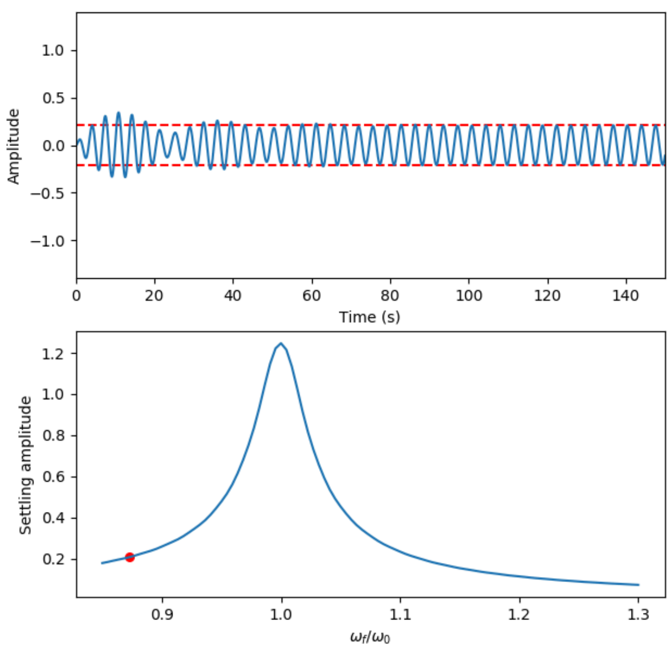
Lab objective

In this lab, we will be using an external motor to add a periodic force to the circuit. The forces on the the Arduino are the sum of spring, the damping force from air resistance and the external periodic force:

$$-kx - b\frac{dx}{dt} + F_0 \sin(\omega t) = m\frac{d^2x}{dt^2}.$$

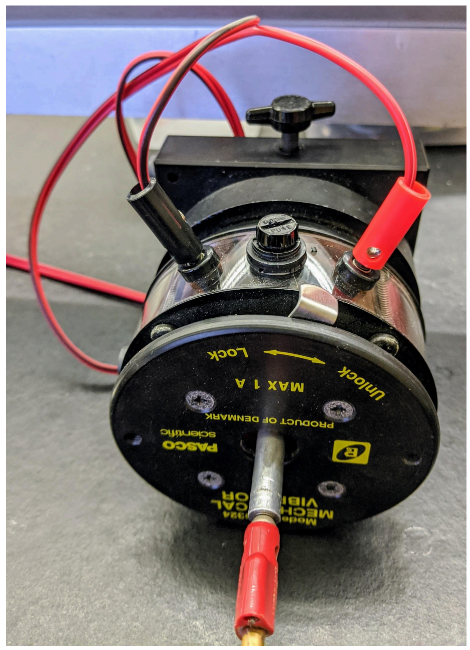
In this lab, we will be changing the frequency ω of the forcing frequency using an Arduino driven motor. The relation between the forcing frequency and the natural frequency of $\omega_o = \sqrt{k/m}$ will set the amplitude of the settled oscillations. Below is a plot of the quantities you will need to measure. You will be measuring the amplitude of the oscillations as measured by the ultrasound after a settling period for several values of the forcing frequency to reproduce the bottom curve.



Components

Motor driver

The motor driver takes a ground and a voltage input from a voltage source, such as our Arduino (shown on the right picture) and will move the clip shown on the (shown on the left picture) up and down. We can attach our spring to the motor in order to force the oscillations

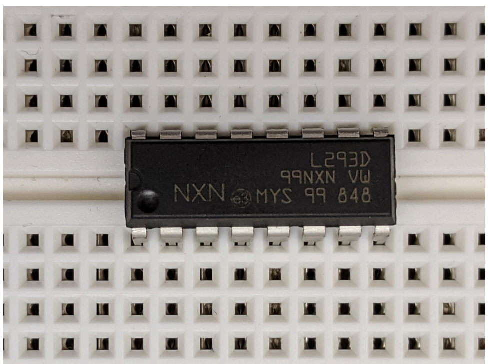


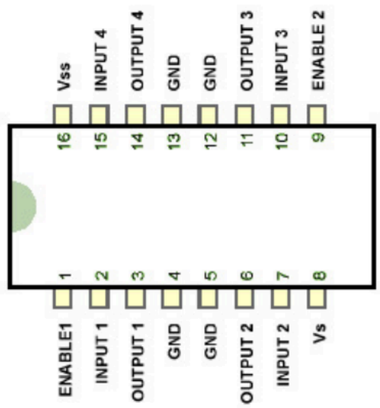
LD93d

The LD93D is an integrated circuit that uses voltage pulses from the Arduino to drive the motor. There are two inputs as well as the GND from the Arduino to the LD93D. One enables the output of signals from the Arduino to the motor driver. The Arduino enable pin is marked in the control software with the ENABLE variable. In the setup function, we enable the LD93D output with
digitalWrite(ENABLE, HIGH);

