Using 3 modified hc-sr04 to triangulate the position of one or more objects

ardcp 1 September 6, 2021, 1:20pm

I have done some tests using three modified hc-sr04 in order to check if it is possible synchronize the receivers accurately enough to determine if an object is to the left, near center, or to the right using time differences.

Using only one hc-sr04 as active transmitter/receiver and two acting as receivers, it should be possible to calculate the position of one or more objects by triangulation.
The two receivers are placed 16 cm apart and angled toward the center with a common focus about 20 - 25 centimeters in front. The T/R is mounted vertically to have both the transmitter and receiver at the center.

For processing the signal out of the amplifier, the onboard mcu is disabled when acting as receiver only.
To replace the onboard mcu, at2313a/at4323a are used with the Attinycore as core.
At2313a requires use of a non buffered simple hardware serial out to be able to print an int.

All the hc-sr04 are synchronized within a few microseconds, using the rising edge of the T/R echo pin to start the timer1 of the two receivers as well as the T/R using pinchangeinterrupt.
Even using a 16Mhz x-talclock divided by 8 takes more than 32 ms before the first overflow, sufficient for the distances in question.

On the two receivers, the onbord mcu has been disabled, unsoldering pin 9, threshold, and pin 10, signal in, order to avoid any interference from the mcu.
The at2313a/at4323a of the T/R is connected to the threshold and signal of the onboard mcu.

On the versions of the hc-sr04 used(versions with 2 transistors), the design flaw of the band pass filter, peak at around 12 Khz, has been modified to a peak closer to 40 Khz, resulting in a much better sensitivity.

The analog comparator, pin AIN0, is connected directly to the signal output of the hc-sr04 (was pin 10 of the onboard mcu).
Every time an echo pulse train starts, the analog comparator interrupt function checks if the interrupt time stamp should be saved, at the moment up to 4 timestamps.
The resolution of the timestamps is 1 usec when sent to the controller.

The controller, at the moment an arduino uno is used, only needs to send a trigger signal to the T/R, then wait around 30ms before collecting the data by sending a signal to the at2313a/at4323a, one at a time, in order to receive the timestamps using the software serial library.

The controller then processes the data to establish the position of one or more objects within the width of the transmitter beam.
A simple median filter is used to smooth up to 8 pings before processing the timestamps.

At the moment, only very simple tests are done.

Let us call the receiver to the right A, center B and left C.
At the moment triangulation is used in three ways, A-B, B-C, A-C.
(**Arduino Playground - IntersectionOfCircles**)

Before the triangulation, a test is made to determine if the distance A is less than C or B.
If that is the case, the object is to the right of the center.
Next test is to see if distance C is less than A or B, in that case the object is to the left of the center.
X-offset is used to move the resulting position in the x direction when calculating B-C.

Here are some test results, only for the first echo shown.

```
First with a wall/board in front:
A 1472usec:    25.82cm
B 1492usec:    26.18cm
C 1487usec:    26.09cm
Echo: 0
Point A (0.00x,0.00y), B (8.00x,0.00y), Obj (2.86x,25.67y)
Distance AB is 8.00, X-offset 0.00
Distance A to object is 25.82
Distance B to object is 26.18
Echo: 0
Point B (8.00x,0.00y), C (8.00x,0.00y), Obj (12.29x,25.82y)
Distance BC is 8.00, X-offset 8.00
Distance B to object is 26.18
Distance C to object is 26.09
Echo: 0
Point A (0.00x,0.00y), C (16.00x,0.00y), Obj (7.57x,24.69y)
Distance AC is 16.00, X-offset 0.00
Distance A to object is 25.82
Distance C to object is 26.09

A can, 5cm in diameter, is used for the next tests.

Placed near the center:
```

Almost usable to avoid objects in the path.
I have searched for ways to do more accurate calculations, but so far without success.

Any suggestions?

---

johnwasser 2 September 6, 2021, 8:36pm

> ardcp:
>
> Any suggestions?

I would not remove the processor on HC-SR04's and let them do the echo timing. I would remove the transmitter from one HC-SR04 and trigger them both with one Arduino pin. I would use Timer1 and an overflow interrupt to get a 16 MHz clock. Use CHANGE interrupts to measure the pulses on the Echo pins.

