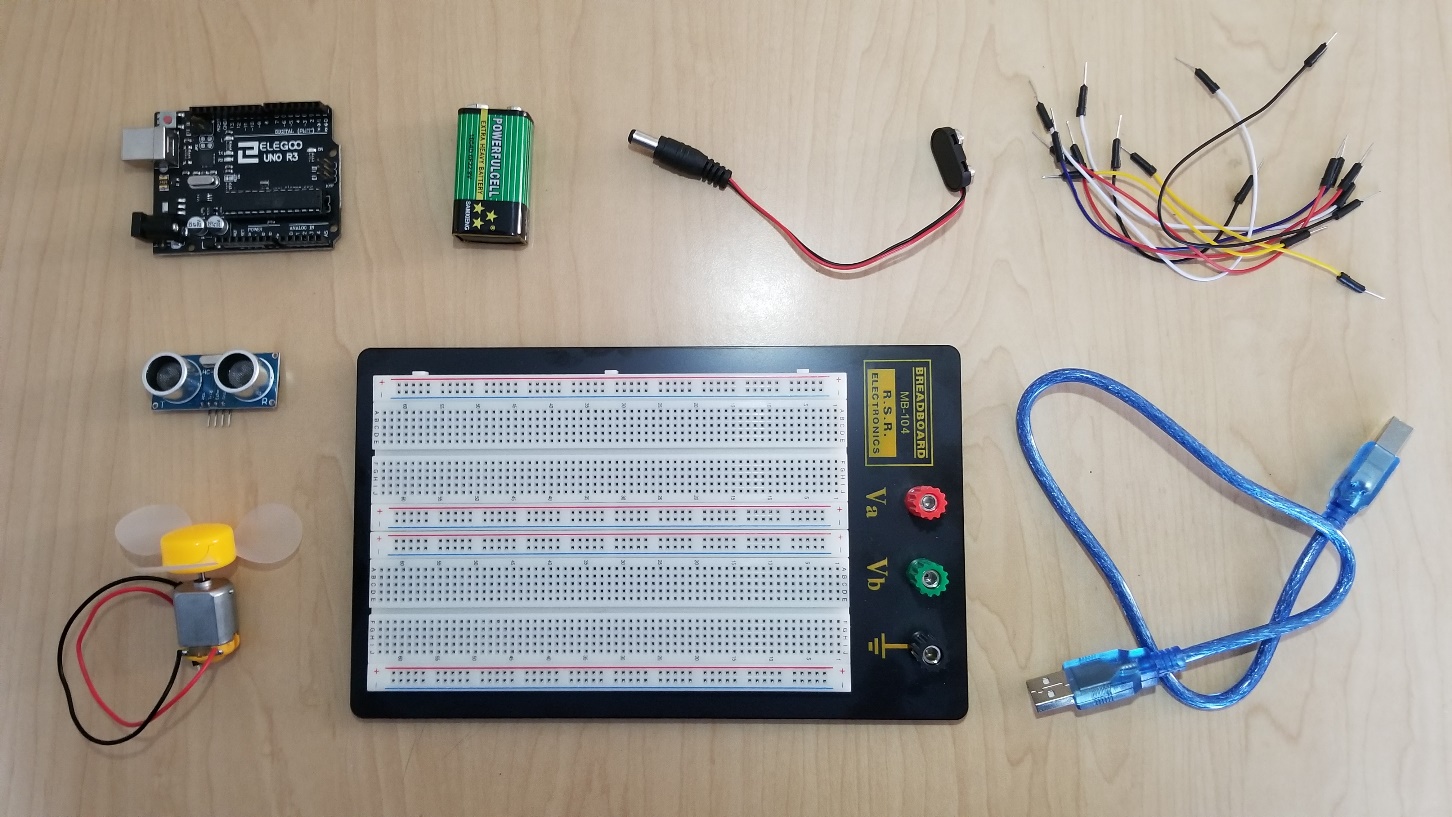
DrivING motor with an Ultrasonic Sensor: An Arduino Project

Emilie Jenkins

Contents

[Introduction 6](#_Toc527041812)

[Required Equipment 6](#_Toc527041813)

[Key 6](#_Toc527041814)

[Instructions 7](#_Toc527041815)

[Hardware 7](#_Toc527041816)

[Software 8](#_Toc527041817)

[Disconnecting from the Computer 10](#_Toc527041818)

[Conclusion 11](#_Toc527041819)

[Additional Information 11](#_Toc527041820)

[Common Issues 12](#_Toc527041821)

[Appendix A – Specifications for Arduino Uno 13](#_Toc527041822)

[Pin Layout 13](#_Toc527041823)

[Technical Specifications 13](#_Toc527041824)

[Appendix B – Specifications for Ultrasonic Sensor 14](#_Toc527041825)

[Pin Layout 14](#_Toc527041826)

[Technical Specifications 14](#_Toc527041827)

[Appendix C – Software 15](#_Toc527041828)

[Complete Code 15](#_Toc527041829)

[Cited Images 16](#_Toc527041831)

[Intro 17](#_Toc527041832)

[Key 17](#_Toc527041833)

[Appendix A 17](#_Toc527041834)

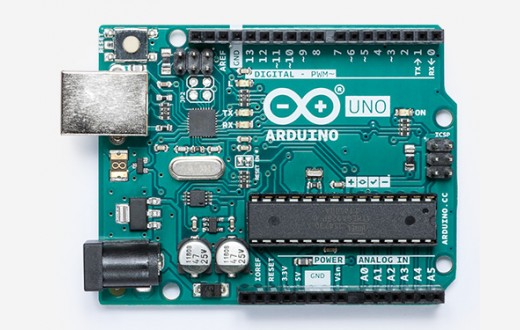
[Appendix B 17](#_Toc527041835)

[Instructions, Conclusion, and Appendix C 17](#_Toc527041836)

# Introduction

The purpose of this instruction manual is to provide those with limited Arduino experience a step-by-step tutorial on how to properly drive a DC motor using an ultrasonic sensor and an Arduino Uno. An ultrasonic sensor uses sound waves to measure the distance between two points, therefore interfacing ultrasonic sensors and motors is common practice in autonomous robotics where you may need to follow a wall without getting too close. This document will act as a guide through completing the hardware, writing the software and disconnecting from the computer. It is important to follow the directions in this manual carefully so you do not damage the sensor, motor, or Arduino. This process should take less than an hour to complete, however times can vary based on experience and the number and types of errors. Product specifications for the Arduino and the ultrasonic sensor can be found in the Appendices beginning on page 10.

## Required Equipment



* 3-6V DC Motor
* 9V Battery
* Arduino Uno
* Battery Snap Connector
* Breadboard
* 5 Male-to-Male Connector Wires
* HC-SRO4 Ultrasonic Sensor
* Laptop
* USB Connector

Figure 1: Arduino Uno

## Key

Tip - Helpful information that will make the process easier and help prevent errors.

Warning – Failure to follow warnings could result in damage to the hardware.

# Instructions

## Hardware

This section will instruct you on how to properly connect the ultrasonic sensor to the Arduino Uno. At the end of this section, all components except the battery should be connected and ready for the software to be downloaded. Be sure to reference Appendices A and B for more information on pin layouts and product specifications.

1. Insert the pins of the ultrasonic sensor into the breadboard so all pins are in different rows.
2. Connect a wire from the ground (GND) pin on the Arduino to the ground column on the breadboard, delineated by a blue line.
3. Locate the breadboard row containing the ultrasonic sensor’s ground (GND) pin.
4. Connect a wire from this row to the ground column on the breadboard.

Warning 1: Be sure to connect voltage supply to the correct pins. Failure to do so can result in burned out pins.



1. Locate the breadboard row containing the ultrasonic sensor’s supply (Vcc) pin.
2. Connect a wire from this row to the 5V supply pin located on the Arduino.

Tip 1: Use different colored wires to connect the pins. This will help you differentiate them if you need to troubleshoot.

1. Locate the breadboard row containing the sensor’s trigger (Trig) pin.
2. Connect a wire from this row to one of the digital pins on the Arduino. Figure 2 on page 4 shows the trigger pin connected to pin 12.
3. Repeat steps 5 and 6 for the sensor’s echo pin, connecting it to pin 13 on the Arduino as pictured in Figure 2.
4. Connect the ground wire (usually black) on the DC motor to the ground column on the breadboard.
5. Connect a wire from the DC motor’s supply wire to one of the digital pins on the Arduino. For this example, we will use pin 8.
6. Use the USB connector to connect the Arduino to a USB port on your laptop.

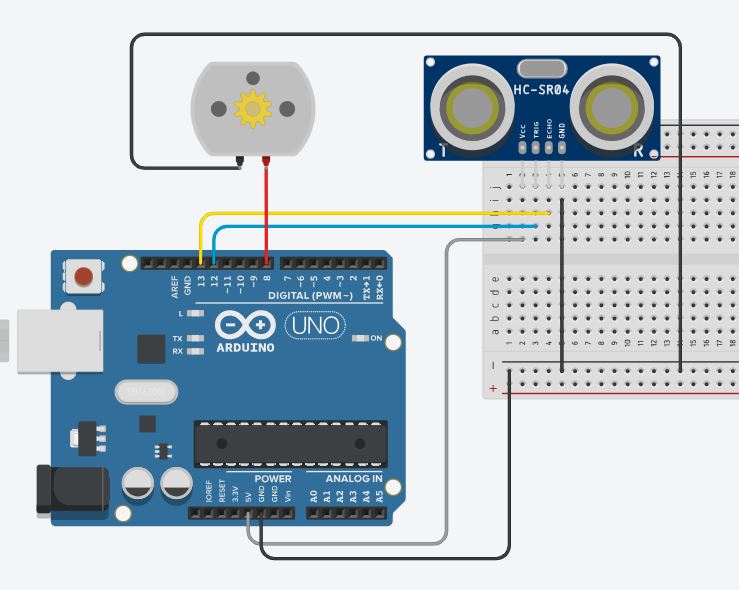


Figure 2: Complete wiring for the Arduino, Ultrasonic Sensor, and Motor.

## 

## Software

In the software portion of these instructions, you will learn where to find the open source example programs in the Arduino Interactive Development Environment (IDE). You will also learn how to modify them to properly drive the motor and sensor. By the end of this section, you should be able to start and stop the motor by moving your hand in front of the sensor. To start this section, be sure to have completed the hardware portion and have connected the Arduino to your laptop via USB. A full reference program can be found in Appendix C on page #.

1. Download the Arduino IDE. This software is available to download for free on all operating systems at https://www.arduino.cc/en/Main/Software.
2. Click on the file named “Ping” under File > Examples > Sensors.

Tip 2: The example programs are a great way to get started if you are new to using the Arduino library.

1. Select the listed port under Tools > Port.
2. On line 25, set the constant int pingPin equal to the number of the Arduino pin you connected the sensor’s trigger pin to. In our example, it was pin 12.
3. Copy the line of code you just modified, and paste it onto the two lines directly below so you have three variable declaration statements.
4. On line 26, change the variable name to echoPin.
5. Set echoPin equal to the number of the Arduino pin you connected the sensor’s echo pin to. In our example, it was pin 13.
6. On line 27, change the variable name to motorPin.
7. Set motorPin equal to the number of the Arduino pin you connected the motor’s signal wire to. In our example, it was pin 8.
8. On line 45, change the delay time from 5 to 10 milliseconds.
9. On line 51, rename the variable from pingPin to echoPin.
10. Repeat step 11 on line 52.
11. On line 63, insert the code provided in Figure 3. This code turns the motor on when something is detected within 10cm of the sensor. Once, nothing is detected within that range, the motor will turn off.

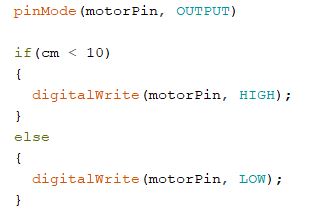


Figure 3: This code checks to see if motor should be on or off.

1. Make sure the Arduino is connected to your laptop by checking to see if the status light on the Arduino is green.
2. Click the checkmark in the upper left-hand corner to compile the code.
3. If your code successfully complies, click the arrow directly to the right of the verify button to upload your code to the Arduino.

Tip 3: Something not working? A Common Issues section has been included in the conclusion.

1. Open the serial monitor by clicking on the magnifying glass in the upper-right hand corner.
2. Move your hand in front of sensor and check to see if the distance reported in the serial monitor is reasonable.
3. Check to make sure when your hand is within the distance set in step 13, that the motor turns on, and when your hand is outside the range, the motor turns off.

## Disconnecting from the Computer

At this point, you are practically done, however you might not always want your Arduino assembly to be attached to the computer. Luckily, once we have uploaded the code to the Arduino, it is only a matter of finding a power source to run it that stands between us and a mobile assembly. In this section, you will learn to connect the battery and perform final checks. Make sure your sensor is giving accurate readings, and your motor is behaving properly before beginning this section.

1. Remove the USB cable connecting the Arduino and the computer.
2. Snap the battery connector onto a 9V battery.
3. Plug the battery connector into the Arduino.
4. Preform the same checks illustrated in steps 18 and 19 of the Software section in order to confirm the final product is working.

Figure 4: 9V Battery Snap Connector

# Conclusion

If you are now able to turn the motor on and off using your hand, you have successfully completed this tutorial! Now that you have learned how to use an ultrasonic sensor to drive a DC motor, you are well prepared to interface many other types of sensors with many other types of motors, as the process described in this document is essentially universal. The best way to improve with Arduino programming is to continue taking on small projects, and to help in this endeavor, links have been provided in the Additional Information section below.

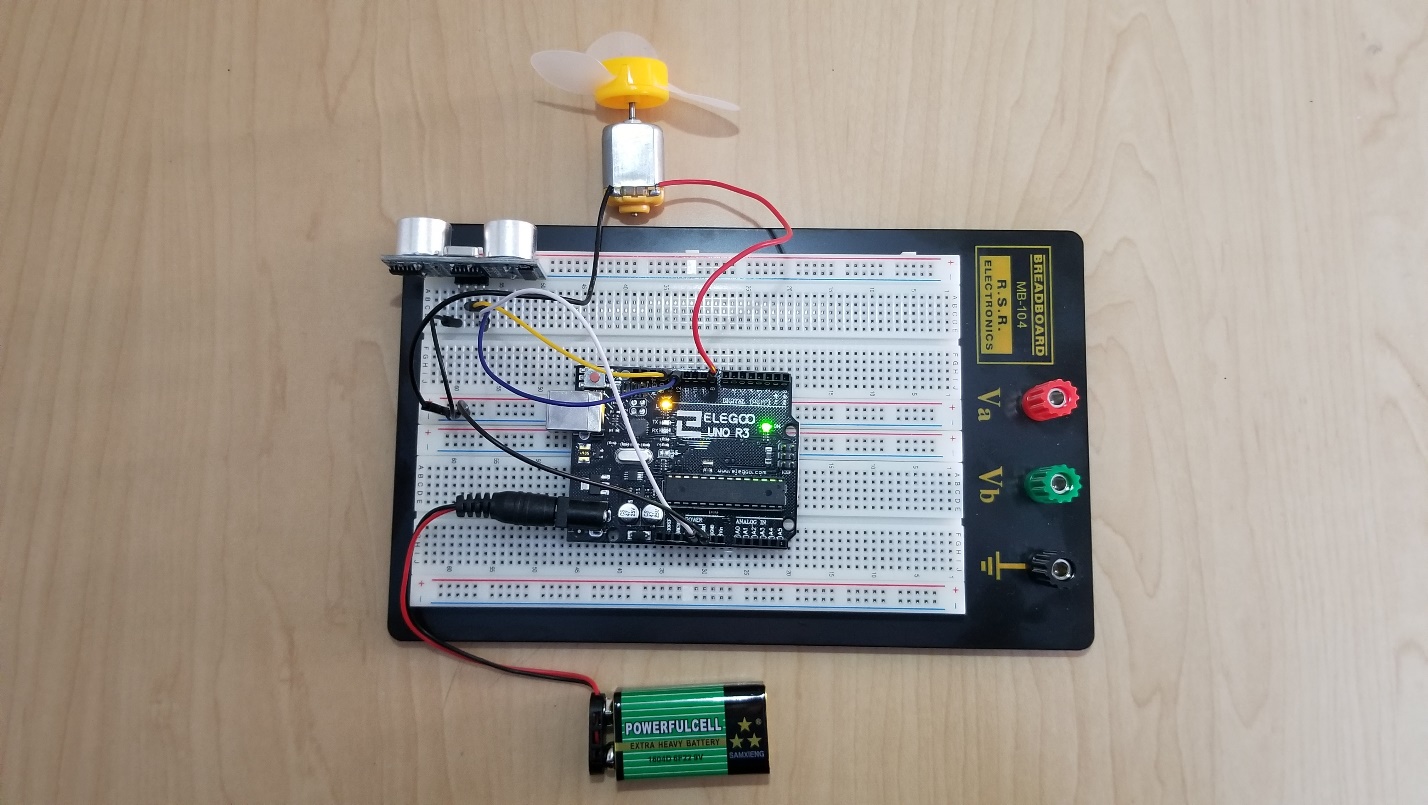


Figure 5: Completed Project

## 

## Additional Information

The following websites are resources for learning more about the Arduino. Use these resources to help you with this or future projects.

http://forum.arduino.cc/

The Arduino Forum is a place where you can browse frequently asked questions or ask your own if you can’t find what you’re looking for.

https://hackr.io/tutorials/learn-arduino

Hackr.io features free and paid tutorials for learning how to use the Arduino.

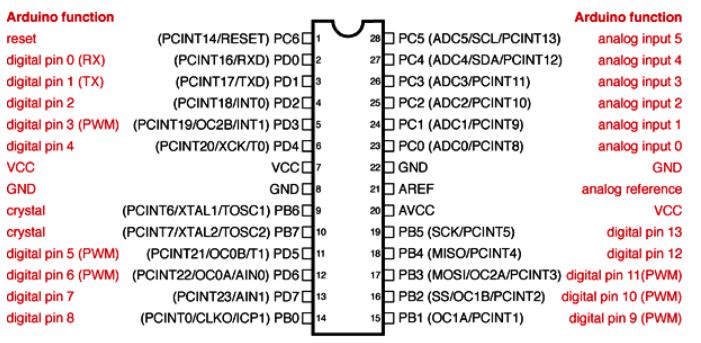
## Common Issues

If you are having issues with your project, try checking the following things to make sure your problem doesn’t have a simple fix. If your problem persists, try checking the Arduino forum provided in Additional Information to see if there is solution there.

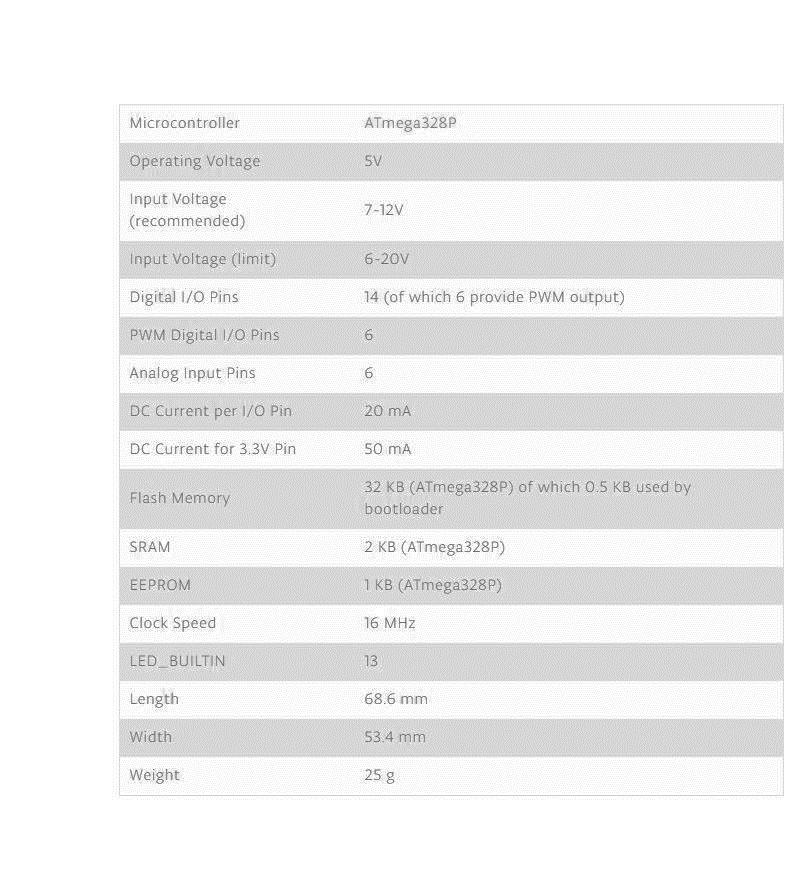
* Code won’t compile.
  + Go to File > Preferences > Show verbose output during: and check both boxes. Compile the code again and read the output on the bottom of the screen to see where the compiler is running into problems.
  + Common Coding Problems
    - Forgot or added an extra semi-colon.
    - Misnamed variables.
    - Incorrect Capitalization or formatting.
* Code won’t upload to the Arduino.
  + Check to see if the green status light is lit to indicate the Arduino is connected to the computer. If not, check to make sure both sides of the USB adapter are firmly attached.
  + Check to make sure you have selected the correct port under Tools > Port.
* Sensor is only reading 0 distance.
  + Check all wires to make sure they are secure.
  + Check to make sure you have wired the sensor’s pins to the same pins you indicated in the code.
* Motor turns on, but won’t turn off.
  + Check to make sure you have wired the sensor and the motor to the correct pins.

# Appendix A – Specifications for Arduino Uno

## Pin Layout

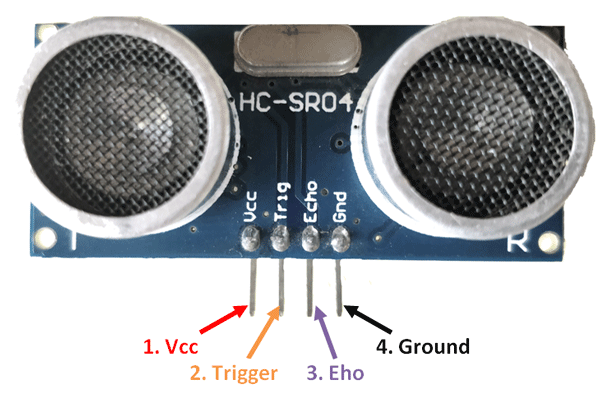


## Technical Specifications

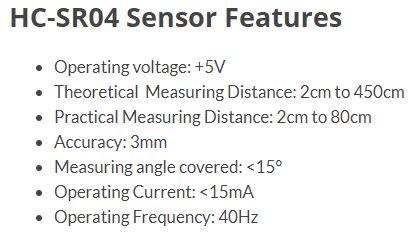


# Appendix B – Specifications for Ultrasonic Sensor

## Pin Layout



## Technical Specifications



# Appendix C – Software

## Complete Code

# 

# References

## Intro

Figure 1:

*Arduino Uno Rev3*, store.arduino.cc/usa/arduino-uno-rev3.

## Key

Robot:

j4p4n. “Standing Robot.” *Openclipart*, 8 Mar. 2018,

openclipart.org/detail/297959/standing-robot.

Danger:

Enolynn. “Danger Panel.” *Openclipart*, 29 Jan. 2015,

openclipart.org/detail/213291/danger-panel.

## Instructions

Figure 4:

“9V Snap Connector.” *SparkFun Electronics*, www.sparkfun.com/products/91.

## Appendix A

Pin Layout:

“PinMapping168.” *Arduino*, www.arduino.cc/en/Hacking/PinMapping168.

Technical Specifications:

*Arduino Uno Rev3*, store.arduino.cc/usa/arduino-uno-rev3.

## Appendix B

Pin Layout and Technical Specifications:

“HC-SR04 Ultrasonic Sensor.” *Components101*, components101.com/ultrasonic-sensor-

working-pinout-datasheet.

Technical Specifications:

*Arduino Uno Rev3*, store.arduino.cc/usa/arduino-uno-rev3.

## Conclusion and Appendix C

## All Images in these sections are original to this document.

## Figure 2 was made using TINKERCAD.