BeyerBot Motors

1 Collaboration Policy.

Collaboration is authorized.

Authorized Resources: You may use any electronic or hard copy resource at your disposal as long as 1) you cite your sources and 2) your use of the materials does not go against the intent of the assignment. For example, you can use a software library that you find online to help develop a project if you cite where you found it. However, you cannot complete your project by copying all of the source code, schematics, etc and simply running it on the required hardware.

2 Objectives.

1. Understand the fundamentals of Pulse Width Modulation (PWM) including period, frequency, duty cycle, and pulse width.
2. Understand how PWM can be used to control the speed of motors.
3. Demonstrate understanding by manipulating the motors at different speeds.

3 Pre-lab - To be done prior of class.

3.1 Reading:

Adapted from Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller, Jonathan W. Valvano, Third Edition, 2019.

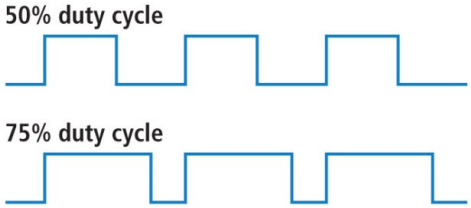
One effective way to deliver power to a DC motor in a variable manner is to use pulse width modulation (PWM). The basic idea of PWM is to create a **digital** output wave of fixed frequency, but allow the microctonroller to vary its duty cycle. The system is designed in such a way that **High+Low** is constant (meaning frequency/period is fixed). The **duty cycle** is defined as the fraction of time the signal is high:

$$duty\ cycle = \frac{High}{High + Low} = \frac{High}{Period}$$

Hence, duty cycle varies from 0 to 1 and will often be referred to as a percent. We interface this digital output wave to an external actuator (like a DC motor), such that power is applied to the motor when the signal is high, and no power is applied when the signal is low. We purposely select a frequency high enough so the DC motor does not start/stop with each individual pulse, but rather responds to the overall average value of the wave.

3.2 Key Terms:

- **Pulse Width Modulation (PWM)**: A technique to deliver a variable signal (power) using an on/off signal with a variable percentage of time the signal is on (duty cycle).
- **Period/Frequency**: Time required to complete one full cycle (on/off). Frequency is the inverse of period and is constant. The Arduino pins used by the robot operate at 490 Hz.
- **Duty Cycle**: Percentage of time a digital signal is **on** over a period of time



3.3 Examples:

1. Pulsing an LED on and off.

You have already applied PWM to pulse the red LED using the Arduino. You varied the duty cycle through the use of the *analogWrite()* function within the *for-loop* which varied the power applied to the LED and consequently, the brightness. The code you used is to the right.

```
// Global variable
static int LED_Pin = 11;

void setup() {
  // Set LED Pin as an output pin
  pinMode(LED_Pin, OUTPUT);
}

void loop() {
  // pulse the LED on (dim to bright)
  for(int i = 0; i< 255; i++){
    analogWrite(LED_Pin, i);
    delay(5);
  }

  // pulse the LED off (bright to dim)
  for(int i = 255; i >=0; i--){
    analogWrite(Led_Pin, i);
    delay(5)
  }
}
```

2. Answer the following given the example shown below in Figure 1.

   (a) What is the period of the signal?

   (b) What is the pulse width of the signal (how long is the signal **HIGH**)?

   (c) What is the duty cycle?

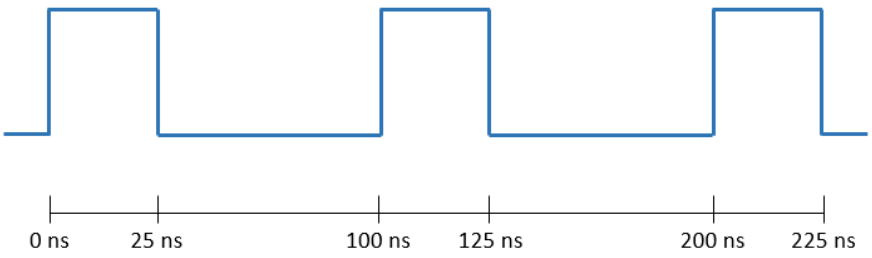


Figure 1

3. Given that the *analogWrite(pin, value)* function's *value* parameter is an integer that represents the duty cycle between 0 (always off) and 255 (always on), what would be the *value* passed to provide a 75% duty cycle to the motors?

