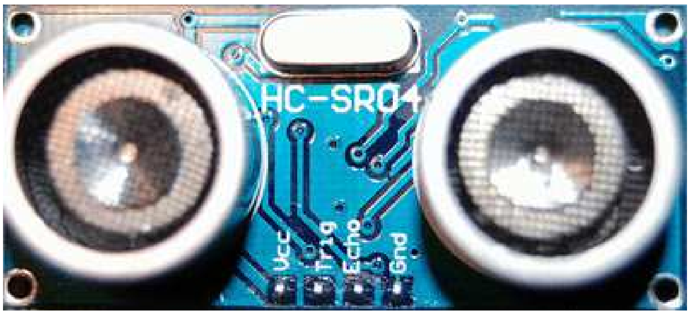
Emil's Projects

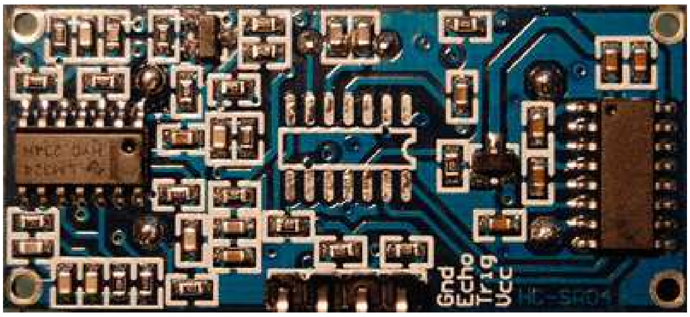
OpenHardware & OpenSource

Making a better HC-SR04 Echo Locator 22nd January 2014

I have bought several cheap echo location circuits from AliExpress and I'm planning to use them in one of my robotics projects.



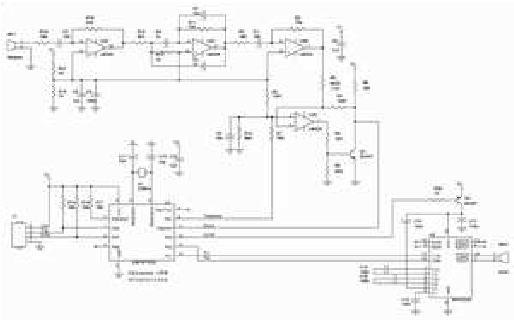
HC-SR04 Front View



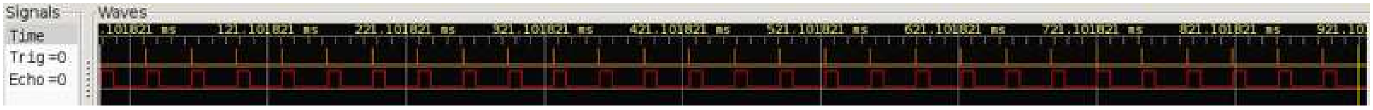
HC-SR04 Back View

They are very cheap especially if you buy a dozen you can get them bellow 2$ shipped.
Once they've arrived, to test them, I wrote a quick C test program for an AVR board which displays the distance in cm to an obstacle. The results were pretty accurate for as far as 3-4m distance for hard obstacles. Unfortunately when there are no obstacles in front of this detector or when the obstacles don't have hard surfaces then erroneous results are returned.

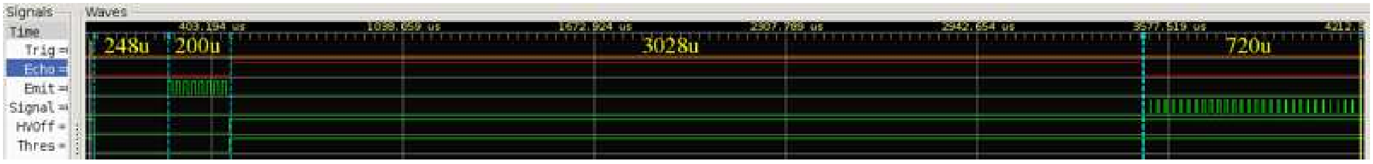
I wanted to understand why this was happening so the first step was to reverse the HC-SR04 module schematic



