- I will be having office hours on Saturday and Sunday this weekend, 10am – 11am
  - If there are people dong work, I will stay until 11:30am or so
- We are getting towards the end of microcontrollers
  - 2 – 3 projects left
  - Quiz
- I'm getting caught up with the grading
  - Majority of the grades made it in to blackboard this past weekend
  - I will post my solutions to the questions once I've got all the submissions
- Austin Thompson will have room open when?

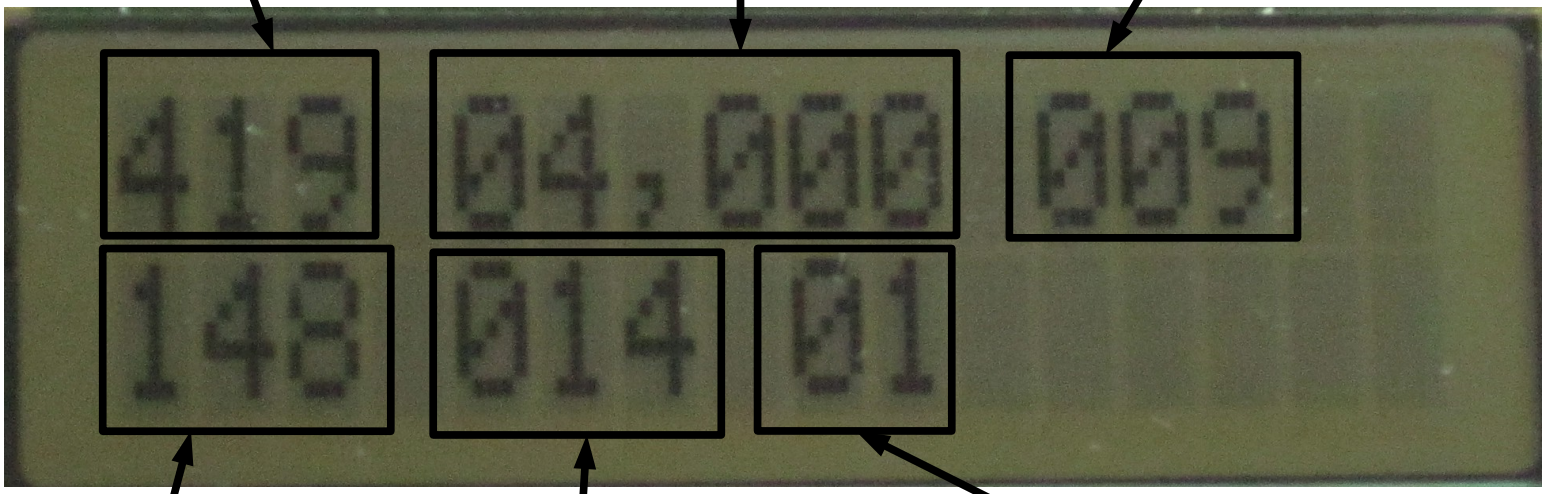
# Labs 6 and 7

- Lab 6 (6 – 7 boards)
  - Advanced PWM with piezo buzzer on DEM 2 board
  - One of the boards LCD is pretty much non functional
  - Another one is functional, but the bottom line of the LCD only shows about 70% of the line
  - We also have my board to use
- Lab 7 (6 – 7 boards)
  - Simple PWM with brushed DC motor on Mechatronics board
- We will be starting  both tonight
  - We need at least 5 groups starting lab 6 tonight

# Lab 6

- Using the DEM 2 board
  - Read
    - Frequency (Hz) and duty cycle (%) from pot
    - Display frequency, duty cycle, and PWM info on LCD
  - Compute
    - PWM settings and apply
    - Observe sound generated by piezo buzzer
    - Be considerate of others hearing
      - Pull jumper J9 or setting the output pin for PWM to input should disable the sound while you are developing
  - Mode
    - Use pushbutton(s) to toggle between
      - Reading frequency/reading duty cycle
  - This lab is by far is the most difficult of the two
  - Get credit via demo and answering questions