3.4 Code:

1. Browse to Cadet Data and copy the Arduino sketch folder *robot_motors*:
   "K:\DF\DFEC\ECE210\Labs\Lab 8b - Robot Motors"
2. Open the *robot_motors* sketch.
3. In the *loop()* function, update the five calls to *robot.Motor_Forward()* to drive the motor at the designated duty cycles.
   - The example provided, *robot.Motor_Forward(50, 50);*, will drive both the left and right motors at (50/255)% of the duty cycle. This is equivalent to a duty cycle of about 20%.
4. Upload the sketch to the robot. Both motors of the robot should now operate at 6 different speeds (from 20% duty cycle up to 100% and back down to 15%).

4 Lab.

This lab will demonstrate how varying the PWM signal changes the average power delivered to the DC motors and, effectively, the speed of the motors.

4.1 Materials:

1. Assembled BeyerBot
2. Arduino programming cable
3. Laptop
4. Male-to-female/male Y-connector and male-to-male wire (provided for you in the Moku:Lab)

4.2 Connect the Moku:Lab probes:

1. Disconnect **ONLY** the wire from *Pin-3* on the Arduino.
2. Insert male end of the Y-connector into *Pin-3*.
3. Connect the male end of the wire coming from your Printed Circuit Board (the one disconnected from *Pin-3*) into the female end of the Y-connector.
4. Connect the positive (red) end of the Moku:Lab probe to *Pin-3* on the Arduino using the Y-connector.
5. Insert a male-to-male wire into the *GND-Pin* on the Arduino and connect the ground (black) end of the Moku:Lab probe to the other end of the wire.
6. Your setup should now look like Figure 2. All other wires should still be connected as they were previously.

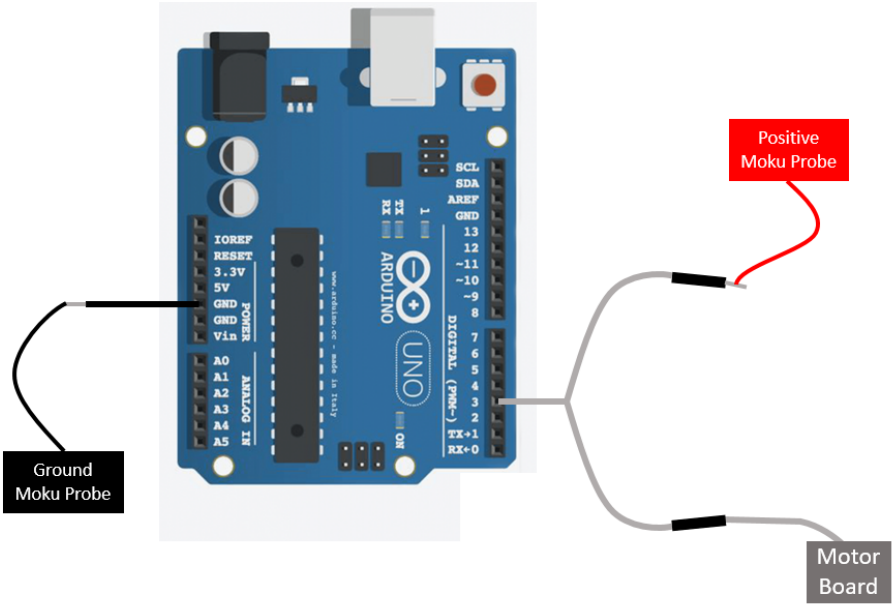


Figure 2

4.3 Measuring Pulse Width Modulation Signals:

1. Place your robot on the Digital Multi Meter (DMM) (to allow the program to run without your robot driving away)
2. Your Moku:Lab display should look like Figure 3. If it does not, raise your hand.