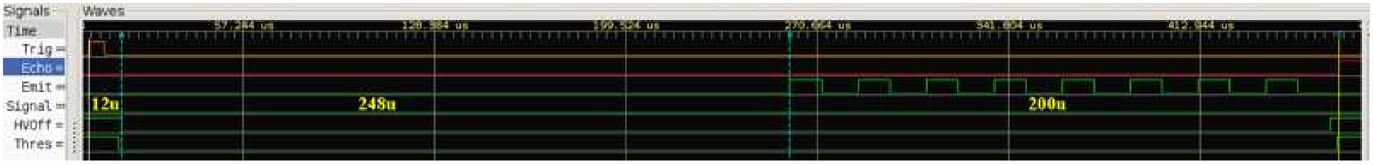
The circuit has 2 transducers one for emission and one for reception. To transmit the ultrasonic pulses a relatively high voltage is needed. A MAX232 (U3) is cleverly used to produce +/-12V (which are the normal USART voltages) from 5V. The transducer is connected between two outputs so it is in fact powered at 24V. Power is only applied to this circuit through Q2 some time before and during pulse emission because the internal switching charge-pump is noisy. When the circuit switches to receive mode the MAX232 power is cut off.
The receiver side uses LM324 which contains 4 OPAMPs. U2D is just a times 6 amplifier. U2C is a 40KHz pass band filter which is followed by another times 8 amplifier (U2B). The last OPAMP (U2A) is used together with Q1 as a hysteresis comparator. The receive and emit circuits are controlled by an EM78P153S chinese microcontroller running at 27MHz.
Bellow are some waveforms captured from the circuit. If you want to see them in detail I have saved all traces in VCD format here and you can use the free gtkwave viewer to analyse them.



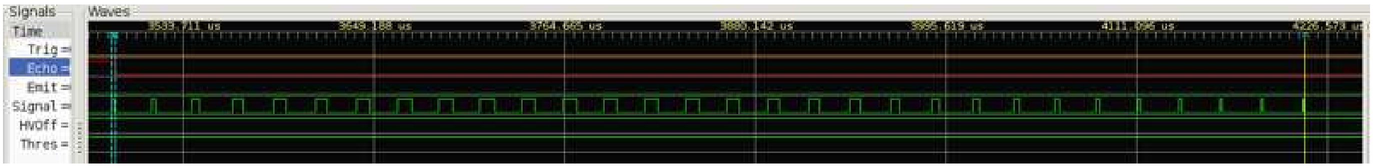
Regular pulses from a fixed hard surface



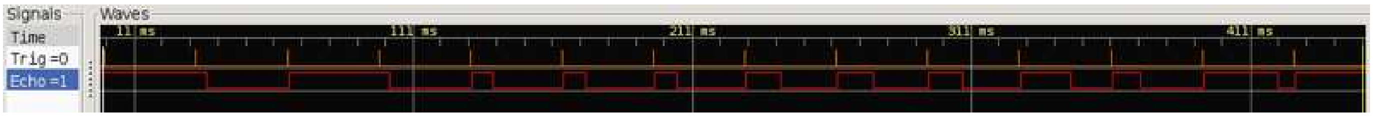
Typical waveforms for one echo detection



