Ultrasonic Range Finder 3pins

The Ping))) is an ultrasonic range finder from Parallax. It detects the distance of the closest object in front of the sensor (from 2 cm up to 3m). It works by sending out a burst of ultrasound and listening for the echo when it bounces off of an object. The Arduino board sends a short pulse to trigger the detection, then listens for a pulse on the same pin using the pulseIn() function. The duration of this second pulse is equal to the time taken by the ultrasound to travel to the object and back to the sensor. Using the speed of sound, this time can be converted to distance.
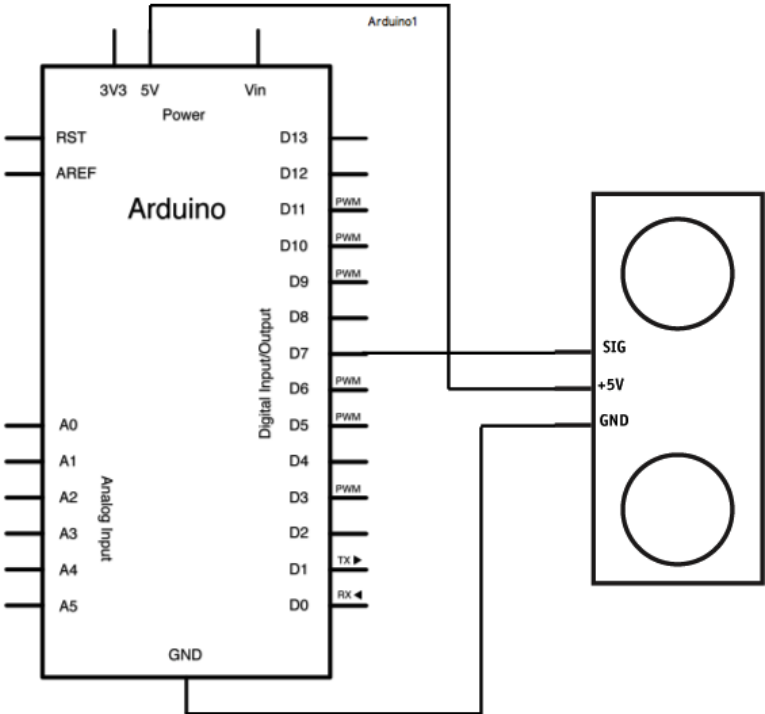
HARDWARE REQUIRED

Arduino Board
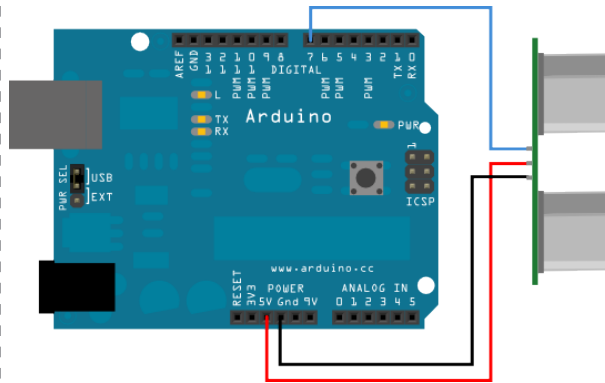(1) Ping Ultrasonic Range Finder
hook-up wire

CIRCUIT

The 5V pin of the PING))) is connected to the 5V pin on the Arduino, the GND pin is connected to the GND pin, and the SIG (signal) pin is connected to digital pin 7 on the Arduino.

SCHEMATIC

IMAGE

/* Ping))) Sensor

  This sketch reads a PING))) ultrasonic
rangefinder and returns the
  distance to the closest object in range. To do
this, it sends a pulse
  to the sensor to initiate a reading, then listens
for a pulse
  to return. The length of the returning pulse is
proportional to
  the distance of the object from the sensor.

  The circuit:
   * +V connection of the PING))) attached to +5V
   * GND connection of the PING))) attached to
ground
   * SIG connection of the PING))) attached to
digital pin 7

  http://www.arduino.cc/en/Tutorial/Ping

  created 3 Nov 2008
  by David A. Mellis
  modified 30 Aug 2011
  by Tom Igoe

  This example code is in the public domain.

*/

CODE

```
// this constant won't change.  It's the pin number of the sensor's output:
const int pingPin = 7;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop()
{
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters: long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);
}

long microsecondsToInches(long microseconds)
{
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second).  This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```