CPEN 291

Computer Engineering Design Studio I

Lab 5: Internet-enabled Range Finding (Arduino sees the world around it like a bat, and sends tweets about it)

ECE – UBC

2016 W2

©Farshid Agharebparast

February 2017

Electrical and Computer Engineering

University of British Columbia

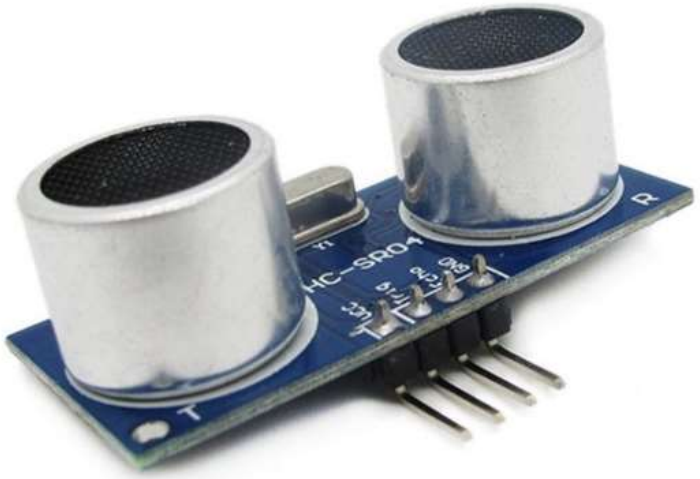
Objective

- This lab is
  - to give Arduino a bat-like eye to sense its surrounding using an ultrasonic range finder,
  - to send the sensor info via the Internet using Twitter.
- Safety review
  - Please do review safety concerns
  - A safety review quiz was done in previous lab
- We will use the ultrasonic range finder module HC-SR04.
- We consider the fact that Arduino can communicate via the Internet through the computer it is connected to as the simplest method to have an Internet-enabled Arduino.
- The remaining teams will also have an introduction to Pi 3.

Ultrasonic Range Finding Sensor

- We will use the popular and inexpensive HC-SR04 as an ultrasonic range finder.

HC-SR04:



