CMPE 460 Lab 6
Motor Control

David Feng, Alice Fischer, and Joel Yuhas

Performed: March 2nd , 2018
Submitted: March 21st , 2018

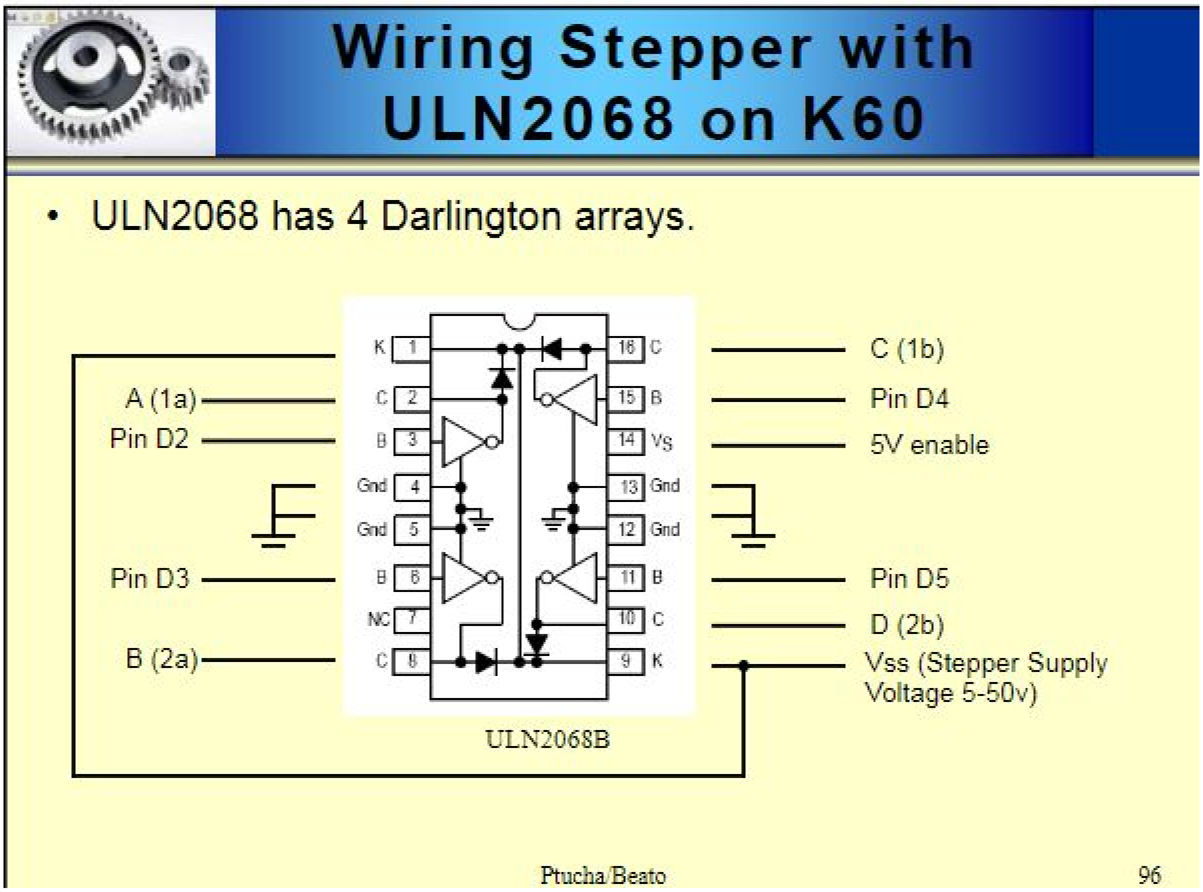
Lab Section: 01-L2
Instructor: Professor Louis Beato
TA: Bao Thai
Adrian Tygart

Lecture Section: 01
Professor: Professor Ray Ptucha

Lab Description (two-line)

The purpose of this exercise was to learn how to amplify low power signals from the microcontroller to drive higher power devices. A stepper motor and a motor using H-bridge circuits were operated using the microcontroller.