The other pin, marked as DIRA sets the direction of the motor to go up or down. Inside the loop, the code will write
digitalWrite(DIRA, 1);
to drive the motor in one direction and
digitalWrite(DIRA, 0);

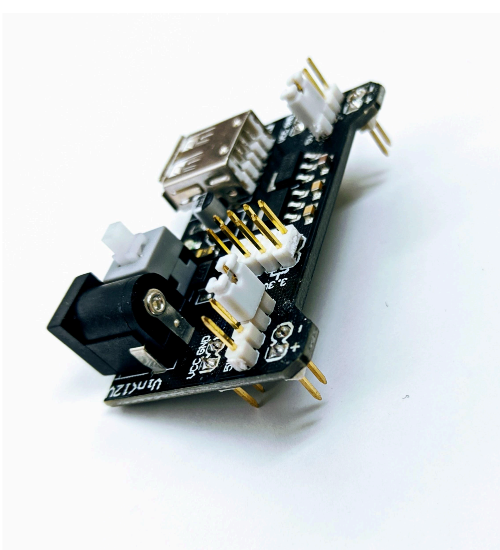
to drive the motor in the opposite direction. In order to create a periodic signal, we will alternate between going in one direction to the other with a delay in the middle. Below is a picture of the component as well as the wiring diagram. Note the orientation of the notch on the left side of the diagram as well as the component

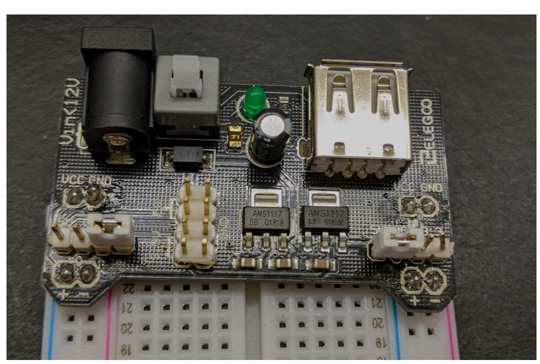




Power supply module

The power supply module allows us to add additional current to the circuit, allowing us to increase the amplitude of the motor driver. The module attaches to the board, and we will plug in an additional battery to supply the extra current.

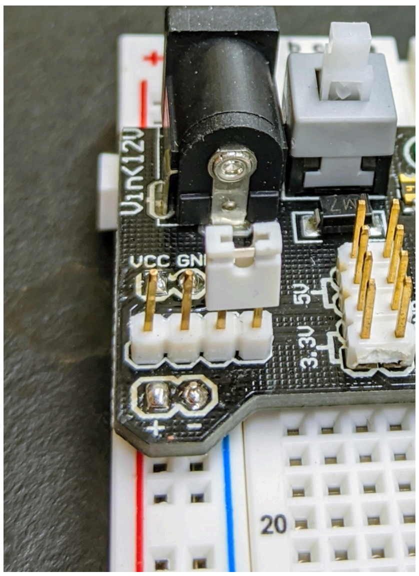




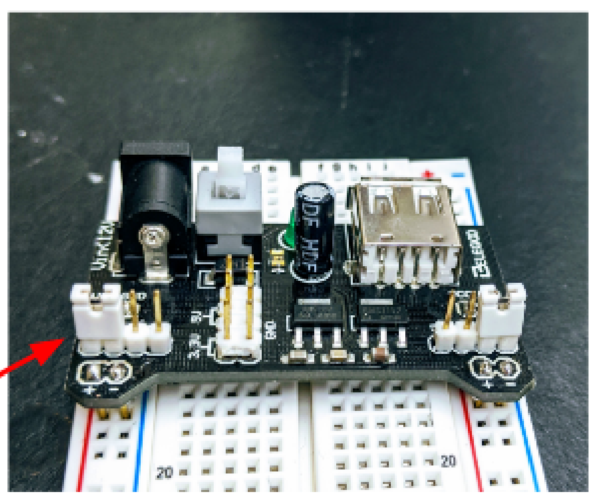
Instructions

Make sure that the Arduino as well as the power supply module are unplugged before assembling the circuit.

