Winkleink - box of wires

A place for random stuff with a bit of a focus on Raspberry Pi and Arduino and my effort to figure them out

Tuesday, May 15, 2012

Arduino - HC-SR04 ultrasonic distance sensor

Last Christmas as part of my stocking fillers I got an HC-SR04 ultrasonic distance sensor. Like most of my electronic bits it's a cheap generic device from eBay. It's a small sensor that is supposed to do the same thing as the Ping sensor but for less money.

I finally unwrapped it and found a library for Arduino IDE 1.0 at HERE. There is even some sample code to read the HC-SR04 and display the results on an LCD display.

Wiring is really simple
VCC - 5V
GND - GND

Trig - Trigger Pin you define in the code
Echo - Echo Pin defined in the code

Since I don't have an LCD display (yet) I modified the code to use the Serial Monitor as the output.

Code Below:

```c
#include "Ultrasonic.h"

int TriggerP = 13; // Trigger Pin for Sensor
int EchoP = 12; // Echo Pin for Sensor

Ultrasonic ultrasonic(TriggerP, EchoP);

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.print("Distance: ");
  Serial.print(u.ltrasonic.Ranging(CM)); // Get Range in Centimetres
  Serial.println(" Cm.");
  delay(1000);
  Serial.println("..."); // Next line.
}
```

This worked great up to about 16cm. But beyond that it started to give me some random numbers.
greater than 4000, so I'd expect not very robust.

I continued my search and found the following code that doesn't use any library, and this code even does the Serial Monitor as the output so no modification needed.

```c
/*
 * HC-SR04 Ping distance sensor
 * VCC to arduino 5v GND to arduino GND
 * Echo to Arduino pin 13 Trig to Arduino pin 12
 * More info at: http://goo.gl/k39G6l
 */
#define trigPin 12
#define echoPin 13

void setup() {
    Serial.begin (9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop() {
    int duration, distance;
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;
    if (distance >= 200 || distance <= 0) {
        Serial.println("Out of range");
    } else {
        Serial.print(distance);
        Serial.println(" cm");
    }
    delay(500);
}
```

This code performed better, working out to about 24cm.

While doing my research on using the HC-SR04 I did find a post (somewhere...) stating that when powering from USB the 5v line can be a little bit off so this might be part of my problem.

But, using either method I have a sensor that is good up to 15cm which could be good enough for a small autonomous robot as if I'm within 15 cms I will need to look around for another direction.

NOTE: If you are using a version of the Arduino IDE before 1.0 then at the following LINK you can get the relevant library. It looks like it's the same as the 1.0 library and has similar performance.

The hunt is on to see if there is a way of getting a reading beyond 24cm that is reliable.

Loving the tinkering;

UPDATE:
I combined some of the code from the Library with the direct code and now it reliably(ish) without detailed testing will work to 70cm.

The modified code is below.

```c
/*
 * HC-SR04 Ping distance sensor
 * VCC to arduino 5v GND to arduino GND
 * Echo to Arduino pin 13 Trig to Arduino pin 12
 * More info at: http://goo.gl/k39G6l
 */
```


```c
#define trigPin 13
#define echoPin 12

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW); // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
  // delayMicroseconds(1000); // Removed this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (distance > 200 || distance < 0) {
    Serial.println("Out of range");
  } else {
    Serial.print(distance);
    Serial.print(" cm");
  }
  delay(500);
}

Once again, I love this tinkering.
```

Posted by Winkle Ink at 10:29 PM

---

Labels: arduino, distance sensor, example, hack, hc-sr04, hcsr04, howto, tinker, ultrasonic
Winkleink - box of wires: Arduino - HC-SR04 ultrasonic distance sensor

http://winkleink.blogspot.com/2012/05/arduino-hc-sr04-ultrasonic-distan...
Winkleink - box of wires: Arduino - HC-SR04 ultrasonic distance sensor

http://winkleink.blogspot.com/2012/05/arduino-hc-sr04-ultrasonic-distan...