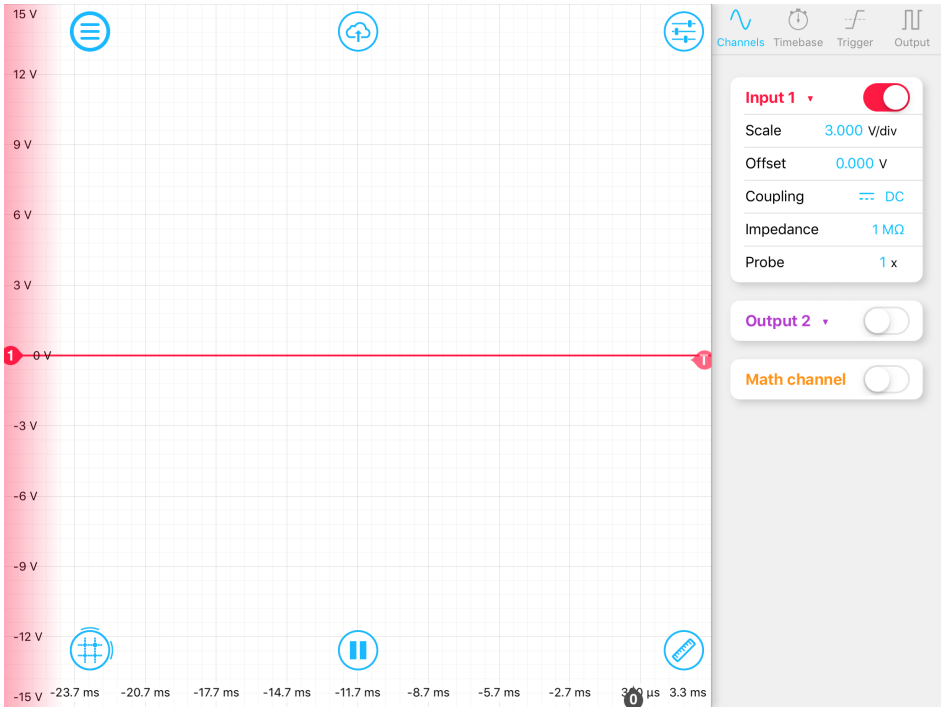
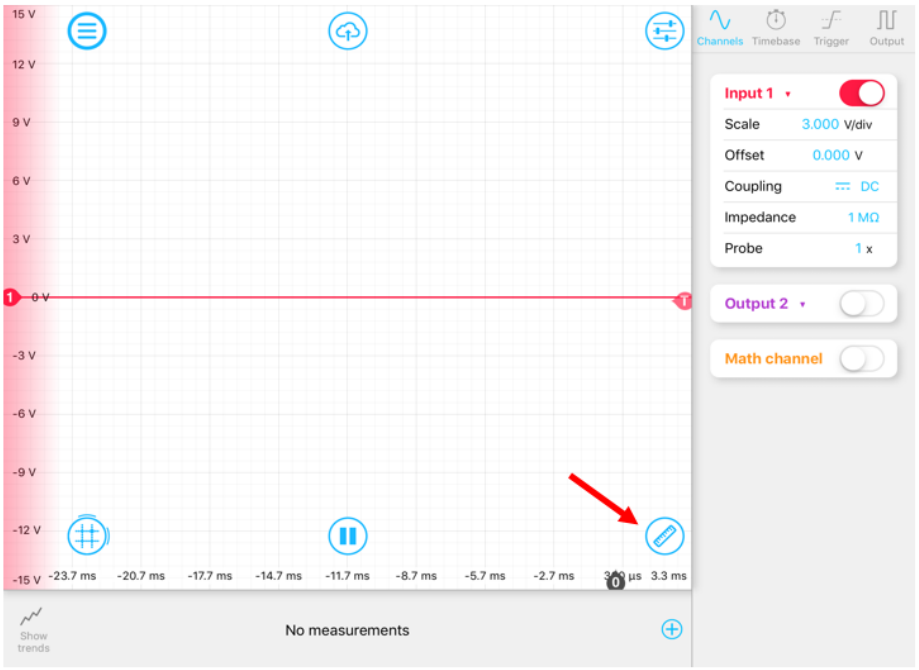