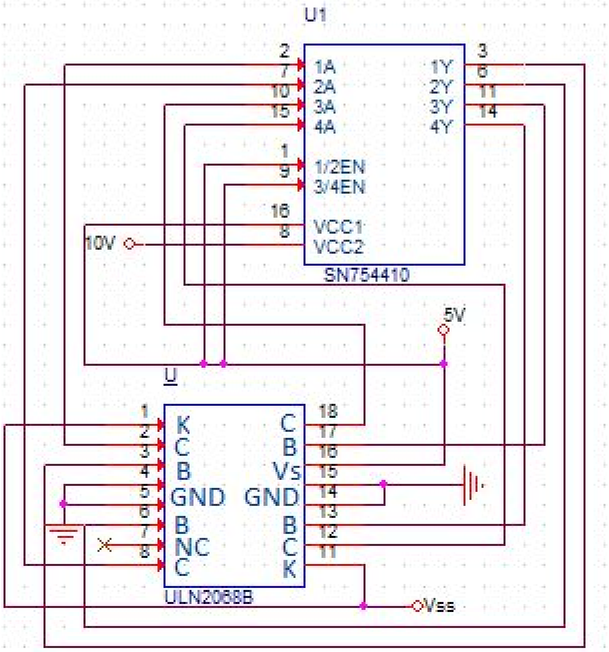
Neat, well annotated circuit schematics (using a schematic capture program like Orcad) with actual resistor values.



Wiring Stepper with ULN2068 on K60

- ULN2068 has 4 Darlington arrays.

Ptucha/Beato 96



Pre-Lab

1) This lab will use Lynxmotion GHM-01 gear head DC motors. The characterization data of the gearbox output is specified with the below curves (Fig. 1) (e. the figures include the gear box). **The gear reduction is 30:1**.