Here is a 16 MHz timer I wrote for such an occasion:

```
// Fast Timer
// Written by John Wasser
//
// Returns the current time in 16ths of a microsecond.
// Overflows every 268.435456 seconds.

// Note: Since this uses Timer1, Pin 9 and Pin 10 can't be used for
// analogWrite().



void StartFastTimer()
{
  noInterrupts ();  // protected code
  // Reset Timer 1 to WGM 0, no PWM, and no clock
  TCCR1A = 0;
  TCCR1B = 0;

  TCNT1 = 0;  // Reset the counter
  TIMSK1 = 0; // Turn off all Timer1 interrupts

  // Clear the Timer1 Overflow Flag (yes, by writing 1 to it)
  // so we don't get an immediate interrupt when we enable it.
```

Paul_KD7HB 3 September 6, 2021, 10:35pm

> ardcp:
>
> hc-sr04

As I recall, the hc-sr04 produces a series of 5 pulses and then begins to listen for an echo. Are you able to tell which pulse you are receiving the echo from?
Paul

Idahowalker 4 September 6, 2021, 10:42pm

The SR04 transducer is tuned to 40Khz, the receiver of the module is tuned to 38Khz. For better accuracy the sr04 can be modified to receive on 40Khz; research required on your part at this point.

Anyways fixing the sr04 receiver will increase sensitivity.

The sr04 has a terrible higher voltage regulator for the transducer. Rebuilding the transducer regulator will be of great benefit.

ardcp 5 September 7, 2021, 3:36pm

Only long after submitting the question and looking at the data, I realised that the triangulation works as expected.
There are still issues like delays which have not been considered for the current test setup.

I use the at2313/at4313 in order to measure the time for one or more echoes.
As the timer starts from zero when the echo pin of the transmitter/receiver rises, all three timers start from zero at the same time, within a few clocks. As no other interrupt than the analog comparator is active, it is easy to get accurate timestamps.
Hence no need to check for overflow as it takes 32K usec using a 2 Mhz timer clock.
If an 8-bit timer is used, however, the overflow would need to be handled as you do in the FastTimer example.
Here is the sketch used as is.
It compiles for the at4313, but an additional non-buffered serial library is needed for the at2313 due to the smaller memory.

```
/***
ATMEL (ATTINY2313a)/4313a / ARDUINO
Pin 1 (PA2) analog input Vcc = 1, Vcc -1.2v = 0. Above 2.2v for 5v vc
                           +-\/-+
            RESET      PA2  1|    |20 Vcc 5v
            RXD        PD0  2|    |19 PB7 SCL/SCK/PCINT17
 Data out<- TXD        PD1  3|    |18 PB6 MISO/DO/PCINT16
            XT2/PCINT9 PA1  4|    |17 PB5 MOSI/DI/PCINT5
            XT1/PCINT8 PA0  5|    |16 PB4 OC1B/PCINT4
            INT0/PCI13 PD2  6|    |15 PB3 OC1A/PCINT3    Echo from T/
            PCI14/INT1 PD3  7|    |14 PB2 OC0A/PCINT2    Pulse in, st
            PCI15/T0   PD4  8|    |13 PB1 AIN1/PCINT1    Reference vo
            PCI16/T1   PD5  9|    |12 PB0 AIN0/PCINT0    Signal in fr
                       GND 10|    |11 PD6 ICIPI/PCINT17
                           +----+

PB0 = In, signal from HC-sr04, positive pulses 40 Khz
PB1 = Reference voltage, adjustable.
PB2 = In, pulse in, send data via TXD.    Trig on negative edge comp.
PB3 = In, Start timer1                    Trig on positive edge from
PD1 = Tx, send data

Uses the analog comparator to save the timer count when an echo is de
```

---

ardcp 6 September 7, 2021, 3:50pm

Looking at the output of the receiver using an oscilloscope, there many more than 5 pulses.
Probably due to the transmitting transducer oscillating even after the pulse train has finished.
For fairly close echoes, 20 - 30 cm, the peak voltage is around 4.8v and the pulse train lasts 1 ms or more.
The very first pulse might only have a peak of less than 1 volt, even less for fainter echoes.
At the moment, the reference voltage for the analog comparator is 0.5v, adjustable using a 10K pot.
The very first pulse might then be missed in some cases.
Experiments shall show how low the reference can be set without picking up noise.

DaveEvans 7 September 7, 2021, 7:21pm

eight pulses... **HC-SR04 | David Pilling**

---

anon57585045 8 September 7, 2021, 9:21pm

> ardcp:
>
> Looking at the output of the receiver using an oscilloscope, there many more than 5 pulses.
> Probably due to the transmitting transducer oscillating even after the pulse train has finished.

Not to mention the numerous pulses bouncing around the room... also if the receiver has a high Q factor it will resonate itself.

---

ardcp 9 November 21, 2021, 9:33am

After many tests, I am now trying to determine the length of an echo using an at2313/at4323 piggybacked on a hc-sr04 in order to process the output of detected echos using the analog comparator.

