EEET2250 – Week 6 Lab Task (due Week 8)
Using PWM on the OUSB board to control a DC motor

Total possible marks: 15 marks (3% of final mark)

Aim:
The following Lab Tasks are aimed at helping you to build up code for the Lab Test 2 – Both the week 5 a week 6 Lab Tasks are directly related to your second Lab Test to be held during your Week 7 lab class (which will be just before the mid-semester break).

The code from your week 5 Lab Tasks is required for these Week 6 Lab Tasks, so please ensure that you have completed the Week 5 Lab Tasks first. The Week 6 Lab Tasks require you to write a program that implements a control loop (to change the PWM duty cycle) to ensure that the voltage across the (simulated) DC motor load reaches a target value. Components that simulate a DC motor load have been soldered onto the prototype are of the OUSB board.

The tasks specified below are aimed at helping you to break down the program into smaller functional 'modules': you should attempt to code in modules, checking the functionality of each module before continuing to code (i.e., do not write the whole program at once and expect it all to work!).

> ***NOTE*:** If you have your own OUSB board you are encouraged to solder a small motor and the required circuitry to your own board. You can use your own OUSB board for a special portion of the Lab Test 2 assessment which can be used to supplement your Lab Test 2 marks up to a value of 100% for the Lab Test 2 component – These are referred to as bonus marks in the Lab Test 2 proforma.

Once you have finished all the tasks notify your tutor who will assess your work. The marking for these tasks is binary – it either works or does not. Remember to use the lecture notes and OUSB board manual as references as well as the prescribed textbook. You also might find the OUSB Board Resources page on the EEET2250 Canvas shell a useful starting point too.

By completing the Week 5 and 6 Lab Tasks, you have effectively written most of your Lab Test 2 code, except for the required error checking, and error codes that are displayed on console output (which is used by autotester to assess your code). The Lab Test 2 proforma cpp file is available on EEET2250 Canvas Shell, and you may wish to code the week 5 and week 6 Lab tasks inside the provided Lab Test 2 proforma to gain familiarity with the Lab Test 2 requirements.

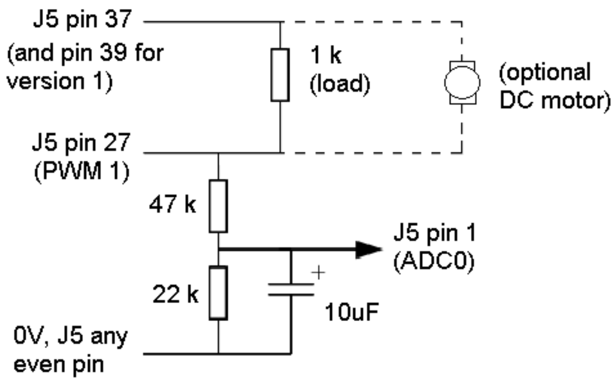
Background:
A (pictorial) circuit diagram of the components soldered onto the OUSB board is provided in the Lab Test 2 proforma cpp file. A 1k Ohm load resistor acts as the simulated motor load (connected across J5 pin 37 and J5 pin 27), and a low pass filter smooths the noisy motor voltage (explained below). J5 is the 40-pin (20 x 2) IDC connector directly above the prototype area on the OUSB board.

From the Week 5 Lab Tasks, we used 'PWM 1' on the OUSB board to generate a PWM square wave signal. 'PWM 1' is not only connected to LED3 on PORTB but also pin 27 of J5:

- When the PWM output goes high, LED3 on PORTB is turned on and pin 27 of J5 goes low thus pulling current through the 1k Ohm load resistor (i.e., the motor).

- When the PWM output goes low, LED3 on PORTB is turned off, and pin 27 of J5 acts as an open circuit thus no current flows through the load resistor.