# Lab 6 Display




Oscillator frequency

PWM frequency

PWM duty cycle

PWM period

PWM duty cycle 8 MSBs

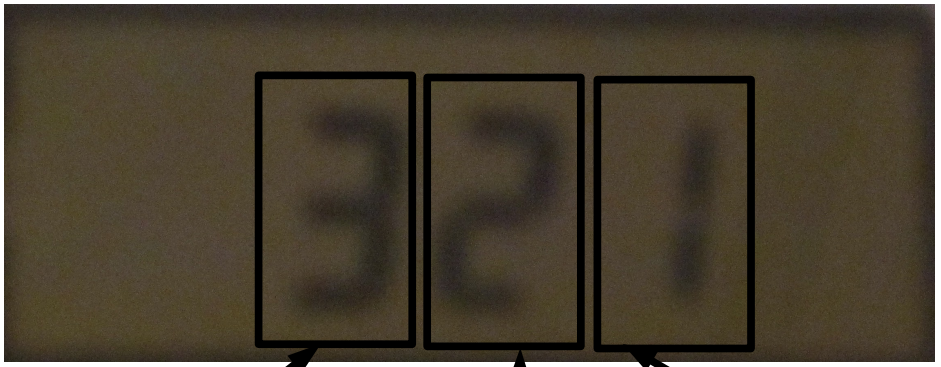
PWM duty cycle 2 LSBs

# Lab 7

- Using the Mechatronics board
  - Read
    - Duty cycle (%) from pot
  - Compute
    - Set PWM duty cycle value
    - Compute motor rotations per second (RPS) via optical interrupter
  - Display
    - Toggle every second between RPS & duty cycle on LCD
  - This lab is the easiest of the two labs
  - Get credit via demo and answering questions
- Although the motor is not very power full, keep your hair clear of motor axis

# Lab 7 Display

- Interface
  - void mech_clearLCD();
  - void mech_initLCD();
  - void mech_writeToLCD(unsigned char position, unsigned char value);