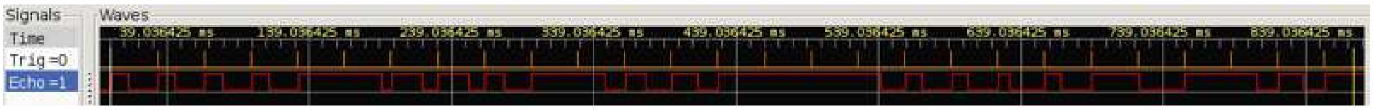
8 pulses at 40KHz are emitted



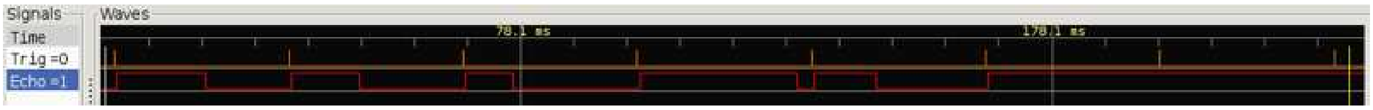
the echo is longer because of the filter response and multiple reflections



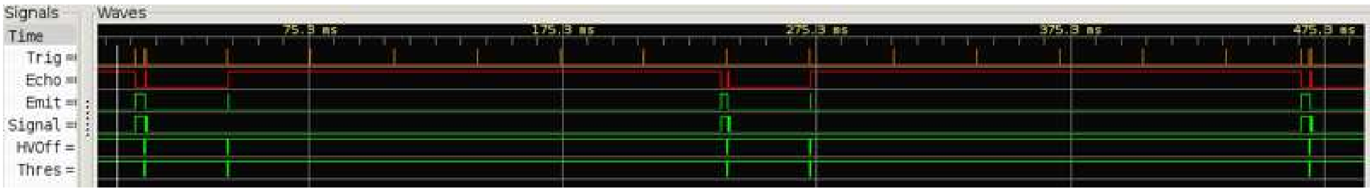
echo when the obstacle has soft surfaces - only some echoes are valid



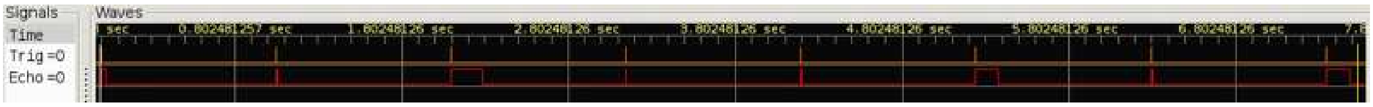
random echoes are returned when there is no obstacle in front



zoom in on random echoes



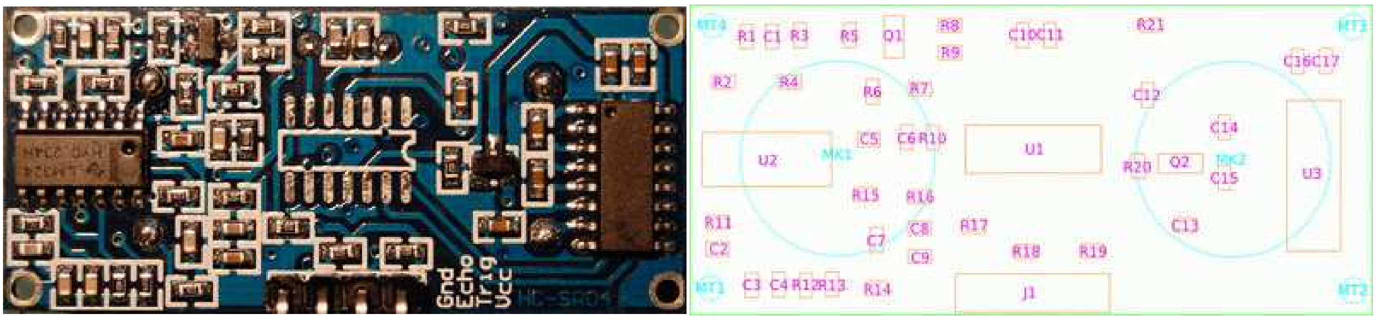
retrigger problem when no echo is received