Detecting the start of one or more echos is not a problem using interrupts.
The timer1 is set to run at 2 Mhz, giving 0.5 us resolution and is only active after the transmitter rises the echo signal.
To be able to watch the output on a scope, a pin was toggled for each pulse giving a 20 Khz square wave(in theory).
As there is around 20 us (16Mhz clock) within the isr to do processing, first step was to count the pulses within the first echo.
Finally, the timer ticks for start and end of the echo is now recorded.
Using interrupts, the end is not detected, it is only possible to store the timer ticks for each pulse over and over until a gap appears and a new echo starts.
Seems to be cutting the time fine as is.
Essentially casting a shadow the width of the echo across the field of view where another object might hide.

Knowing the length of the echo makes it possible to detect another object beyond the shadow.
Makes for a very low resolution sonar.

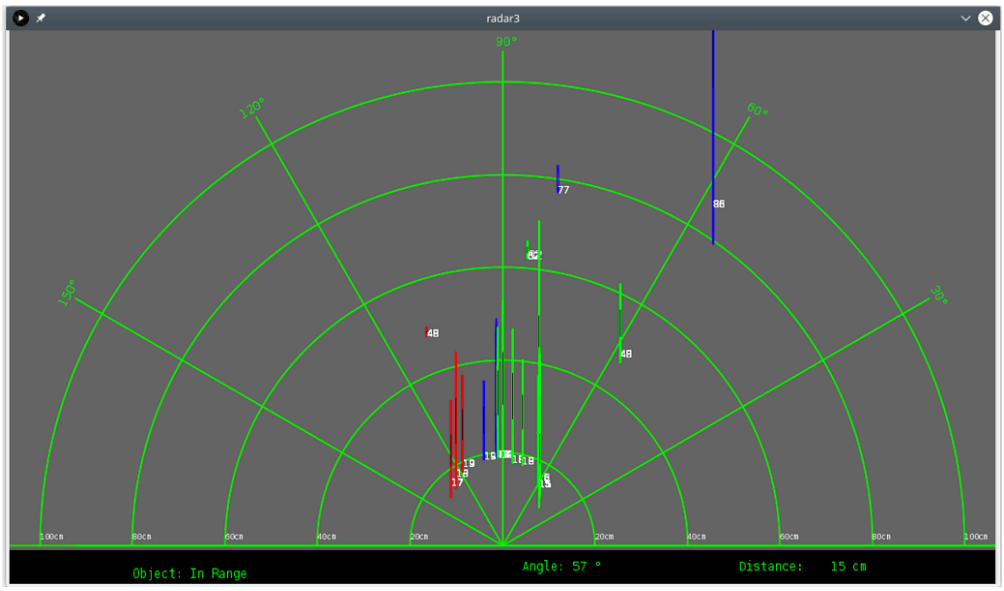
Is there an easier way to obtain the length of an echo than shown in the isr code below?

```
#define ROUNDTRIP_CM 57         //Sonar centimeter round trip in us
#define MAX_ECHOS   4

#ifdef USE_16MHZXTAL
#define PULSEWIDTH  58        //Just above 25 usec, 2 Mhz
#else
#define PULSEWIDTH  28
#endif

#ifdef USE_16MHZXTAL
#define MIN_ECHO_DISTANCE ROUNDTRIP_CM * 2
#else
#define MIN_ECHO_DISTANCE ROUNDTRIP_CM
#endif

volatile uint16_t distance[MAX_ECHOS];      //Store up to 4 echos
volatile uint16_t echo_end[MAX_ECHOS];      //Store end time of echos
volatile int8_t distcount;
volatile uint16_t old_clock;
volatile byte state;       // 0-= idle, 1 = store echo time, 2 = send
volatile uint16_t lastPulse;
```

---

ardcp 11 November 23, 2021, 7:57pm

Just a quick update.
The ISR code posted did not work as intended. It is still being modified.
However, I have now modified radar3 to show echoes with a shadow.
The length of the shadow is the number of 40 Khz pulses in the received echo.



The image shows the result of moving a can, Ø 5cm, across the three hc-sr04.
In short, the green color is the result of triangulating the right and center listener, the red is the result using left and center listener, and the blue is result from using right and left listener.
Also shown are some random echoes.

---

anon57585045 12 November 23, 2021, 11:42pm

So, a complete failure, right?

---

system Closed 13 March 23, 2022, 11:42pm

This topic was automatically closed 120 days after the last reply. New replies are no longer allowed.

Related Topics

Topic	Replies	Activity
Using three HC-SR04 sensors in tandem	3	May 6, 2021
Triangulation HC-SR04	20	October 7, 2022

Topic	Replies	Activity
HCSR04 ultrasonic sensor as transmitter/receiver only?	3	May 6, 2021
How can I use HCSR04 to measure distance between two object?	26	March 27, 2024
Measuring distance with hc-sr04	6	May 6, 2021