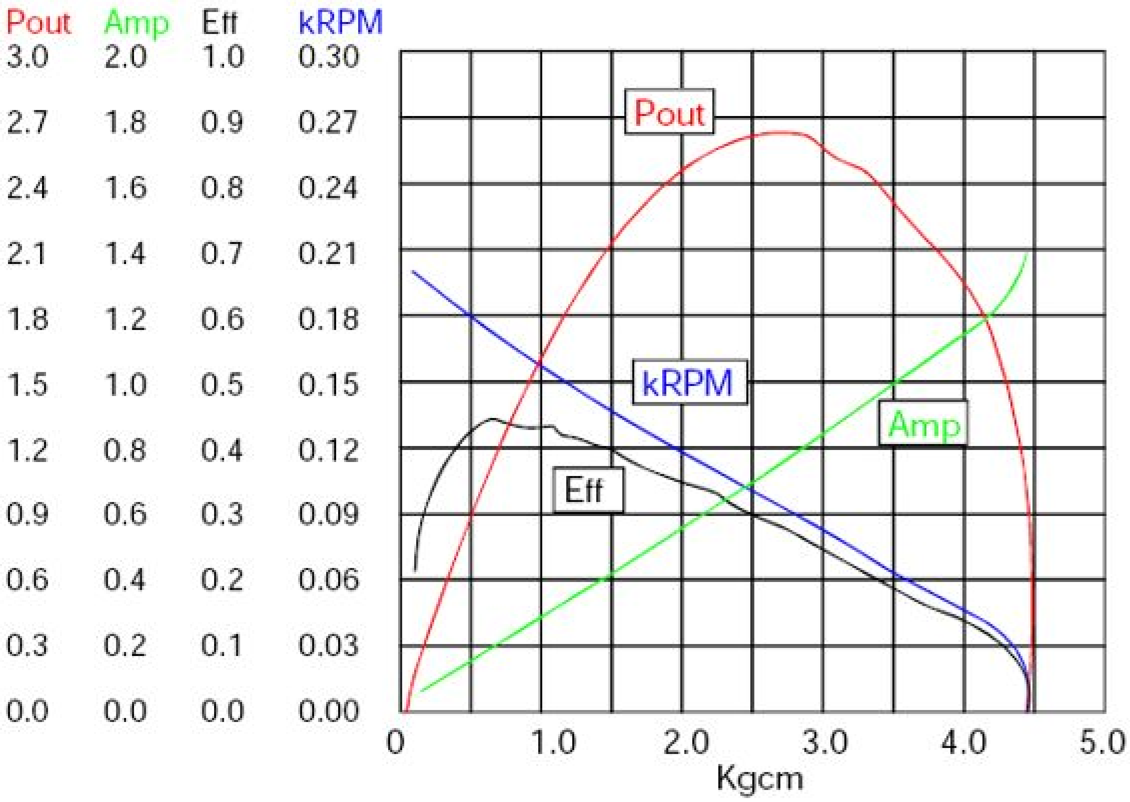


Fig. 1 Motor Characterization curves

From the curves above, estimate:

a) <u>Motor stall torque in oz-in and in kg-cm **before reduction**</u> *divide by 30*

P_{max} = ¼ ($T_{max}w_{max}$) , T_{max}= stall Torque, w_{max} = max rotational velocity

Pout: 3.0

kRPM: 0.3 => RPM:300 =>wmax=300 **=10π

3 = ¼(T_{max}(31.416)) => 3 = T_{max}(7.85398)

T_{max} after = 0.381972 Nm

0.381972 Nm = * * * = 54.1377 oz-in

T_{max} before = 0.012732 Nm

0.012732 Nm = * * * = 1.80459 oz-in

<u>T_{max} before reduction:</u> 1.80459 oz-in *or* 0.012732 Nm b)Maximum current draw *same*

2A

c) Maximum motor turn speed **before reduction** *multiply by 30*

9 kRPM

d) Maximum torque in oz-in **after gear reduction**

T_{max} after gear reduction: 54.1377 oz-in *or* 0.381972 Nm

e) Maximum turn speed in kRPM **after gear reduction**

0.30 kRPM

2) A typical microcontroller can source or sink only about 10-25mA per IO pin. As such, if we were to generate a 5V, 20mA PWM signal from the K60, it would not provide sufficient power to turn our DC motor. To solve this problem, transistors, H-bridges, and motor controllers are used. Each of these amplify the output signal from our IO port to the appropriate voltage and current necessary to drive our motor. For example, we could use NPN and PNP transistors to construct our own H-bridge. In the circuit below (Fig. 2), by having IO A high and IO B low, the motor turns in one direction, and swapping their polarities makes the motor turn in the other direction.

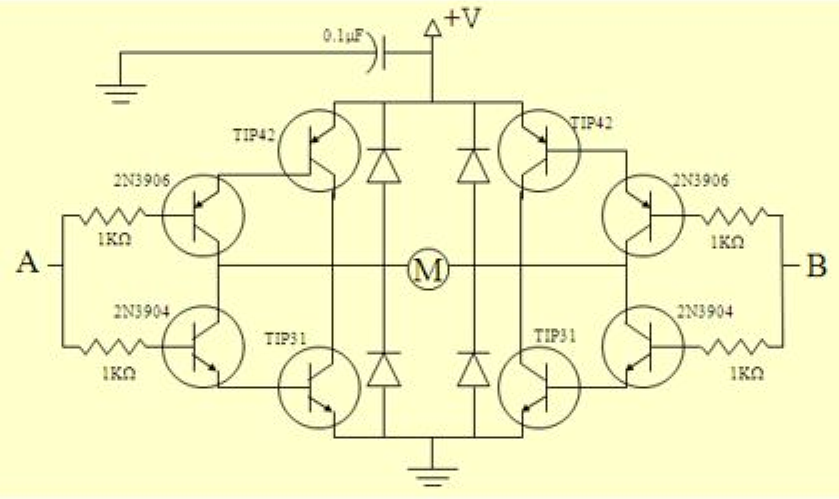


Fig. 2 DC-Motor Control H-Bridges

What is the purpose of the diodes in this circuit?

The diodes (clamp diodes) in this circuit give the current a safe path to the source to prevent voltage spikes from frying the transistors.

What is the purpose of the capacitor in this circuit?

The capacitor absorbs energy when voltage spikes and discharges energy when a voltage dip occurs which helps reduce the line noise (voltage fluctuations) by minimizing the changes.

Why are the transistors in pairs?