1) Attach the power supply module to one end of your breadboard (either direction works).
2) Ensure that the jumper on the left side is on the correct set of pins as shown in the lower image. If the jumper is on the wrong pin, you can pull it off as shown in the upper image

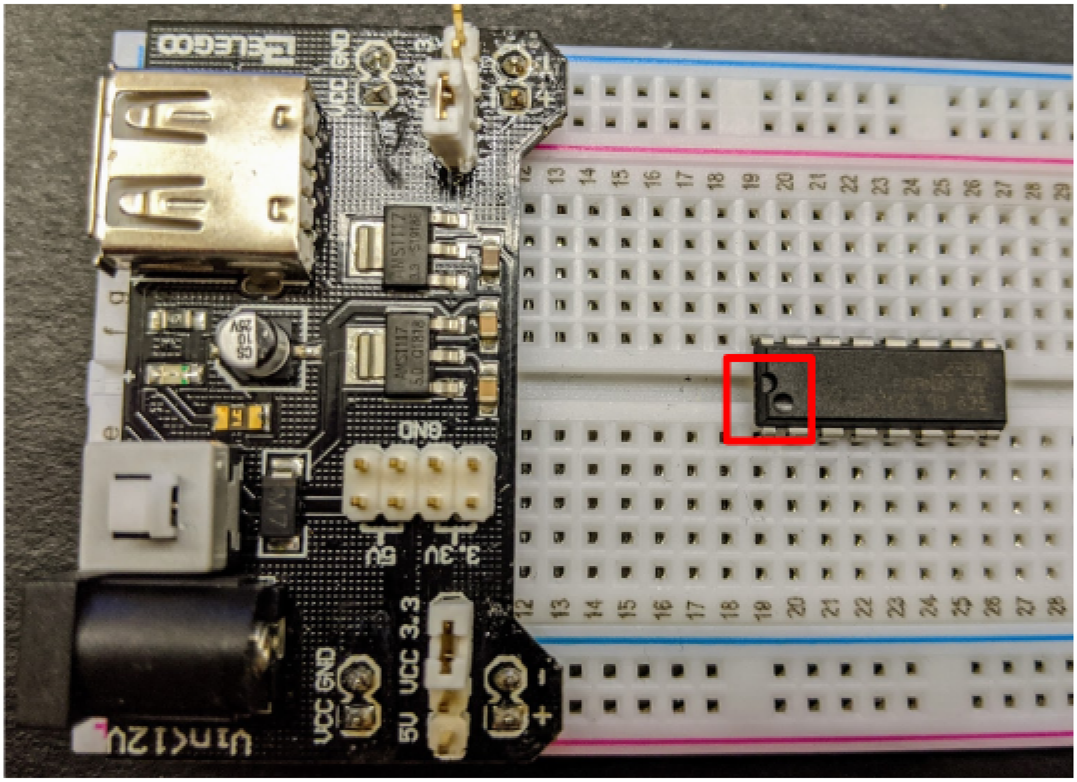


Correct placement:

