

# EE 458 – Power Electronics Controls

## Experiment 3 Pre-Lab Assignment: 3 Phase PWM and Setting up ADCs

For this prelab, you may work in your lab groups. Submit the ".c" file to the assignment. To grade, I will run your file and test functionality. You may want to submit waveforms or other proof of functionality should your code not work immediately.

### Task 1: Three Phase PWM

Lab 3 will need to use additional EPWM outputs to send 3-phase signals to an inverter (simulated). Prepare for the lab by setting these up beforehand. Use EPWM2,3, and 4, and they should be configured in the same way as EPWM1. A simple code for 3-phase sinusoidal duty to implement in your ADC Interrupt is below. This code won't work out of the box, you need to make sure to initialize important registers and variables, but this should be review from Lab 1.

```
// The following generates the frequency of the PWM
// The PWM frequency is the frequency at which the
// stator field rotates, and hence the rotor too
//In this code, w_s is the stator frequency
float w_s = 2*pi*10;
float f_div = 2*pi*fsamp/w_s;
if(wt>2*pi)
{
    wt=0;
}
wt = wt + (2*pi/f_div);
//f_div = 20000/10 This means we want 10 Hz sinusoid.

// EPwm 2,3,4 is set up for A,B and C phase respectively

EPwm2Regs.CMPA.bit.CMPA = (0.5+ 0.5*mod_index*sin(wt))*N;
EPwm3Regs.CMPA.bit.CMPA = (0.5+ 0.5*mod_index*sin(wt-al))*N;
EPwm4Regs.CMPA.bit.CMPA = (0.5+ 0.5*mod_index*sin(wt+al))*N;
```

## Task 2: ADC Set Up

We will need several ADCs for this lab. We already set up ADCINA3,5,6 which you can use to sense the motor stator currents. Also set up ADCINA9 to sense the angle from PLECS. It should be set up in the same way as Lab 1.

**Bonus (just for learning):** The end-goal of this lab is to integrate the boost converter code with the motor drive code. You may not get this far this quarter, although you will next quarter! To get a jump start, set up additional ADCs so that the boost converter control does not use the same ADCs as the motor control.