The transistors are in pairs to allow for the motor to bedriven clockwise and counterclockwise

Why use 2N3904/2N3906 for some transistors and TIP31/TIP42 for others?

While H-bridge chips can drain 1-2 amps, TIP circuits can drain up to 5 amps which lowers the voltage to the motor

Oscilloscope captures

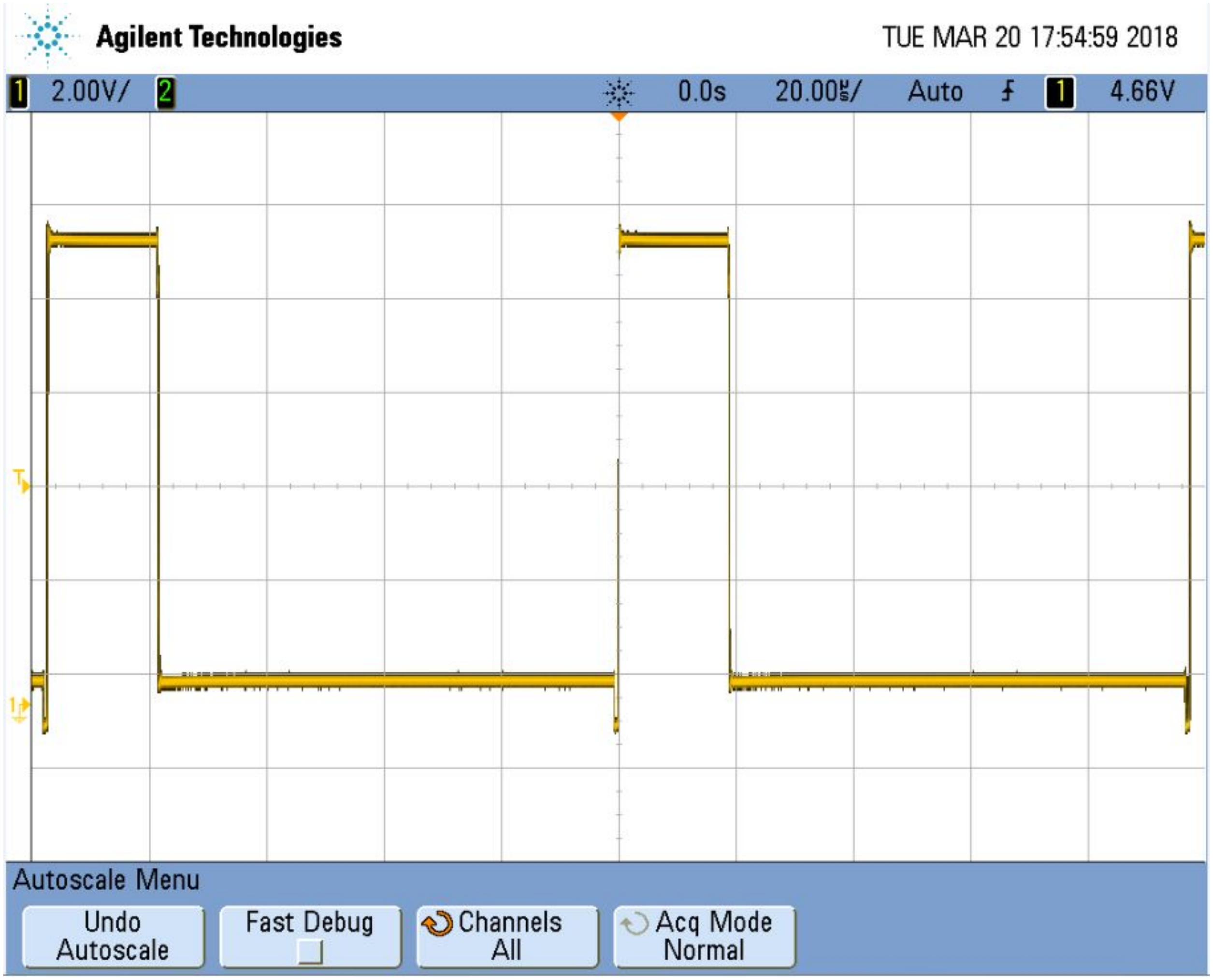
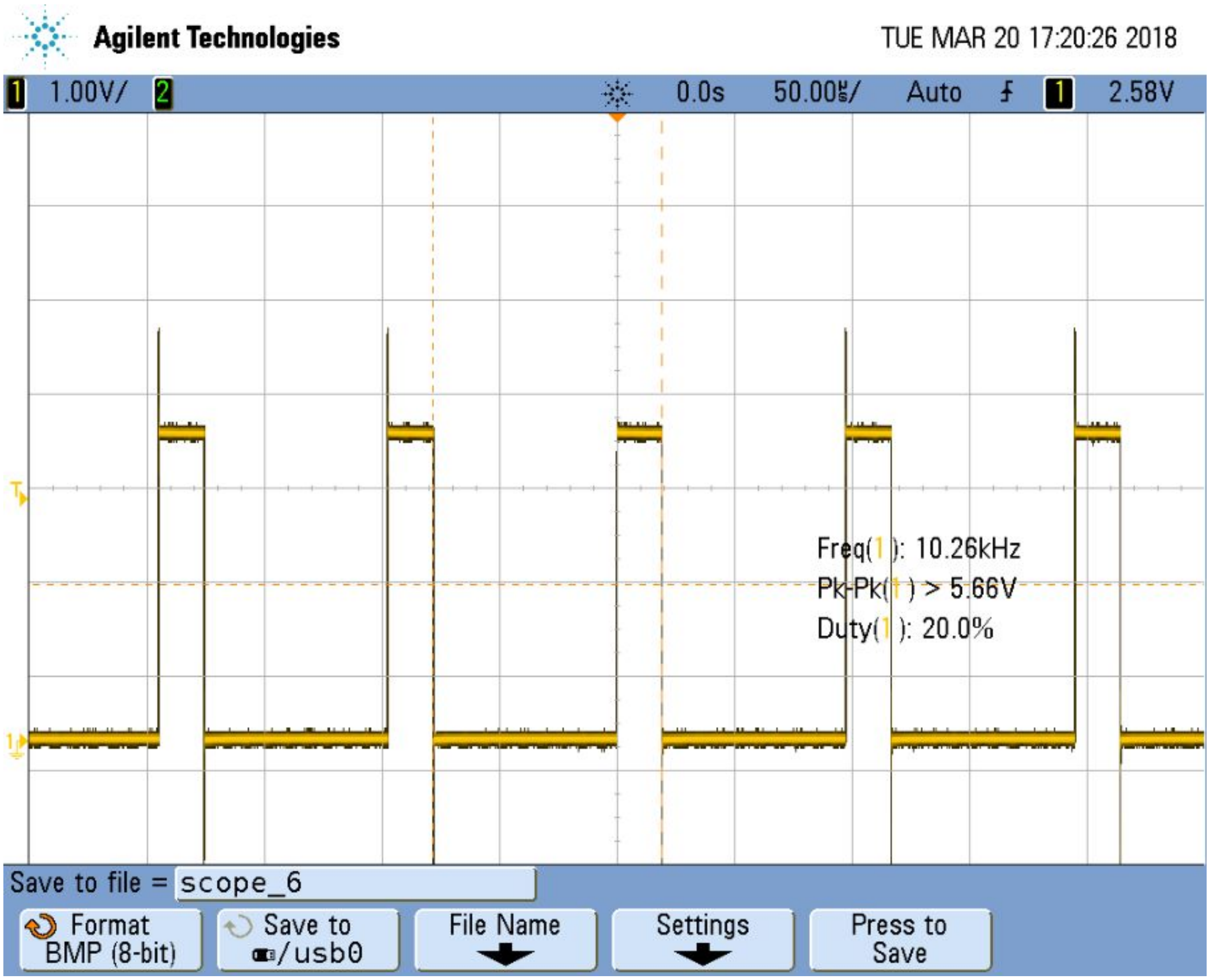