Referring to the figure below and the (pictorial) circuit diagram in the Lab Test 2 proforma cpp file, note that there is a low pass filter to smooth out the noisy motor voltage:



The 47 k Ohm resistor and 22 k Ohm resistor form a voltage divider:
(Voltage on J5 pin 1) = (Voltage on J5 pin 27) * 22 / (22 + 47)

The resistors and capacitor form a low pass filter: the capacitor acts to smooth any voltage bumps from the square wave so J5 pin 1 sees a nice smooth voltage level. The motor voltage can then be read from J5 pin 1, which is connected to the port 0 of the analogue-to-digital converter (i.e., ADC0). This converts the analogue voltage to a number: 0 means 0 volts, 1023 means +5V DC.

In the Week 6 Lab Tasks, your code needs to implement a control loop which has the aim of getting the ADC read value to a desired target setting, by changing the PWM duty cycle. Your code can set an initial PWM value, say 50%. The code then reads the current ADC value, and if this does not match the target then the PWM duty cycle can be moved up or down 1% at a time. After several loops the ADC read value should be close to the target value.

Tasks

1. When you attempt each of the three tasks below, remember the six (6) steps of engineering design: draw flowcharts and write pseudocode to represent the logical flow of your program for each of the tasks below.

   [2.5 marks]

2. Using your code from the Week 5 Lab Tasks, modify your program to accept a target value for the ADC read value on ADC0 (i.e., motor voltage). Your program must accept a ‘L’ character (parameter) followed by the target value:
   e.g., `myProgram.exe L 372` // Sets a target ADC0 value of 372

   Write a control loop of 60 iterations to set the ADC read value to as close to the target value as possible (by changing the PWM duty cycle):
   - **IMPORTANT:** Before the loop starts, remember to set the PWM frequency to 46 Hz. Do not set this value every time the loop executes as it will lead to unstable PWM output, as the PWM system will reset every time you write the pwm-freq
     - **Make a note of what happens when you execute the pwm-freq command inside the for loop. Why is this a bad idea?**
   - On the first loop, set the PWM duty cycle to 50%
   - After setting the PWM duty cycle, read the current value of ADC0

- If the ADC value does not match the target value then the next PWM duty cycle may be set to +1 or -1 of its current value (but the PWM duty cycle must remain between 0 and 100 inclusive!)
- When your loop terminates, print out the last PWM duty cycle you sent in the PWM command

(Hint 1: Debugging the 'L' command can be difficult and it is strongly suggested that in development you print out a line of information at the end of each loop iteration (where ww, xx, yy and zz are numbers):

```
Loop# ww,  ADC target= xx,  ADC read yy,  Next PWM zz
```

(Hint 2: If your loop saturates the PWM at 0% or 100% try swapping the sign when you do a +- 1% change to the PWM. Also try changing the target value. See the Lab Test 2 proforma for the 'M' command for choosing a workable target for the ADC read value.)

[10 marks]

3. Perform the following error checks in your program (building on your Week 5 code):
   - If letters other than 'P', 'A' or 'L' are entered onto the command line, an error message is printed onto the screen.
   - PWM duty cycle entered must be between 0 and 100 inclusive (from Week 5)
   - ADC port number entered must be between 0 and 7 inclusive (from Week 5)
   - Target ADC read value must be between 0 and 1023

[2.5 marks]

4. *This task has no marks associated with it, however if you wish to access the bonus marks component during next week's lab test (Lab Test 2) then you should attempt to connect a real motor to your own OUSB board! Refer to the Lab Test 2 proforma explaining the requirements for command 'M', connecting a real motor to the OUSB board for bonus marks. The Bonus marks are for demonstrating your DC motor control and the 'M' command after you have submitted the formal solution to the autotester during your Lab Test 2 session in week 7. You can only do this if you have your own OUSB board! No motors should be connected to RMIT OUSB boards!!!*

[0 marks, 0 bonus marks, but should be ticked off if attempted as a lab task]

Once you have finished these tasks you are free to move on to working on your solution for the Lab Test 2 exercise.

You should not leave the laboratory after completing the tasks on this sheet - ensure that you are prepared for the Lab Test in Week 7.