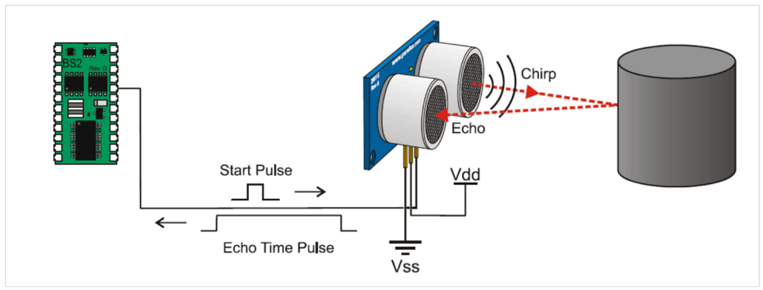
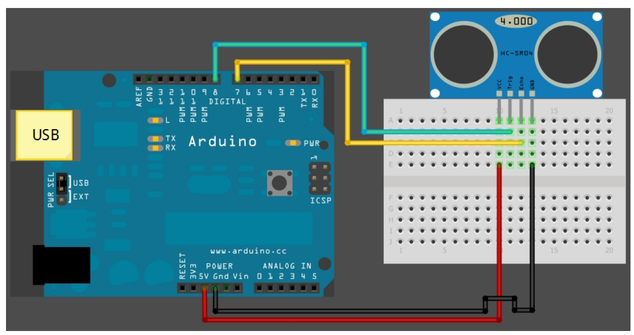
Robert Florence
Prof. Claveau
Robotics
Spring 2017

# Assignment 2

- Ultrasonic Sensor

1). The basic underlying principle of an ultrasonic sensor is using our knowledge of the speed of sound (within certain mediums), and bouncing high frequency sound waves off a surface and measuring how long it takes to receive the echo from those sound waves. After the sound waves trip time has been measured, it is possible to calculate just how far the object/surface is away from the sensor. An ultrasonic sensor consists of two main components; an emitter (a speaker), and a sensor (receiver). The emitter plays a very specific, and very high frequency sound, and the receiver 'listens' for a response among all the other noise.