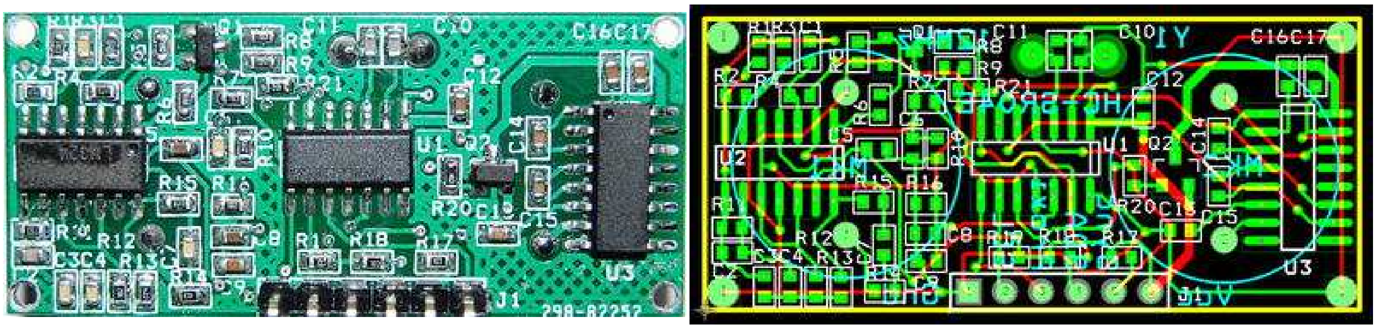
false readings even when triggers are 1s apart (all reflections have died off)

After the trigger input is raised the uP detects it and after some 10us powers on the MAX232. After 248us, time to ramp up the +/-12V of the charge-pump, the 8 pulses 40KHz train is produced and then the power is switched off. During this time the comparator threshold is also kept low to prevent any spurious signals to be detected in the receiver. The circuit then asserts the echo signal and waits for the echo to return. As soon as the first pulse is detected in the receiver the echo signal is deasserted and one can measure the width of the echo pulse to calculate the obstacle distance.
There are several problems with this circuit. First the uP uses polling to detect a return pulse. This can be observed because sometimes only the second pulse is detected. There is no attempt to check is that first pulse is part of a 40KHz train of pulses. A noisy environment will probably produce false pulses which will pass trough the analog 40KHz filter (which has a modest 6db/octave attenuation). The circuit biggest problem is the false echo pulses which are returned when there is no obstacle in front. These have a plausible and variable range and cannot be filtered out in software.
The processor is OTP so cannot be reprogrammed and there is no easy way to change it and hook another one in its place. The whole circuit is made to minimise the cost to extreme. The 27MHz crystal was unmarked and when I've measured it had almost 1/1000 deviation.

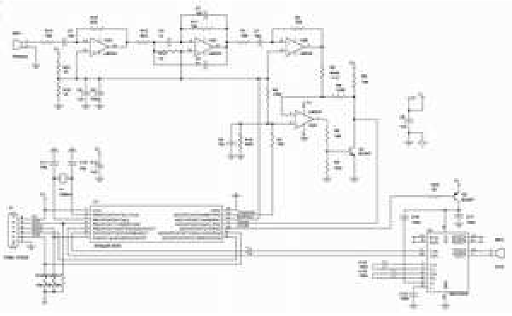
In the end I've decided to change the processor and the board so I can improve this design. The analog part is pretty decent so I will keep all the components except the uP and the crystal.



First I've marked exactly where each component was placed on the original PCB



Then I've designed my own PCB keeping the same component locations



I've used the AtTiny24 in a SOIC14 package with a 12MHz crystal. This is the new HC-SR04E Schematic (I've appended an 'E' to the module name), the Bill of Materials and the Gerber files or Gerber panel if you want to use the cheap ITeadStudio PCB manufacture and get 20pcs for a tenner. Soldering one of these new boards takes about 20 minutes. The way I do it is by placing the original and new PCB side by side and desoldering the components from one with a hot air rework station and soldering them back in the same position on the new board. In this way I don't have to pay attention to component's values. Also notice that there are now 6 pins instead of the 4 on the old connector.