Figure 1: 10V Waveform



Freq(1): 10.26kHz
Pk-Pk(1) > 5.66V
Duty(1): 20.0%

Save to file = scope_6

Figure 2: Pin Waveform

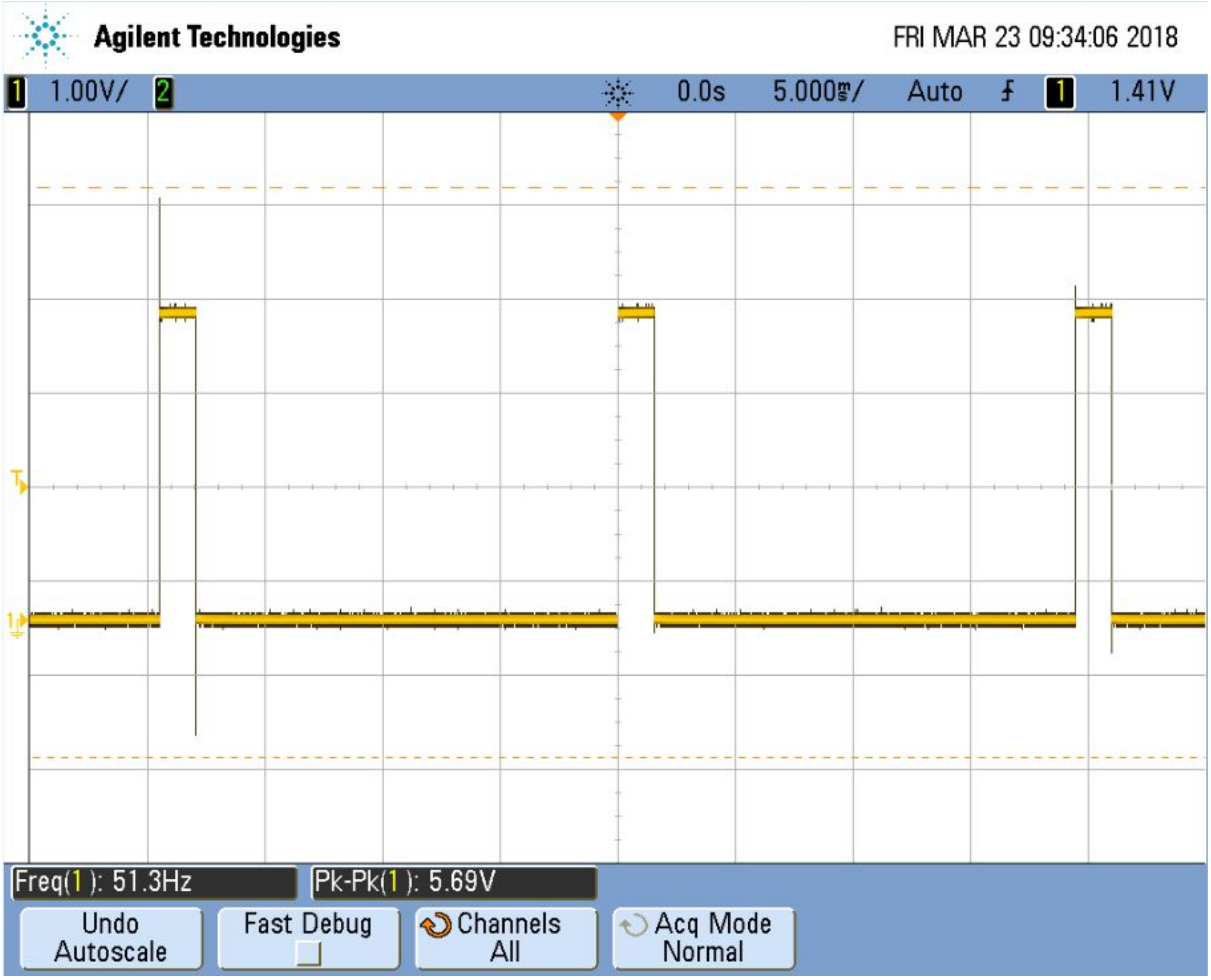


Figure 3: NXP car

High level description of how the code works (include how the FTM works with mod values)

Essentially, the code for part 1, 2 and 4 utilized two pins that had been configured to output timers. These timers acted as PWM to the motor. Of which, there were two, one pin that sent the pulses, the other that acted as the low potential.