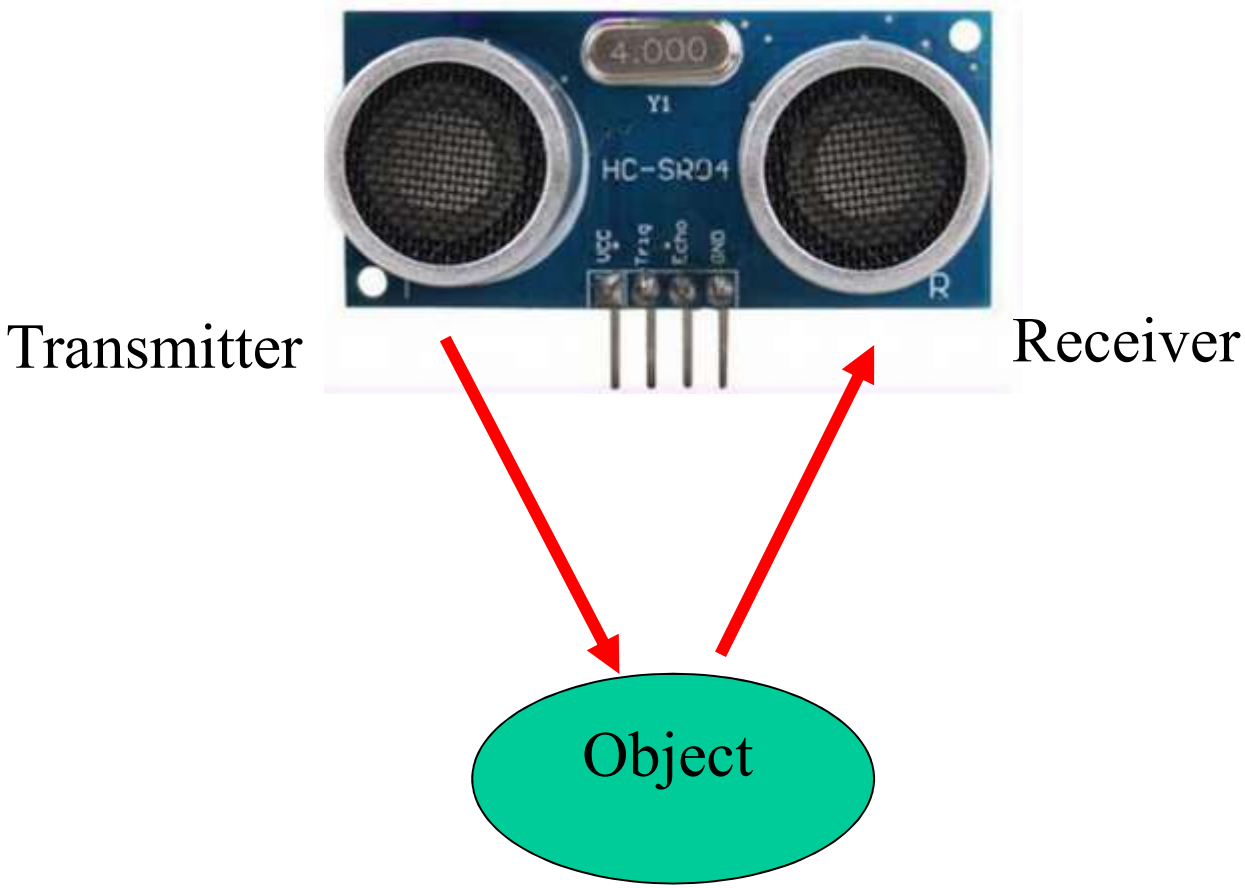
Specifications

- Power supply: 5V DC
- Quiescent current: <2mA
- Effectual angle: <15°
- Ranging distance: 2cm – 500 cm
- Resolution: 1 cm
- Ultrasonic Frequency: 40k Hz

VCC = 5V
Tri = trigger input for Sensor
Echo = Echo output from Sensor
GND = ground

HC-SR04

- HC-SR04 can be used to measure the distance from an object by measuring the time it takes for a transmitted signal to reflect from that object and be received.

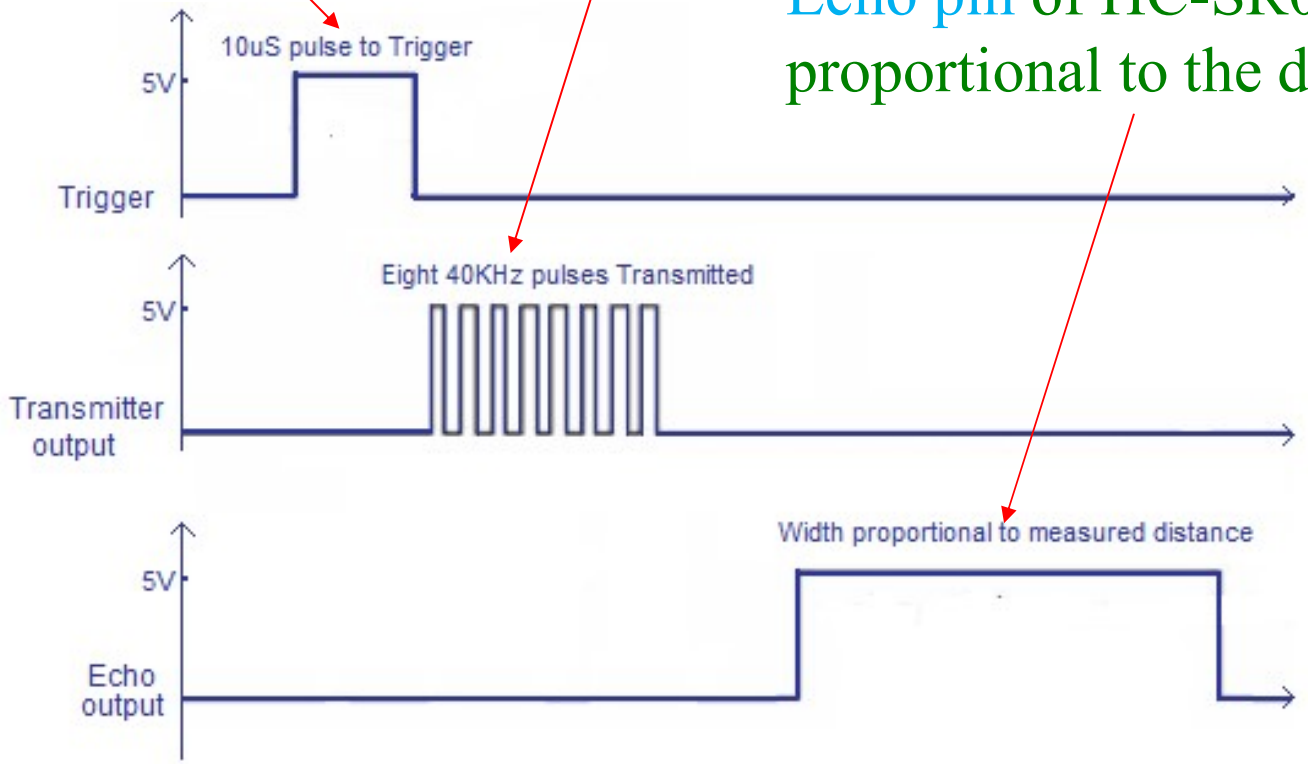


HC-SR04 Timing Chart

A 10us pulse is applied to the Trig pin of HC-SR04 as a trigger signal

HC-SR04 then transmits 8 pulses (40kHz, internally/automatically created).