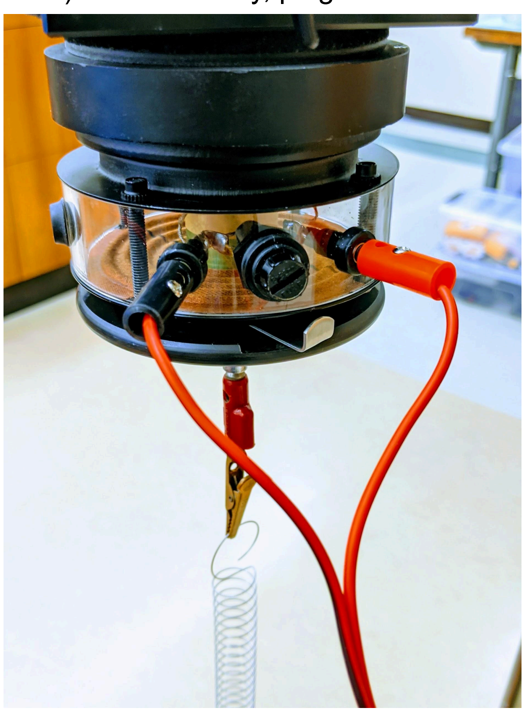


Jumper

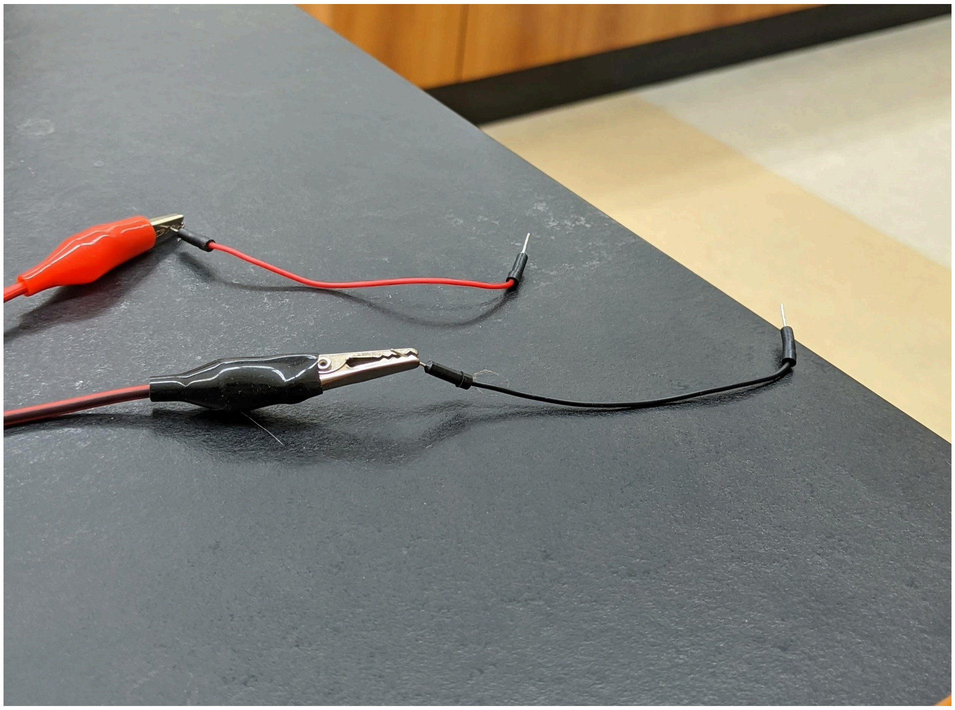
3) Position the L293D such that the notch points towards the power supply module



4) If not already, plug in the banana clips leads to the motor driver.