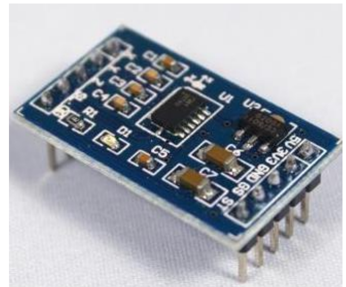


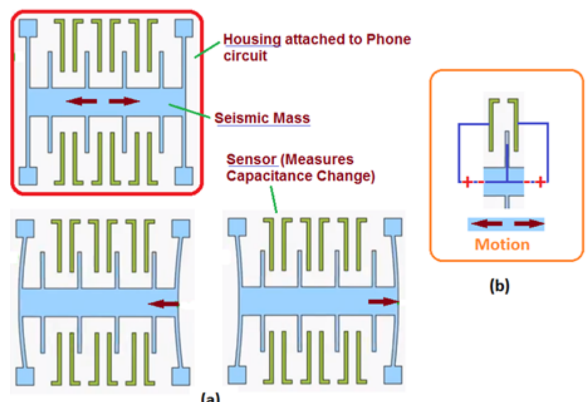



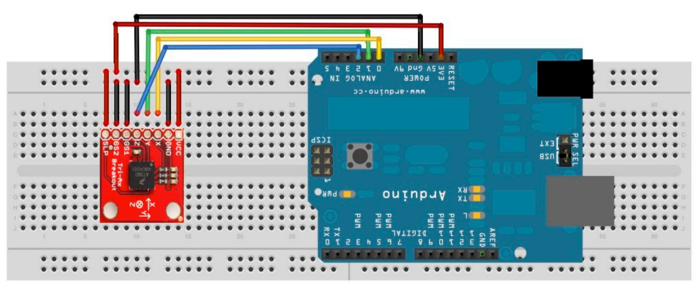

To tie those components together, a controller is used to switch back and forth between emitting a frequency and receiving signals, thousands of times a second. Because the speed of sound is relatively constant in air, it is assumed that the waves don't lose energy/momentum while in transition from emitted to received, so we can calculate (using a formula like d=rt) a very close approximation of the full trip distance, halving that distance gets the distance from the sensor to the object itself. The ultrasonic sensor has a relatively simple circuit diagram, as shown above, in which the emitter, and the receiver have their own input and output connection to the Arduino board. The ultrasonic sensor also has an on-board ADC which converts the received signal into a digital one. To test the accuracy of the ultrasonic sensor, we printed out values onto a console and graphed with very minor delay (10ms) in between, which lead to a low latency. The output of the program was the physical distance in cm, and the sensor had trouble with values under two to three cm. The sensor also seemed to have a maximum value (around 3 meters), which might have been a timeout issue where the sensor might not receive the signal back within a certain time and assume it to be lost. Outside of those issues, the graph seemed to track motion with a very high precision and didn't seem to have any extraneous values.

- Accelerometer

2). The basic structure of an single-axis electronic accelerometer relies on the concept of a seismic mass shifting when the mass of the structure shifts. It is based on the idea of when a sufficient force is applied to an object, it changes velocity in some direction, this changing of velocity is called acceleration. Anything with a constant velocity has an acceleration of zero, which can be equated to Newtons first law of motion. Electronically this is accomplished with a, on a microscopic scale, a seismic mass layered between sensor plates that are interleaved with fins of the seismic mass. When the object accelerates in the direction perpendicular to the plates, it causes the seismic mass to move in that direction, which causes a change of capacitance in the sensor plates. The more extreme the acceleration, the higher the change in capacitance. This change in capacitance can be directly converted to a digital signal for a controller to interpret. This idea is combined into a 3-axis accelerometer by three separate single axis accelerometers each aligned within their own separate axis (x, y, and z). This combination results in any acceleration in our three dimensional world being able to be tracked. These 3 outputs could also be combined to mathematically calculate in which 3 dimensional direction the object is accelerating. When using an Arduino as a controller their is 3 output pins (x, y, and z), which can be directly mapped to three input pins on the board. My group partner and I attempted to use the accelerometer to monitor/measure heart rate from the radial artery in the wrist. The accelerometers was precise enough to pick up minor changes like the slight pulse, which is not normally visible to the human eye. We designed a program to take a running average so, as long as the accelerometer was held still for a few seconds it achieved a base value to measure any discrepancies from that base value. Unfortunately, possibly due to the setup of resistors in our circuit, we received quite a bit of noise data, which disrupted our consistent values. Outside of the errors, we did semi-successfully get a steady measurement of a my heart rate.





- Electro sensor(Photodiode)

3). A photodiode is a semiconductor device that converts light into current. The current is generated when photons are absorbed in the photodiode. A small amount of current is also produced when no light is present. A photodiode is a p–n junction or PIN structure. When a photon of sufficient energy strikes the diode, it creates an electron-hole pair. This mechanism is also known as the inner photoelectric effect. Of the three sensors used, the photodiode is the only one that doesn't need an Arduino, or similar controller to witness the effects of the sensor on a circuit. A relatively simple circuit diagram can be set up like the pictures on right. One side of the photodiode can be connected to a transistor to regulate the flow of current through the circuit(as shown in picture 2), then could be routed through an LED to show how the light sensor effects current flow. When the sensor detects less light, less current flows through the transistor (can be thought of as a valve), and less through the LED. The opposite is also true, when the sensor detects more light, more current flows through the system and the LED glows brighter. The performance of a photodiode depends on the quality of the photodiode itself. The latency of the communication of a photodiode and a controller, such as an Arduino, is much lower because of the extremely high speed of light. Any activity to restrict light, or to increase the level of light immediately affects the brightness of the LED.