The advantages of this new schematic are:
* The SPI interface of the AtTiny is available at the 6 pin connector and it can be programmed in place as many times you want.
* There are 3 signal pins available instead of 2 and if you program the Reset even 4 signal pins but you'll lose the in-circuit programming.
* The received analog signal also connects to the AtTiny internal comparator so a timer with capture can be used to store all returned pulses and filter them by position and gettind rid of any potential noise.
* The I2C interface is available so multiple modules can be wired together and have different slave addresses.

I have several plans with these new boards. First I'll just re-implement the original functionality with one Trigger and one Echo output while correcting the false echoes problem. Then I'll make an I2C version where the modules triggers itself and computes the distance in 'mm' locally and stores it as 2 bytes. Another master processor asynchronously reads this data only when needed. One of the available pins will signal a proximity condition when the distance decreases under a certain value. Another iteration will use an open drain pin to sync the emission of several modules to be able to use multiple sensors on the same robot without having to wait for each one to complete a measurement. AVR software to follow when I'll have some more time available.

Tags: avr, hc-sr04, ultrasound.

Comments On This Entry

Tom Submitted at 10:06:28 on 16 March 2014

Good work! What is the schematic/PCB package that you use? Thanks, Tom

Emil Submitted at 13:57:15 on 16 March 2014

OrCAD/Layout v15.7

Mike Submitted at 21:22:17 on 16 March 2014

And you will be selling assembled and tested versions when?

Emil Submitted at 22:05:24 on 16 March 2014

I do not plan to sell anything. The design is open source so if someone wants to pick it up when it's finished they are welcomed.

John Submitted at 01:57:41 on 17 March 2014

Great work. DO you happen to know what the physical difference is between the Tx and Rx transducers?

Emil Submitted at 16:06:16 on 17 March 2014

The transducers have the same physical dimentions, they look the same but they are marked differently. I haven't tried to swap them though. A few years ago I remember designing an echo locator with the same transducer used for both emit and receive but the HV was produced with a transformer and it was as high as 400V. The gain on the receive side was also much higher.

Geebles Submitted at 11:29:49 on 18 March 2014

Have you tested the range/sensitivity to soft objects? I'd be interested to know this for use over grass! Could higher power be used by using so other source to generate a higher voltage? to increase range/clarity of pulses? Looking forward to firmware to come out as i'd be interested in trying this out!

Emil Submitted at 16:05:34 on 18 March 2014

I have not tested the original design sensitivity rigorously. Chairs made of fabric were detected to about 1m. The design I was describing in the comment above was for a 3m diameter quadcopter with one transducer on each arm. On grass the precision was in the cm range and it was working up to 1.5m. I very much doubt that this $2 sensor can sense the grass above 50cm.

Chris Submitted at 17:28:40 on 18 March 2014

I can't help but wonder: Why is a two-transducer design less expensive to manufacture than a single-transducer circuit? So many inexpensive sensor modules use two, so there must be a very good reason. I'm just curious to learn. Another question: If the new AtTiny software stores all returned pulses, an exciting new possibility opens up. Multiple object detection! If there at two objects - let's say 1m and 2m - we should receive sixteen pulses in two groups of 8. It could even be useful to know the amplitude of each reflection. The return value would need to be a more complex data structure, perhaps one byte for number of reflections, two bytes per reflection for distance and one byte per reflection for amplitude. It may even be possible to estimate the size of a single object. A great post, you've really got my mind going now!

Emil Submitted at 18:10:46 on 18 March 2014

A two transducer design is less expensive because you don't need to use an analog switch. This switch must block high voltage on the receive side while also having low impedance so that it won't attenuate the received pulses. I remember in my design I've used PIN diodes to protect the receiver side but some of the emission power was dissipated on this input protection circuit.

Domen Submitted at 15:47:18 on 19 March 2014

Hi! Nice project! Thanks for putting it out there, that I can buy this sensor for this cheap :D Just orderd 10 of then from Aliexpres for $12.5. Also, do you have a time frame for the firmware?