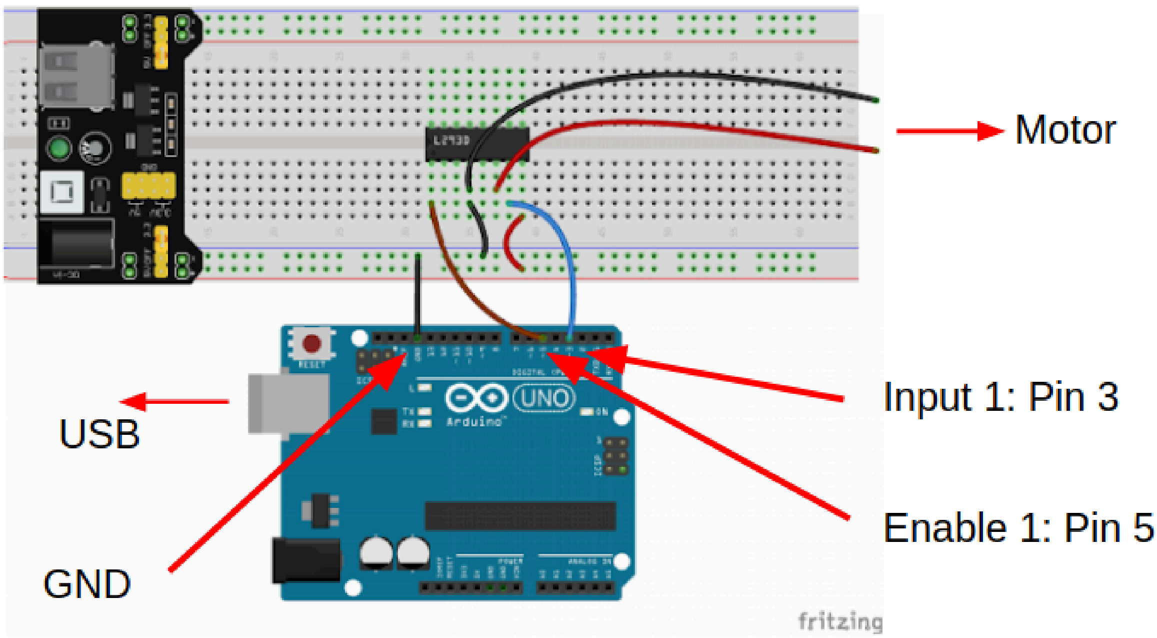


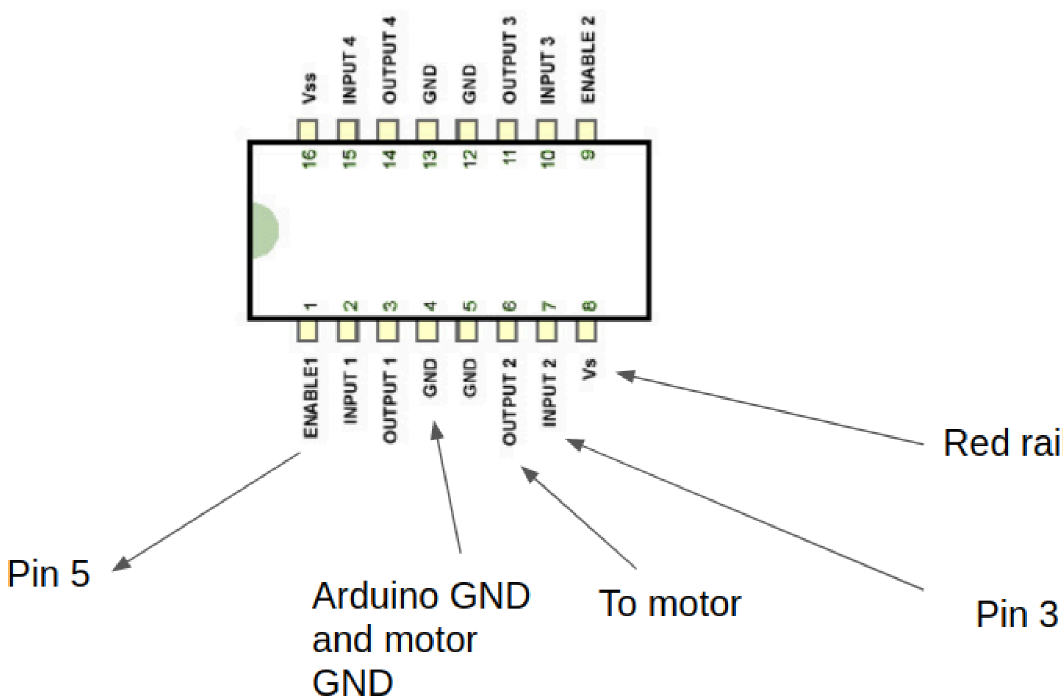
5) Attach Arduino wires to the banana clips



6) Mount the motor drive onto the spring stand

7) Use the diagrams for the forcing motor Arduino as well as the L93D



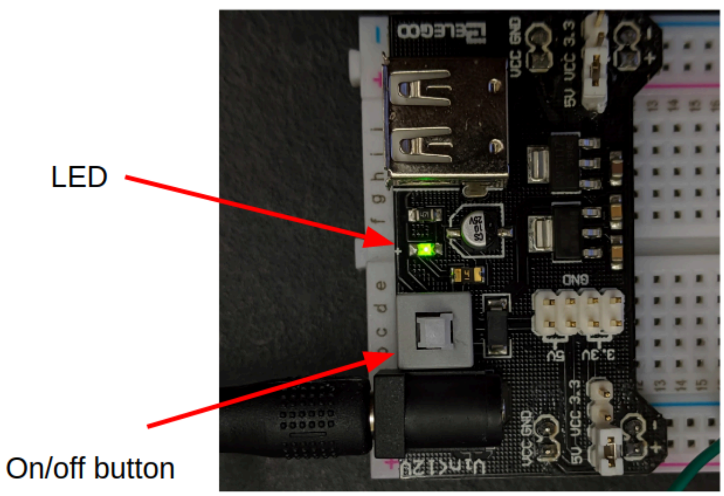


9) Upload code that reads ultrasound data and transmits it onto Arduino with a wifi and ultrasound chip which will be the transmitting chip. We will not be using the accelerometer.Attach the Arduino to the bottom of the spring.

