Experiment 2
PCB Layout and MCU Programming
ECE 482

The objectives of this experiment are:

- To understand layout parasitic and their mitigation in power electronics
- To design and fabricate a PCB for use in future experiments
- To become familiar with Altium software using a simplified circuit topology
- To become familiar with microcontroller programming and the MSP430 to be used later in this course
- To understand and implement the gate driver signals of the Boost converter to be constructed

This experiment is divided into two sections. Complete Section I before moving on to Section II. This experiment does not require a lab report.

Section I: Power Stage PCB Layout

Section I of this lab is completed *individually*. While you may collaborate with your team members, you should complete and submit our own PCB layout. In this section, you will design a printed circuit board (PCB) for a half bridge + gate driver, that will later be used to implement a Boost converter. In Experiment 3, you will design this boost converter based on your measurements and modeling from Experiment 1. For the moment, it is sufficient to know that the inductor current may be as high as 15A, and the dc bus voltage may be as high as 50V. The schematic and all component footprints are given in the supplied starter files.

PCBs will be fabricated using OSH Park 2-Layer Super Swift Service,

https://docs.oshpark.com/services/super-swift/

You will receive the required minimum 3 copies of your design. Your PCB should not exceed 4 in^2.

I. Half Bridge PCB Layout

A starter project in Altium is provided with this procedure. You should not need to edit the schematic, and may not alter part footprints or remove any components. The goal of this experiment is to complete the PCB layout of this schematic within the limitations on total PCB area.

Watch the recorded lectures posted to the course website along with this experiment. These videos cover topics related to power converter layout and practical implementation. After watching the lectures, complete the PCB layout of the half bridge power stage prioritizing (in order)

1. Power loop
2. Gate drive loops
3. All other connections

II. Deliverables

Follow instructions on the course website or OSH Park support page to generate a zip file of Gerber and NC-drill files to submit to OSH Park. The layout must successfully pass the OSH Park automatic file verification (run automatically when uploading the design files to the main page) before submitting to canvas.

You must submit, through canvas, one .zip archive of the completed PCB layout. The zip archive should contain the following file types from the Altium project:

1. PrjPcb
2. SchDoc
3. SchLib
4. PcbLib
5. PcbDoc

You do not need to submit gerber files in your assignment submission in Canvas as long as the files listed above correspond exactly to those used to generate the manufacturing files that you submitted to OSH Park. The instructor will review the Altium design and provide feedback with an opportunity to revise before purchasing the PCB.

In your submission, include your username and password (make sure to change your password to something not used for any other personal account) to the OSH Park webpage. Prior to submission, place only the design for this class into your cart under your account. EECS staff will login to your OSH Park account to complete the purchase of the PCB design.

Section II: MCU Programming

In this section, you will use Texas Instruments' Code Composer Studio to program your central controller in order to implement a pair of PWM signals using the MSP430 Launchpad development board. The MSP430 family User's Guide, Datasheet, and example code files are available from the experiment webpage.

III. Voltage-Controlled PWM

In order to drive the MOSFETs in a synchronous boost converter, two complementary, constant-frequency PWM signals are needed, as shown in Fig. 1

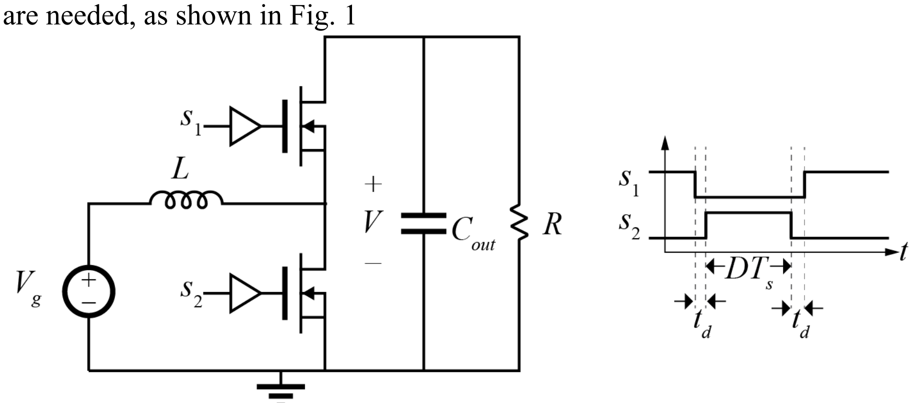

Figure 1: Boost converter gate drive signals s_1 and s_2, with deadtime t_d shown.

Using the example codes from Texas Instruments, or starting from scratch, write code to generate two complementary 20kHz PWM signals. Set the two deadtimes t_d to 500 ns. The duty cycle of the signal should be determined dynamically and set proportional to an analog input voltage. Limit the duty cycle so that at zero input voltage, the duty cycle is 1% and at 3.3V input, the duty cycle is 99%. You are free to have some error on all values mentioned here, as long as the basic functionality is achieved.

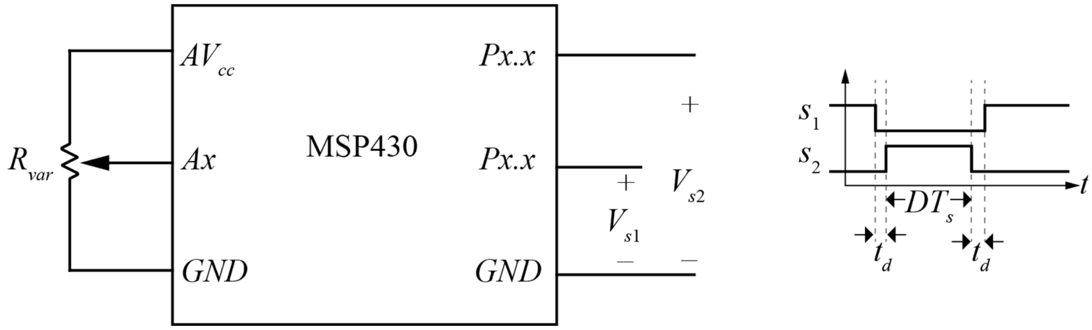

Figure 2: Connections to MSP430 board

Connect a variable resistor R_{var} as shown in Fig. 2 to one of the analog inputs of the MSP430. In your code, output the PWM signals to available GPIOs *Px.x* of your choice so that you can view them on the oscilloscope.

By taking experimental datapoints, complete the table below, including points spanning the full range.

Analog Voltage										
Duty Cycle										

Plot the resulting data in MATLAB. Once your plot is complete, demonstrate your working circuit, and submit the plot, by itself, to canvas as the deliverable for this Section of the experiment.