Emil Submitted at 21:09:39 on 19 March 2014

The time frame is: whenever I have time. I'm doing this hobby in my spare time and that is a limited resource. If you're in a hurry you can always program it yourself. It's not a hard program to write.

miceuz Submitted at 21:19:43 on 21 March 2014

great stuff, thanks for sharing. Now probably I will desolder piezo elements from the board I have and breadboard it around.

Yogesh tiwade Submitted at 17:58:05 on 6 April 2014

HC-SR04 it is a great kit of ultrasonic tx & rx using 8051..... It a very nice ideal to make a mini project & also used it future for maga project...

RiGonz Submitted at 07:41:01 on 17 April 2014

Hi, Emil! Quite impressed by your work. Though I'm just an Arduino amateur I've got a question which you seem quite qualified to answer (and as far as I have searched internet and I can judge, probably the best one). I am trying to use two HC-SR04, so that I can send the ping from one, and read the travelling time on the other. (The reason is that I expect accuracy to be increased, as well as,maybe, the screening range). Then: do you think this can be achieved just from the software side? (I do not even think of desoldering the "drums"!)

Emil Submitted at 11:12:25 on 17 April 2014

With current configuration you can only send the pulses from both at the same time. These will not be in sync because each module has its own clock and the detection of your start impulse is done with pooling. Nevertheless the time shift will not be greater that one impulse period so your error will be minimal. The biggest problem are the false impulses detected when there is no obstacle (at least this happens with the modules I have). This is what I'm trying to improve with my new design. I plan to use 6 of these (all synced together) on a balanced robot and some additional IR sensors.

Jim Remington Submitted at 02:59:54 on 18 April 2014

Nice work! I also had the idea to improve the circuitry, and am glad you were ahead. However, I'm puzzled by the function of U2C, which is appears to be a filter. I used LTSpice to simulate the basic U2C module using the component values you specify on the schematic, and from the AC sweep analysis, it appears that the overall gain peaks at 20 kHz, at -28 dB and drops off from there to -40 dB 12 and 36 kHz. That doesn't make much sense to me, so I wonder if you are certain of the capacitor values associated with the feedback path of U2C. You can email me for the spice file if you like.

Jim Remington Submitted at 03:13:06 on 18 April 2014

I can get the filter to peak at about 40 kHz if C3 is 220 pF, but the gain is still quite low, at -26 dB.

Emil Submitted at 13:25:15 on 26 April 2014

Hi Jim. Thanks for pointing this out. I've used an RLC meter measure all components and I don't think I've mixed anything up (but it's still possible). When I'll build my next module I'll measure the caps again. If you are right then I'll redesign the filter to be centered at 40KHz. I already have the schematic in OrCAD so I'll use its own Spice simulation.

Add A Comment

Your Name

Your EMail

Your Comment

Submit Query

Your submission will be ignored if any field is left blank or if you include links. Your email address will not be displayed.

Archive

- 2014
  - January (1)
- 2013
  - March (1)
  - April (1)
- 2012
  - June (2)
  - December (1)
- 2011
  - October (1)
- 2010
  - April (1)
- 2009
  - June (1)
- 2006
  - March (1)
  - December (1)

- 2005
  - August (1)

Tags

- avr (3)
- blheli (1)
- c compiler (1)
- code obfuscation (1)
- disassembler (1)
- esc (1)
- gazelle (1)
- gps (1)
- graphic lcd (1)
- hc-sr04 (1)
- i2c (1)
- ida (1)
- illumination (1)
- jtag (1)
- kint (1)
- led (1)
- linux (1)
- pcb (1)
- reverse engineering (3)
- sad (1)
- silabs (1)
- system76 (1)
- tektronix (1)
- tla (1)
- ublox (1)
- ultrasound (1)
- uv (1)
- vcd (1)
- wifi (1)
- xpdf (1)