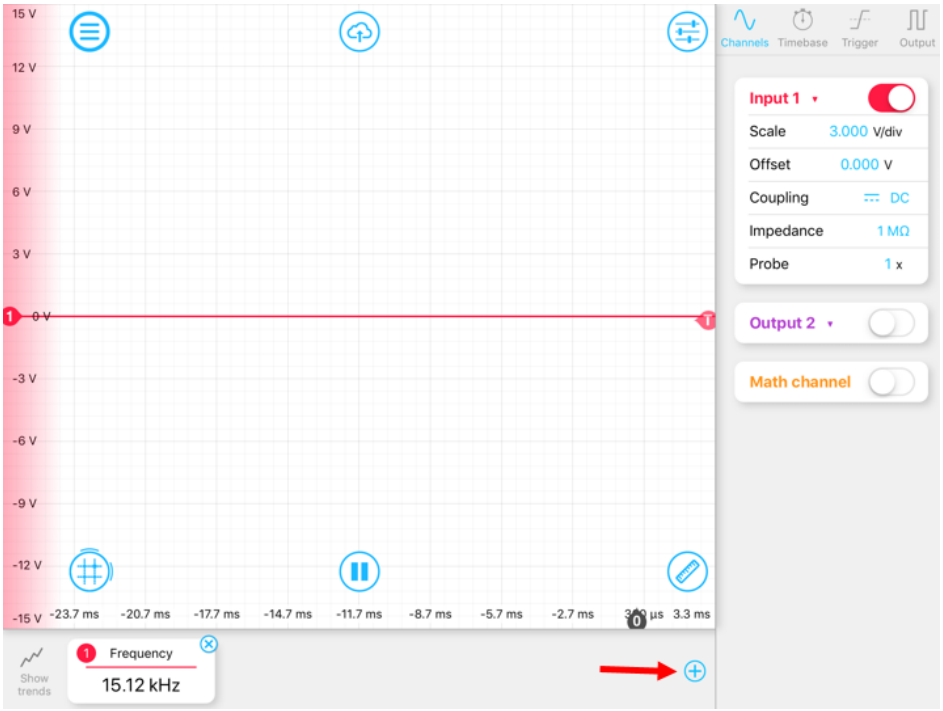