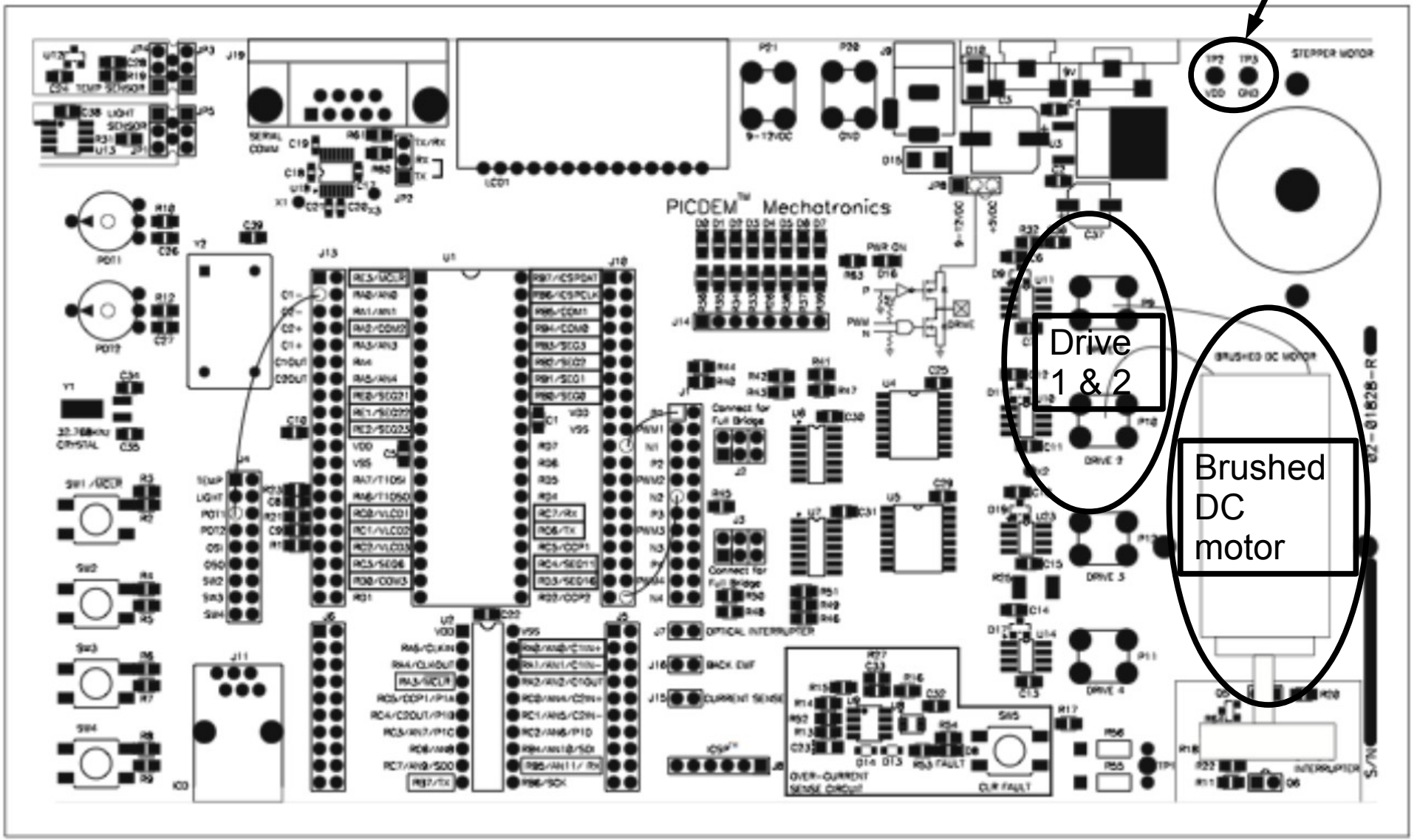

Position 3 Position 1

Position 2

# First Day on a Lab

- Get the input and output working
- Input
  - Potentiometer for both labs
  - One pushbutton for DEM 2 (use polling)
- Output
  - LCD for both

# Mechatronics Board




From PICDEM Mechatronics Demonstration Board Users Guide

# Mechatronics Board

- Some background on lab 6
  - Using the brushed DC motor
  - Drive 1 provides VDD
  - Drive 2 provides ground (GDN)
  - Provide constant VDD and oscillate GDN
- You can touch one of the motor's wires to VDD and the other to GDN and it will run a full speed
  - Likewise connecting an output port to each of drive 1 & drive 2 and outputting a 1 to each will cause the motor to run at full speed (constant power)
  - Rapidly changing port to drive 2 between 0 & 1 will change the speed of the motor, approximately proportional to percent of time with 1

# Mechatronics Board

- Given we can already control the speed of the motor just by changing how often we provide a 1/0 to an output port, why use PWM
  - With PWM you only compute and set the values when things change (requested speed, on/off)
  - Then setup the values for PWM
  - The microcontroller does all of the work via hardware
  - Leaving your code and processor available for other things
  - Each instruction cycle the microcontroller does this “for free” versus via code would take many instruction cycles

