EET 2812

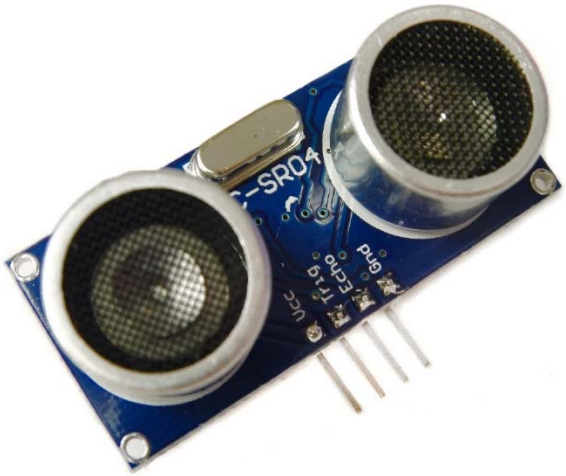
Activity 10

Single Board Computer (RPi)

Ultrasonic

Cuyahoga Community College

Youth Technology Academy



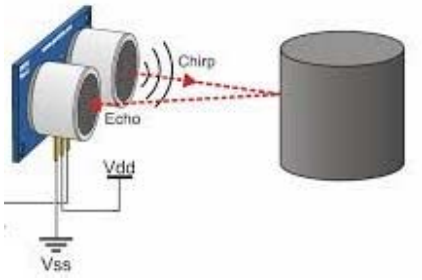
ACTIVITY

RPi Sensors Programming - Distance

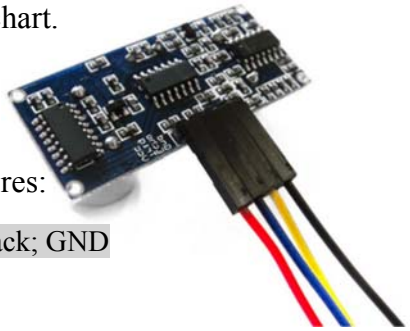
Overview: Connect, program and use an ultrasonic sensor to measure distance.

Vocabulary: RPi, GPIO Pins, Ultrasonic sensor (HC-SR04), resistor, breadboard, positive and negative rails, ground, function library.

One side of the HC-SR04 (Ultrasonic sensor) sends out a "chirp" and the other captures the echo that bounces off any object that is in front of the sensor. We will set up the Ultrasonic (HC-SR04) to send out a pulse and then calculate the time it takes for the ECHO to return. The program will then convert the time is takes for the ECHO to return into the distance travelled by the sound.



We need to use 2 resistors in the Ultrasonic (HC-SR04) circuit. To determine the resistance of a resistor we use the following chart.



We will use two 1k Ω resistors.

To set up the circuit we will use the following wires:

Red; Vcc Blue; TRIG Yellow; ECHO Black; GND

The circuit for setting up the RPi and the HC-SR04 is as follows:

Plug the RED Vcc wire into the positive rail on your breadboard.

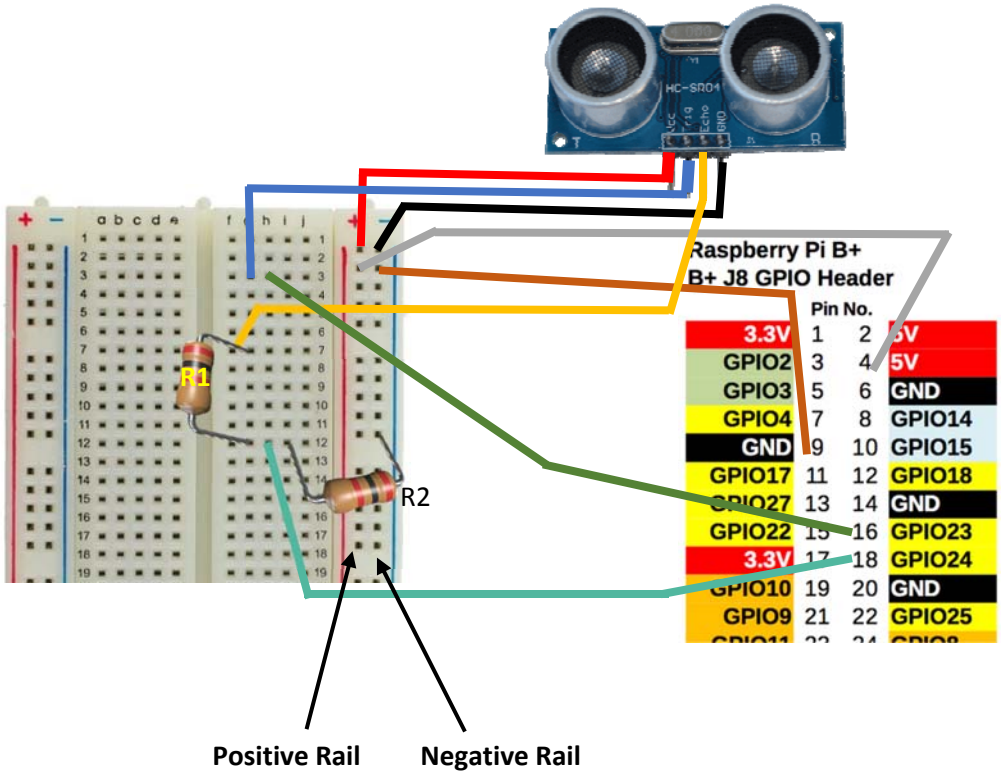
Plug the Black GND wire into the negative (GND) rail.

Plug 5V pin 4 to the positive rail under the RED Vcc wire.

Plug TRIG into a blank rail and plug that rail into Pin 16 using another wire.

Plug ECHO into a blank rail. Link it to another blank rail using R1.

Link R1 to the GND rail using R2 – leave a space between them.



Plug a wire into the space between R1 and R2 and connect it to your RPi at pin 18.

Programming the HC-SR04 using the RPi and Python

Connect and boot up you RPi. Open a terminal window and change to your my_python folder.

$cd my_python

Create a new program called ultrasonic using nano.

$sudo nano ultrasonic.py

Type in the following into the program.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

TRIG = 23
ECHO = 24

print "Distance Measurement In Progress"

GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG, False)
print "Waiting For Sensor To Settle"
time.sleep(2)

GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)

while GPIO.input(ECHO)==0:
  pulse_start = time.time()

while GPIO.input(ECHO)==1:
  pulse_end = time.time()

pulse_duration = pulse_end - pulse_start

distance = pulse_duration * 17150

distance = round(distance, 2)

print "Distance:",distance,"cm"

GPIO.cleanup()
```

Import the libraries we will use.

Set the mode of the pin numbers we will use.

Define variables denoting where the HC-SR04 is plugged into the board.

Set up the HC-SR04.

Calculate the distance traveled by the sound wave emitted and received by the sensor.

Print the distance to the screen.

"Clean up" the board so we don't have errors when we use it again.

Save the program pressing CTRL O and ENTER. Then exit nano by pressing CTRL X.

Run the program as a superuser: **$sudo python ultrasonic.py** **To exit the program press CTRL Z**