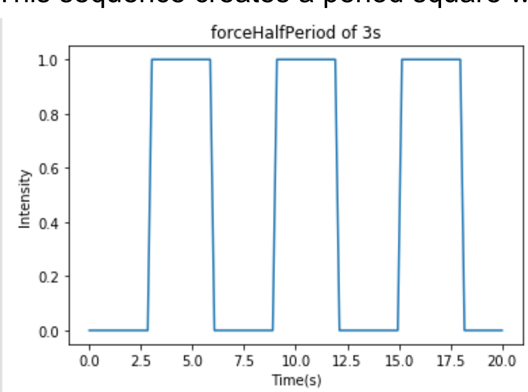
11) Do not turn on any oscillations for this step. Find the natural frequency of the spring with an Arduino by measuring the time the Arduino takes to complete 20 cycles. Note this frequency in your assignment slides.

12) Plug the USB into the Arduino but do not plug in the battery for the external power supply. Note the range of motion of the spring.

13) Now plug in the battery to the external power supply. Make sure that the LED on the power supply is turned on. If not, press the white on/off button. You should immediately see the difference in amplitude.

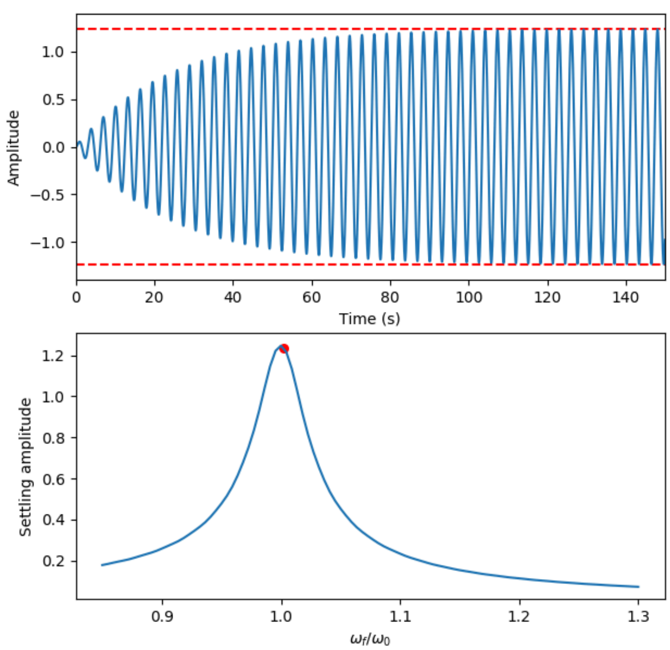


15) Upload the [forcing code to your Arduino](). In this code, you will change the variable forceHalfPeriod to change the delay between moving the spring down and moving it back up. This sequence creates a period square wave as shown below



16) Now we will use the forcing code to make another measurement of the natural frequency. When the forcing frequency is equal to the natural frequency, the amplitude of the oscillations are maximal. We have two ways of evaluating if the forcing frequency is at the natural frequency. The fastest way is observe the movement of the forcing motor and the motion of the Arduino.

- Set the forceHalfPeriod to half the natural period of the spring measured earlier as a starting guess
- Hold the reset button on the forcing motor Arduino and steady the motion of the transmitter Arduino
- Release the reset button and observe the motion of the motor and the motion of the Arduino
- If the Arduino is oscillating faster than the forcing motor, increase the forceHalfPeriod, and if the forcing motor is faster, decrease the forceHalfPeriod
- Once you are close to the correct period, reset the system and hold the Arduino steady
- Open the Serial plotter and observe the ultrasound sensor data. As shown in the plot below, a system at resonance should have the amplitude steadily increase. We will not be able to get the system at exact resonance, but you will find the closest value. Note down the maximum amplitude for several different frequencies. Note the best frequency in your slides.



17) Write out a file and plot the time vs ultrasound data at this resonant frequency

Plot and show the time vs amplitude plot to your TA before continuing

18) Alternate between taking 10 oscillations of data for frequencies above and below the resonant frequency, for as many datasets you can as time permits with a minimum of 3 above and 3 below a maximum of 16. Having more datasets will result in better analysis. Take steps large enough to see an appreciable difference

- REMEMBER : After changing the frequency, give the system enough time to stabilize before recording data.