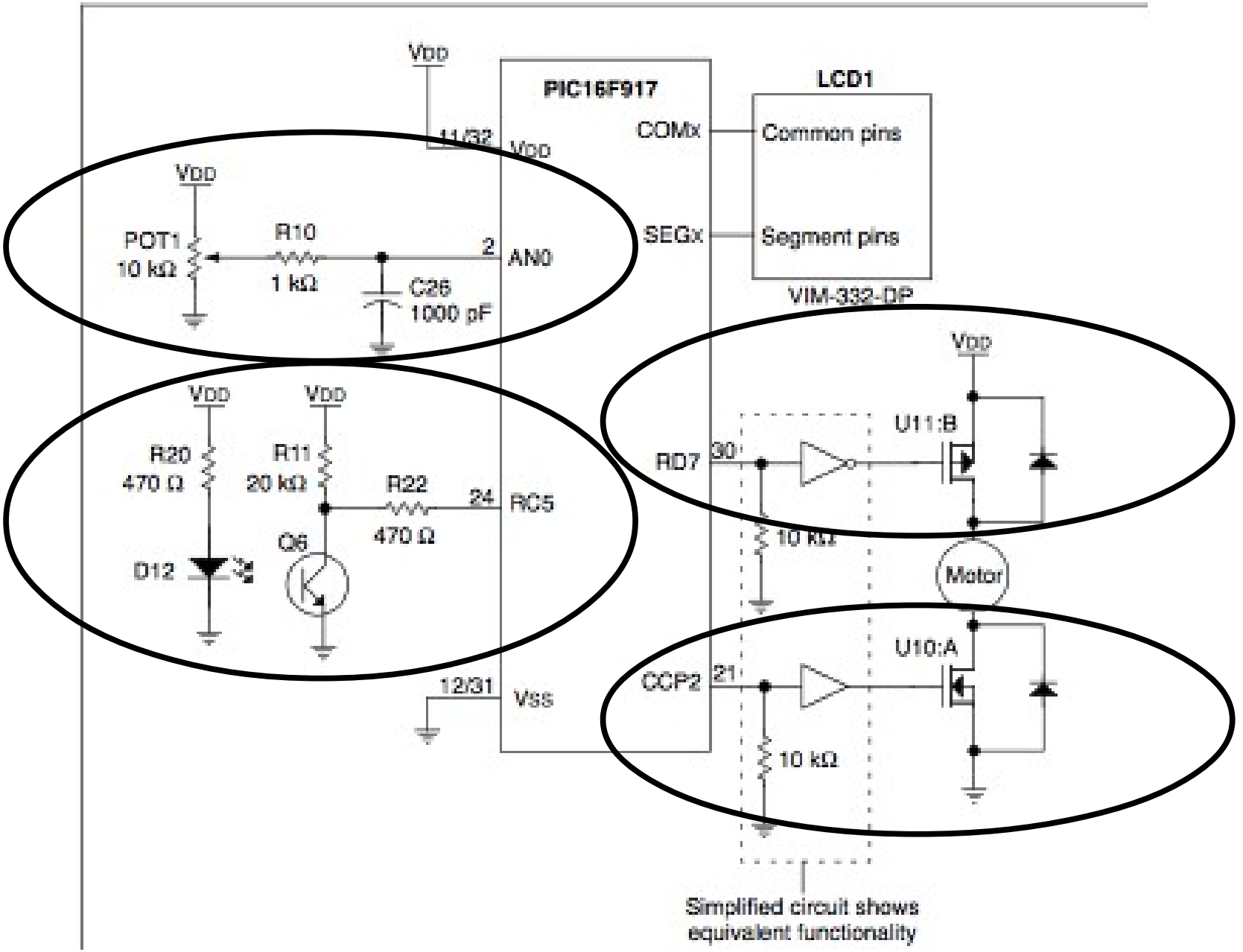
# Mechatronics Board

- We will use pulse width modulation (PWM) to oscillate power to drive 2 to control the speed of the motor
- For lab 7, duty cycle only
- For lab 6, duty cycle and frequency/period
- The mechatronics board also has a stepper motor that has precise angular control, whereas the brushed doesn't have this kind of control
  - We won't be programming the stepper moters

# Lab 7

- Connect
  - One wire from the motor to each of drive 1 & drive 2
  - Pot1 → AN0
    - Read duty cycle from potentiometer
  - RD7 → P1
    - Constant power to VDD
  - RD2 → N2
    - PWM output to GND (completes circuit)
  - Optical interrupter (J7) → RC5
    - Generate input to RC5 twice per motor shaft rotation
    - Use interrupt to count inputs on RC5
  - RD6 → RD0
    - Toggle LED and duty cycle/RPS to display every second

# Mechatronics Board




From PICDEM Mechatronics Demonstration Board Users Guide

# Lab 7

- AN0 is analog input from pot 1
  - Controls duty cycle
- Motor
  - RD7 is always set
    - Connected to P1 – positive for drive 1
  - RD2/CCP2 is output from PWM
    - Connected to N2 – negative for drive
  - Output from RD2 completes circuit and motor runs, with speed proportional to duty cycle
- Set internal oscillator to 8MHz
- A to D conversion clock Fosc/16
- A to D VCFG of 0b00

# Lab 7

- RC5 is output from optical interrupter
  - Optical interrupter generates a signal every half rotation of the motor
  - I've been able to have the motor going between 1 and 30 RPS
- Connecting the output of PWM (RD2/CCP2) to an LED will adjust the brightness
- Connecting RD5 to N2 and setting RD5 to 1 will run the motor at full speed (to test working)
  - I toggle RD5 each time my ISR is called, so that when I connect RD5 to N2 the motor doesn't run at full speed all of the time

# Pragmas and PWM for Lab 7

- #pragma config WDTE = OFF
- #pragma config FOSC = HS
- #pragma config BOREN = OFF
- #pragma config DEBUG = ON
- Use CCP2
  - Period of 20kHz (outside of hearing range)
    - With a lower value you may hear the motor
- You may need to push the CLR FAULT button on the Mechatronics board if the amber light is on