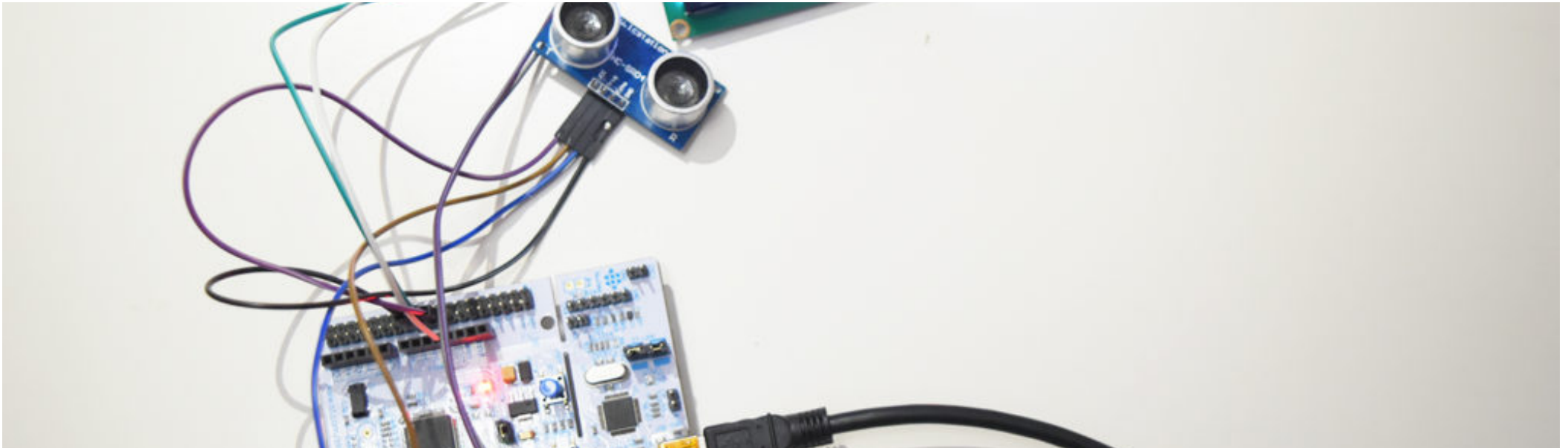


[English](#)[Italiano](#)

# PLAY embedded

[ARTICLES](#)[HOW TO START](#)[DOWNLOADS](#)[FORUM](#)[ABOUT](#)

[PLAY Embedded](#) > [Articles](#) > [ChibiOS and STM32](#) > [Interfacing externals](#) > [Detecting obstacles using an ultrasonic sensor HC-SR04](#)

# Detecting obstacles using an ultrasonic sensor HC-SR04

Published on Monday, January 9, 2017 Updated on Saturday, August 5, 2017

## 1 HC-SR04: a cheap ultrasonic sensor

### 1.1 Description

HC-SR04 is a commonly used ultrasonic sensor which is capable to detect obstacles in a range of 2-300cm. The sensor looks like a small PCB having two metal cylinders on the front-side and a small circuit on the back-side (see Fig.1).

### 1.2 Documentation

In this article we will provide a simple demo to use HC-SR04. This requires a preliminary read of the sensor user manual:

 [HC-SR04 User Manual](#) (78.74 kB)

### 1.3 Pinmap description

The sensor has 4 PIN:

1. **VCC**, which must be connected to 5V;
2. **Trig**, which is an input PIN to trigger the measurement;
3. **Echo**, which is an output PIN which sent out a square wave;
4. **GND**, which must be connected to ground.

*Note that user manual specifies which GND must be connected first since a floating 5V could permanently damage the sensor.*

## 2 How it works

The two cylinders are actually an ultrasonic receiver (R) and an ultrasonic transmitter (T) which are driven by the circuitry. The working principle is quite simple: when triggered the transmitter shoots some pulses. When sound encounters an obstacle it is echoed back and detected by the receiver.



Fig.1 - An HC-SR04 seen from front side.