The width of the pulse on the Echo pin of HC-SR04 will be proportional to the distance.



HC-SR04 Timing

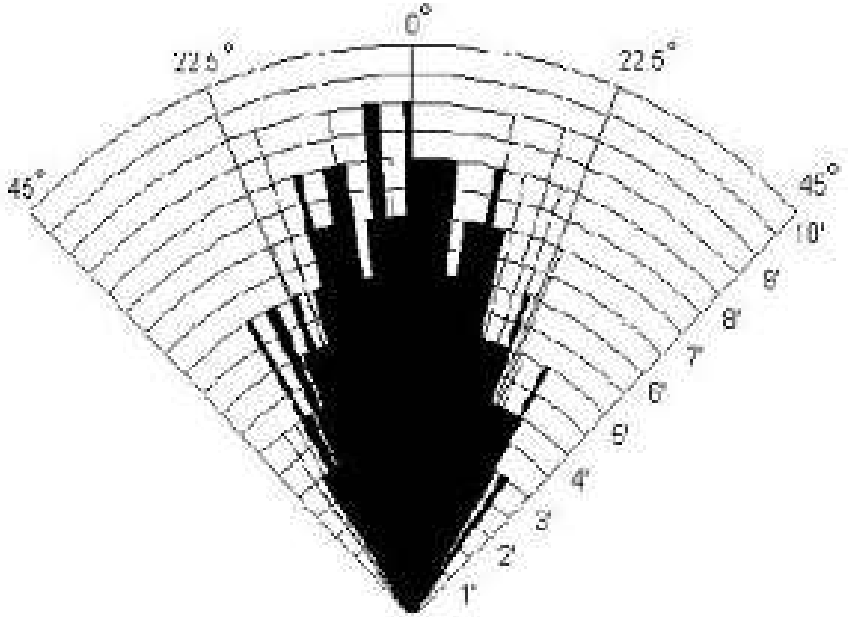
- HC-SR04 is told when to start to transmit/measure by a trigger signal.
  - The sensor may be triggered by a 10uS pulse on its Trig input pin.
- When triggered, it transmit a burst of 8 ultrasonic pulses, and then it raises its Echo output pin.
  - HC-SR04 lowers the Echo signal whenever it receives the echo signal (echoed from the object).
- The width of the Echo signal is then used to measure distance (Assuming speed of sound = 343 m/s => It takes 29 us per cm, and to account for the round trip 2*29 = 58):

  $$\text{Distance in cm} = \text{Echo pulse width in } \mu\text{s} / 58$$

- A new signal can be sent only once the echo fully fades. This is called the total cycle period, and according to the datasheet should not be less than 50 ms.

HC-SR04

- We should be aware of the range of angles for which HC-SR04 has the best performance.
- The following diagram shows the practical test of performance from the datasheet.



Practical test of performance,

Best in 30 degree angle

Improving Measurement Precision

1. Integer calculations are faster but less accurate. Although using floating point calculations is a burden on the Arduino CPU, but it may be justified here to improve measurement precision.

2. The formula provided for distance is actually dependent on a number of factors that affect the speed of sound. For example, we can improve our calculations if we know the temperature. Use an LM35 temperature sensor to measure the temperature and to measure the actual speed of sound:

   $$\text{Speed of sound} = 331.5 + (0.6 * \text{Temperature in } ^{o}C)$$

   Then replace the actual speed of sound in the formula for calculating distance.

Twitter

- We will use a combination of the following software to make our Arduino Internet-enabled.
  - Processing,
  - Twitter java library, and
  - Twitter
- Pease see the lab info document (html) for more info.

References

- Arduino Reference:
  http://arduino.cc/en/Reference/HomePage

- Processing download website:
  https://processing.org/download/
  Processing Tutorial: https://processing.org/tutorials/

- Twitter: https://twitter.com/?lang=en

- Twitter apps: https://apps.twitter.com/

- Datasheets:
  - See the datasheets posted on Connect