header pins of your experimenter board. You can use this box to download code into your microcontroller, and also to debug your code. Please return this board to your TA at the end of your lab session.

There are two ways to connect the development board to a computer: (1) using an MSP-FET430UIF JTAG debug interface box, or (2) a TI Launchpad board. Your TA will instruct you which of these two options to take and will provide the appropriate cables:

1) MSP-FET430UIF JTAG debug interface box plugs between a USB port of your lab computer and the JTAG connector (JP1) of the Development Board.
2) The Launchpad also connects to a USB port of your lab computer, but requires four jumper wires to be connected between the Launchpad and the Development Board. Please see detailed instructions in the “TI Launchpad” on the course website in the Experiment 2 folder.

Laboratory Procedure

1. *Code Composer Studio and Blinking LED*

The laboratory Windows 7 computers include the development system:

Code Composer Studio version 6.1.3

Log into your lab computer, and start the Code Composer Studio program. If the program asks for a license, choose the free CODE SIZE LIMITED version. If the welcome page comes up, click on “Start using CCS”. Then choose the menu item:

File > New > CCS Project

Enter a project name. Create a directory inside the “temp” directory on your C: drive or on your Z: drive, and enter this location in the window (note that if you use the C: drive, this data will be erased when you log off of the computer). Select the “MSP430” Family, “MSP430x5xx Family” and “MSP430F5172” variant. Don’t change other menu settings, and click the “Finish” button. A default “main.c” edit window opens up as shown below, and you are now ready to begin coding:

```
#include <msp430.h>
/*
* main.c
*/
int main(void) {
     WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
     return 0;
}
```

Note that the first executable line within your main procedure is the code to disable the watchdog timer. This line should be included in the main file of all your projects. Enter the code discussed in Lecture 2 for blinking the LED, then save the file. Note that the LED is connected to pin 1.0, so it will be necessary to modify the code so that it outputs to the correct pin. You can now compile and link your code using the following menu command (or click on the build 🔨 button):

Project > Build All

If there are no errors, your code is ready to be uploaded into your development board. Start the debugger using the following menu command (or click on the debug 🐞 button):

Run > Debug

If an ultra-low-power advisor window pops up you can select “Do not show this message again” and Proceed. This should load the program into the MSP430. If Windows does not recognize the JTAG programmer or the Launchpad, it may be necessary to add this device manually. The driver for the JTAG debug interface board (MSP-FET430UIF) should be on the local C: drive inside the Texas Instrument program folder.

To start execution of the program on the Development Board, click on the green run button ▶, or use the pulldown menu:

Run > Resume

If you want to make changes, click the red stop button, type your changes, and save. Then repeat the process of uploading and running the program.

Demonstrate your blinking LED program to your TA, and ask the TA to record that you have completed this task by writing their initial in front of Task 1 on the attached Lab Report Sheet.

2. *Pulse Width Modulation*

Create a new workspace directory using the command

File > Switch Workspace …

Create a new workspace directory within the temp directory. After the program restarts, edit the CCS project settings and create a new .c file, as described previously.

Enter and modify the code (main.c) discussed in Lecture 2 to cause Timer D to output a PWM signal having a duty cycle of 36% and a switching frequency of 80 kHz. Operate Timer D with a clock frequency equal to 200 MHz, so that Timer D has a time resolution equal to 5 ns. Connect an oscilloscope to view the waveform, and debug as necessary.

Demonstrate your operational PWM signal to your TA, and obtain his/her initials in front of Task 2 on the attached Lab Report Sheet.

3. *Driving a Power MOSFET*

Thoroughly read the remainder of this procedure document before proceeding.

The MSP430 is unable to directly supply the voltage and current required to drive the gate of a power MOSFET. The parts kit contains a TC4428 gate driver IC capable of producing gate currents of 1.5 A, and of shifting voltage levels from 3.3 V up to 12-15 V.

Connect one of the low voltage power MOSFETs (HUF75321) in your parts kit on a breadboard. The MOSFET source should be connected to ground, and the drain should be connected through a power resistor to the 0–35 V bench power supply through a power resistor. You should choose the power resistor value and bench supply voltage so that the MOSFET drain current is approximately 0.5 A when the MOSFET conducts. Be careful to bypass the bench supply voltage on your circuit board, as discussed in Appendix A.

Mount the TC4428 on the breadboard, near the MOSFET. Connect the noninverting driver of the TC4428 between the MSP430 output pin and the MOSFET. Be sure to bypass the power supply of the gate driver IC with a ceramic capacitor, mounted as close as possible to the power supply and ground pins of the IC. The TC4428 should be supplied by the lab bench 12-15 V power supply. It is strongly suggested that you use twisted pairs to connect the MOSFET gate and source to the driver IC, and also a twisted pair to connect the MSP430 and its ground pin to the driver IC.

Connect the oscilloscope 10:1 voltage probes to measure the voltages at the gate and drain of the MOSFET. Measure the rise and fall times of the gate-to-source and drain-to-source voltage waveforms, as well as the turn-on and turn-off delay times of the MOSFET. Compare the duty cycles of the MSP430 code vs. the measured drain-to-source voltage.

Demonstrate your working circuit to your TA, and obtain his/her initials in front of Task 3 on the attached Lab Report Sheet.

Appendix A - Notes on construction and debugging of power electronic circuitry

Supply voltage decoupling

In all circuits, especially those that produce pulsating waveform, decoupling capacitors are essential to ensure that the dc supply voltage is undisturbed by large current transitions and spikes, and to prevent propagation of noise through the power distribution circuitry. Without decoupling capacitors, current spikes and transitions can cause large voltage transients and oscillations on the power supply wires, because of the inductance of the wiring. Such voltage “spikes” and “ringing” can disrupt operation of all devices connected to the power supply.

It is necessary to ensure that adequate capacitors are included in your circuitry for power supply decoupling. Here are some suggestions:

- Between the positive supply and ground terminals of the controller circuit board, connect an electrolytic capacitor of at least 10 μF (in the case of a power converter circuit board, a considerably larger capacitance may be required). This capacitor filters low-frequency variations in the power supply. Note that this electrolytic capacitor is polarized; connecting it to a power source with the wrong polarity will cause the capacitor to fail.
- For all integrated circuits on the controller board, a ceramic capacitor of at least 0.1 μF should be connected between each power supply pin and ground pin, as close as possible to the chip. This capacitor filters high-frequency variations in the power supply. Gate driver circuitry often requires additional high-frequency decoupling capacitance, perhaps 1 μF or more.

Constructing and debugging test circuits

Do not forget to power down the circuit before you make any additions or changes to the circuit. This is especially important in later experiments that drive the converter power stage. Turn on the power only after you have checked that your connections are correct. In particular, always check the polarity of the power supply connections to the chips.

If your circuit does not work, it is suggested that you troubleshoot it in a systematic manner. Use the oscilloscope to view the voltage waveforms at every pin of the first chip, beginning with its input. Are they as you expect? If not, then you have localized the problem. When the operation of the first chip is correct, then repeat the process for the next chip.

Course # (Circle One): 4517 or 5517______________________________

Names of Group Members: ______________________________________

____

____

Task	**TA Initials**
1. Demonstration of blinking LED	
2. Demonstration of PWM output having D = 36% and f_s = 80 kHz, with Timer D resolution better than 10 ns	
3. Demonstration of MOSFET driven by MSP430 PWM output	

After demonstrating all three parts of this experiment and getting your TA's signoff, upload this sheet in place of a written lab report in D2L.

This sheet is due by February 3, 2017, 11:59 pm (Mountain Time).