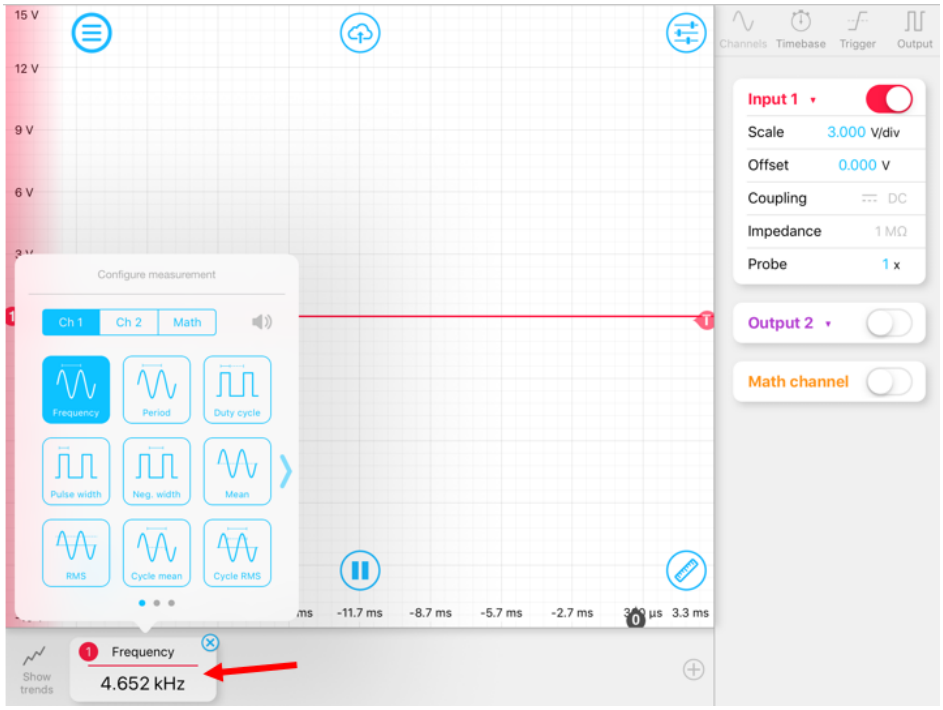
Figure 3

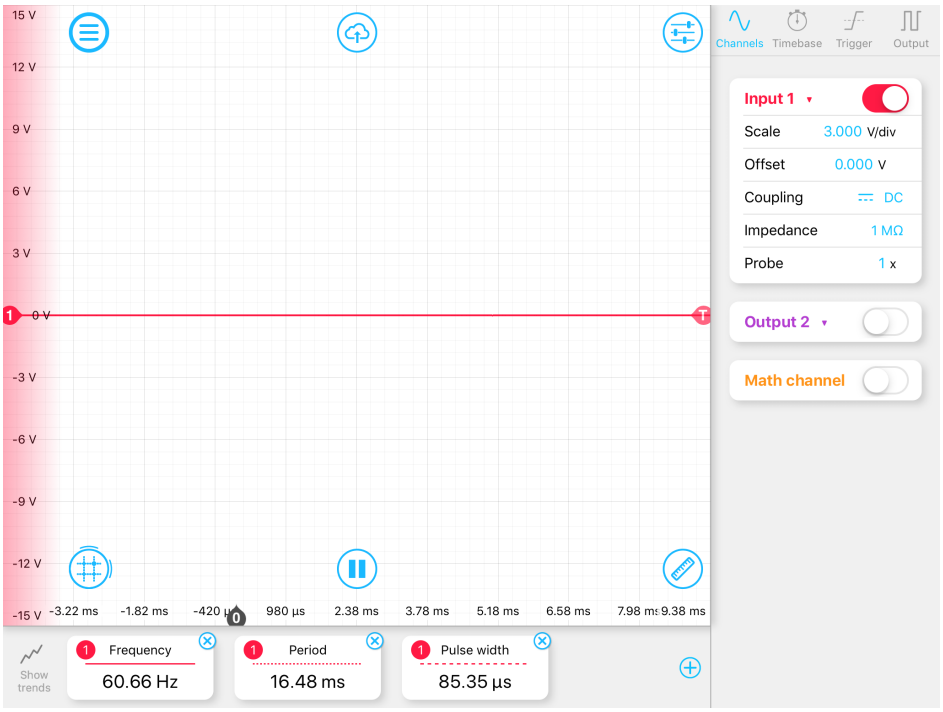
3. Open the measurements bar by pressing the ruler in the bottom right of your screen.



4. Add three measurement tools: Frequency, Period, and Pulse width. Press the "plus" sign within the measurements toolbox to add a new tool. Select the tool to change it. Repeat this for each of the four tools.







5. Turn on your robot and run the previously uploaded code.
6. What do you notice?

7. Comment out all of the code within ***loop()*** except for the portion that runs the motors at 75% duty cycle (*robot.Motor_Forward(191,191)*; *delay(2000);*) and answer the following questions:

   (a) What is the frequency of the signal?

   (b) What is the period of the signal?

   (c) What is the pulse width of the signal (how long is the signal **HIGH**)?

   (d) Using these values, calculate the duty cycle.

   (e) Is this duty cycle what you expected?

   (f) Add a Duty cycle measurement tool on the display (reference step 4.3-4). Does the value match what you calculated?

   (g) How does the speed of the motors correlate to the duty cycle?

5 Feedback.

1. Was there any part of the lab that was confusing or unclear?

2. Is there any aspect of PWM that you still find confusing or would like more clarification on?

3. What would you change from this lab for next semester?