Each motor had a specific frequency that was fine tuned for the motor it was driving. For this lab, the frequency remained consistent throughout all test for each individual motor, respectively. In order to increase the speed of the motor, the duty cycle was increased.

The tricky part was in setting the FTM mods, which directly impacted the duty cycle of the output. For the DC motor, the mod was set to the CLOCK value, divided by the Frequency, multiplied by the desired Duty Cycle, all divided by 100. The servo motor was identical, except the CLOCK value was divided by an additional 128. These formulas configured the timers to have the desired PWM effects.

Part 3 did not use the FTM timers, but instead used GPIO output pins. This was done because the stepper motor had 4 coils that needed to be turned on and off in a clockwise/counterclockwise rotation. One of the advantages of using the GPIO pins was that once the loop was created that set the pins and incremented the order over time, a delay was added at the end of the loop. The bigger delay, the slower the stepper motor. This allowed it to act similar to PWM without needing to use a clock.

Answers to in-lab questions

1) Explain how to change speed and direction of turn of the DC motor.
    a) **To change the speed of the DC motor the Duty cycle must be changed. A larger duty cycle corresponds to a faster speed and a smaller duty cycle corresponds to a slower speed. To change the direction, The opposite pin must be pulsed. This is done in the code by sending the pulse from FTM0 channel 3 to FTM0 channel 2. This causes the pin that used to be ground to now have the high potential pulses, which causes the current to move in the opposite direction from before, and thus, make the motor spin in reverse.**

2) The method given to turn the motor at different speeds and direction ties up two PWM lines, one for forward, and one for reverse. Describe an alternate method in your report that uses only one PWM line and one GPIO line.
    a) **An alternative method could be to replace the non-pulsing PWM line with a GPIO pin that is tied to ground. This would have the same effect, since the other PWM cutoff value was getting set to zero anyway.**

3) Stepper motors are good for precise positioning and when there is a need for strong holding torque. You are provided with a sample stepper motor. Is it unipolar or bipolar?

   Unipolar

4) Modify your main function with sample code below, demonstrate turning the stepper motor at different speeds and directions to the TA.

```
// Enable clocks on Port D.
// Configure the Signal Multiplexer for the Port D GPIO Pins
// Configure the GPIO Pins for Output.
int forward=1;
int phase = 0;

while(1) {
//Turn off all coils, Set the GPIO pins to 0.
//Set one pin at a time to high
  if (forward) {
     if (phase == 0) { /*turn on coil A*/; phase++;}                //A,1a
     else if (phase == 1) { /*turn on coil B*/; phase++;}           //B,2a
     else if (phase == 2) { /*turn on coil C*/; phase++;}           //C,1b
     else { /*turn on coil D*/; phase=0;}                           //D,2b
   }
  else   {//reverse
     if (phase == 0) { /*turn on coil D*/; phase++;}                //D,2b
     else if (phase == 1) { /*turn on coil C*/; phase++;}           //C,1b
     else if (phase == 2) { /*turn on coil B*/; phase++;}           //B,2a
     else { /*turn on coil A*/; phase=0;}                           //A,1a
   }
//Note- you need to write your own delay function
   delay(10);      //smaller values=faster speed
}
```

Referring back to the class notes, which stepping mode does this code use?

   Full Step, Low Torque mode

Lab Sign Off Sheet

Lab 6 Sign Off Sheet

Team Member Names: David Feng, Joel Yuhas, Alice Fischer

Pre-Lab	TA/Instructor: Date: 3/23/18 Comments:
Part1: Demonstrate 20% duty cycle signal at 10KHz	TA/Instructor: Date: 3/20/2018 Comments:
Part2: Demonstration of DC motor functionality	TA/Instructor: Date: 3/20/2018 Comments:
Part3: Demonstration of stepper motor functionality	TA/Instructor: Date: 3/23/18 Comments:
Part4a: Demonstration of NXP Cup signal generation on o-scope	TA/Instructor: Date: 3/23/18 Comments:
Part4b: Demonstration of NXP Cup car servo and rear motors being driven simultaneously.	TA/Instructor: Date: 3/23/18 Comments: