

Robots I - Week 6

Ultrasonic Sensing

How it works?



The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. The transmitter 'T' sends an ultrasonic waveform that the receiver 'R' tries to receive. The round trip time of the signal (the amount of time that takes for the signal to be sent, bounce off of something, and then be received) determines the distance from the sensor to the object.

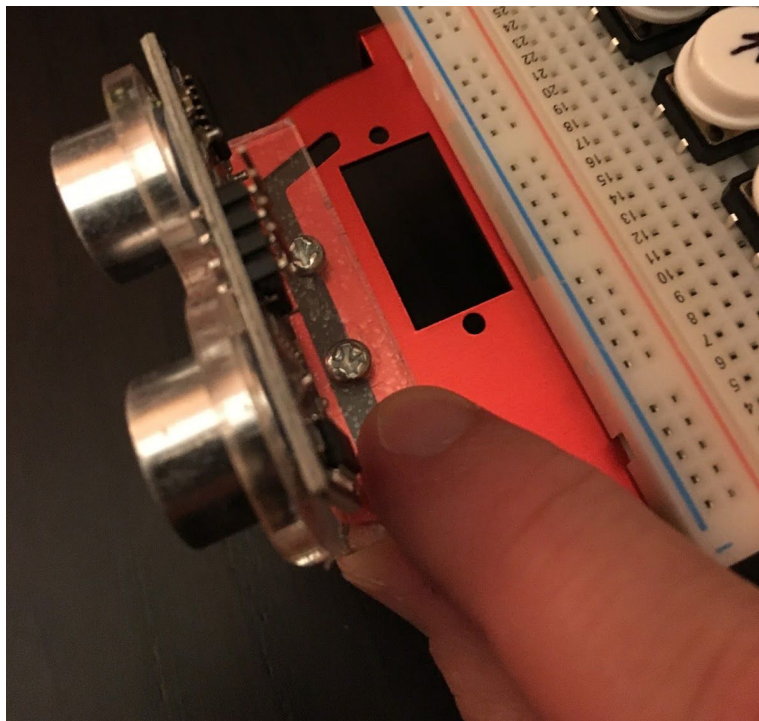
Ultrasonic sensors are used in a number of applications from backup sensors in cars, to fish finders, and our robot. They provide a great way to know when the robot is about to crash into something.

Want to learn more? Watch this quick video:

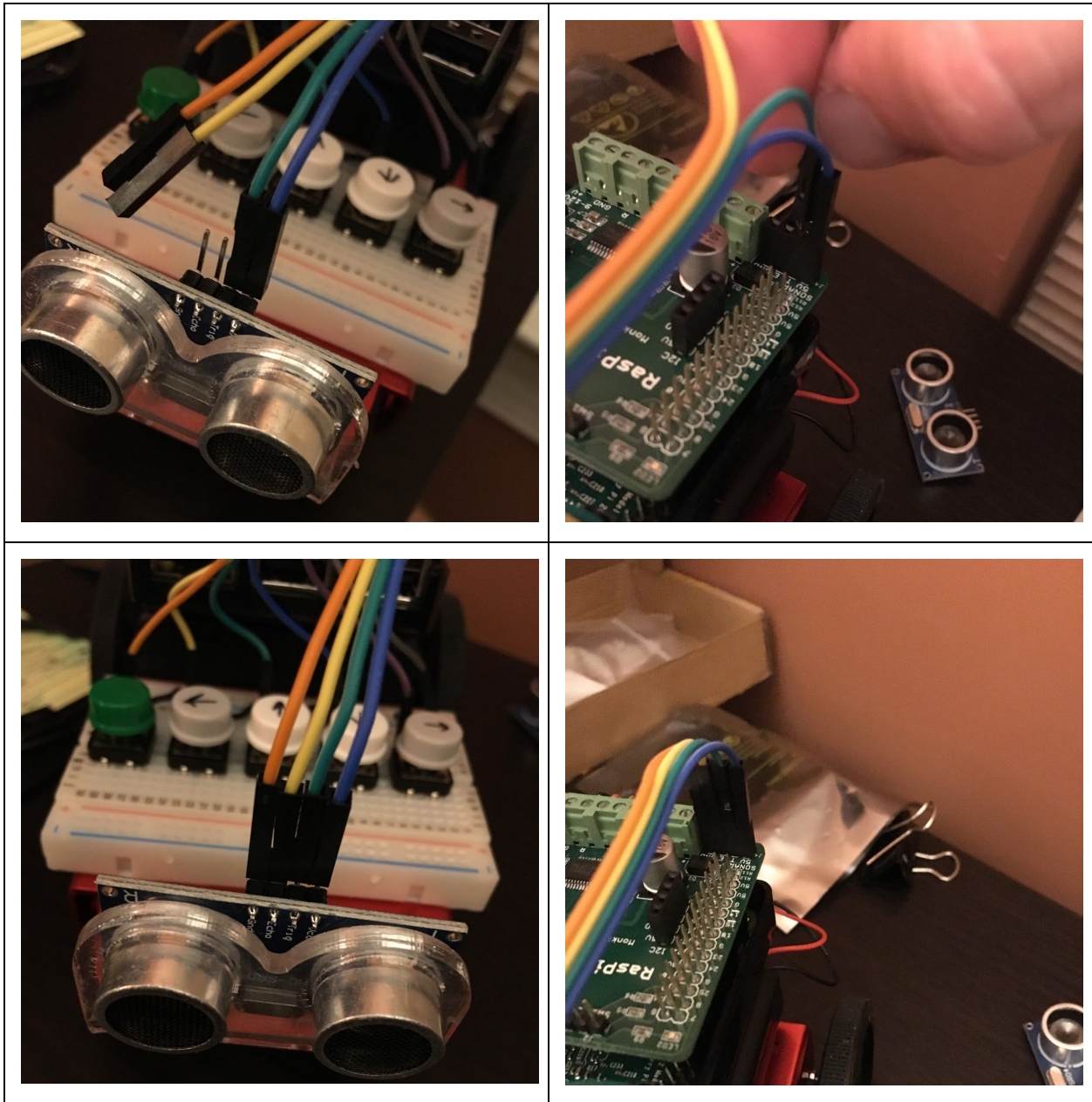
<https://www.youtube.com/watch?v=gejVi4OSJCY>

Wiring

In order to get our rangefinder installed we'll need to make a few changes to our robot. First, remove the battery and prepare the rangefinder and mounting bracket. Then mount to the robot as pictured below.



For wiring you will need to use four female to male jumper wires.



Basic Code

Our HC-SR04 measures distance in centimeters. Here is a basic code example that prints the measured distance.

```
from rrb3 import *  
import time
```

```

rr = RRB3(6,6)
try:
    while True:
        distance = rr.get_distance()
        print(distance)
        time.sleep(0.2)
finally:
    print("Exiting")
    rr.cleanup()

```

Put two robots next to each other and see if their values are the same. How far can the rangefinder detect distances? How close?

Distance Overrides

We want to integrate our rangefinder code into our keyboard control code from last week. This way we can AUTOMATICALLY stop if the robot is close to striking something. Unfortunately, our code has **blocking function call** this means that the robot will be moving while the code is waiting for user input from the keyboard. So, it is not possible to measure distance and stop the robot while the robot is waiting for keyboard input. This calls for a **thread**. Threads are a great way to do more than one thing at once. While our code is waiting for keyboard input, our code is also measuring distance! Let's take a look at the combined end product.

```

import os
import sys
from time import *
from rrb3 import *
import threading

FORWARD = 56
LEFT = 54
RIGHT = 52
REVERSE = 50
STOP = 53

os.system("stty raw -echo")

robot = RRB3(6,6)

def worker():
    while True:
        if robot.get_distance() < 10:
            robot.stop()
            time.sleep(0.2)

```

```
t = threading.Thread(target=worker)
t.start()
```

```
os.system("stty raw -echo")
while True:
    r = ord(sys.stdin.read(1))
    print r
    if r == FORWARD:
        robot.forward()
    elif r == LEFT:
        robot.left()
    elif r == RIGHT:
        robot.right()
    elif r == REVERSE:
        robot.reverse()
    else:
        robot.stop()
```