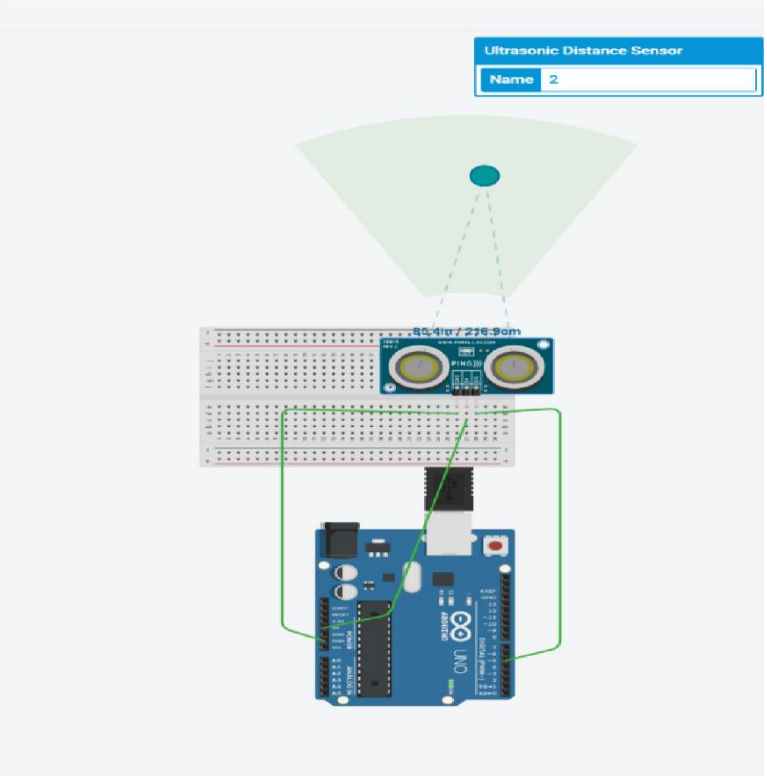
LAB REPORT #9

(Names) Kenil Patel, Austin Guiney
(Date) 5/4/2021
Screenshot and components



Ultrasonic Distance Sensor: Send waves outward , the waves would bounce back to the sensor and the sensor can measure the distance to the object based on time it takes to hit the object and bounce back.

Summary:
First, I set up the hardware and then copied the code over to the project. I took a test run and the serial monitor prints out the distance of the object from the Distance Sensor. In order to calculate the velocity, I would need to calculate the displacement between the current distance and the last distance of the object as I move the object around. The displacement is the vertical change in distance from the sensor. I made a global variable int lastD to store the last distance. In the loop, I initialize a float v to store velocity of the object. Then, I calculate v using d and lastD and constant 10/29.00. The constant 10/29.00 acts as 1 dividing change of time. So v is to be (float)(d - lastD)* 10/29.00 After the calculation, I would store lastD as d in order to update lastD to be used in the next loop. After that, I print out the current distance and the velocity. Result: The serial monitor prints velocity values closely related to the sample outputs. The velocity would be negative if the object moved closer to the sensor and positive away from the sensor.

1. **How is the input from an analog pin different from the input from pulseIn()?**

Pulsin() returns the amount of time in ms a digital pin is high for while analog pin gives an integer between 0 and 1023 based on the voltage of the circuit from 0 to 5 V.
Conclusion:
I learned how to use a Ultrasonic Distance Sensor and how its ability to measure the distance from the Sensor. Since it would only know the difference in distance if the object traveled straight away from it, not sideways since the radius of going sideways would be the same. It is also important the wave has to hit the object and go back to the sensor so the actual distance would be half. Since the delay time was too fast, I increased the time in order to test out my velocity with the sample outputs. I also had trouble saving the previous distance in the loop, but having an extra global variable like d helped a lot. const int pingPin = 5;

```
int d;
int DELTA_T =
1000; int lastD = 0;
void setup() {
  Serial.begin(9600);
}

void loop() { float v; //lastD = d;
  d = ping(pingPin); v = (float)(d
  - lastD)* 10/29.00; lastD = d;
  Serial.print("D: ");
            Serial.print(d);
  Serial.print(" | ");
  Serial.print("V: ");
  Serial.println(v);
  delay(DELTA_T);
}

/*returns the distance in centimeters to where
* the wave hit something and turned around
*/ int ping(int
pingPin) {
  int duration;


pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2); digitalWrite(pingPin,
  HIGH); delayMicroseconds(5);
  digitalWrite(pingPin, LOW);
```

```
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin,
  HIGH);

  return duration / 29 / 2;
}
```