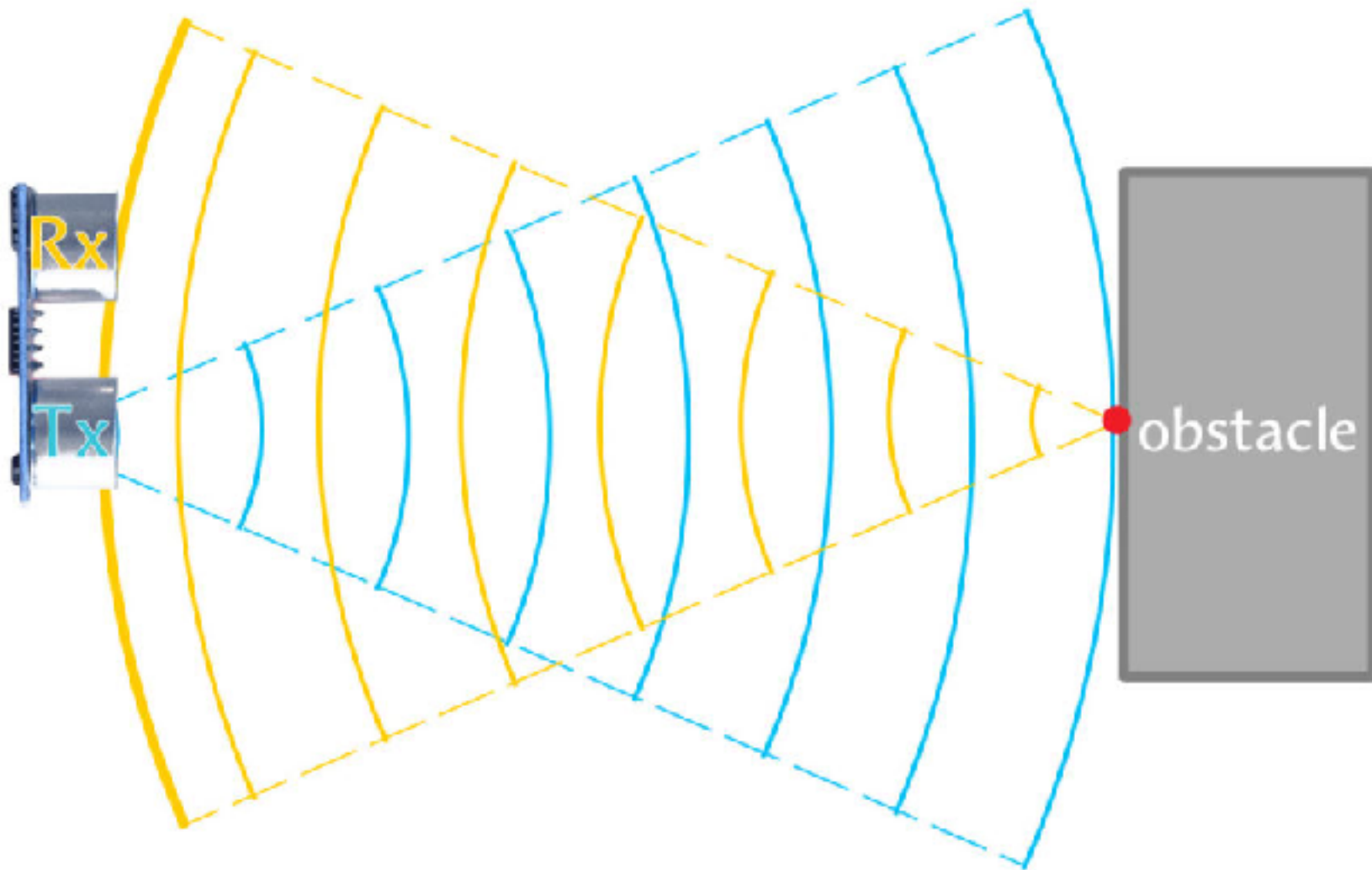


Fig.2 - An example of obstacle detection using an HC-SR04

Since the speed of sound is almost constant, the information about the distance is strictly related the elapsed time from the shoot to the receiving. Indeed, sound has to travel twice the distance (roundtrip). Known the elapsed time the distance to obstacle is

$$d = \frac{v \cdot t}{2}$$

where

- **d** is distance between the obstacle and the sensor
- **v** is the speed of sound (about 340 m/s)
- **t** is time elapsed

Note that, here we are not discussing about accuracy of the measurement which is affected from from different factors:

- speed of sound is not constant: it depends on material medium, temperature, pressure and humidity.
- time measurement accuracy: the measure is made by on board circuit and we don't have control on this.
- the obstacle may not have a flat surface and there could be multiple diffraction or spurious echo.

### 3 How to use it

If **Trig** is set as high for at least 10us the sensor will shoot 8 pulses of sound. For the the whole time required by the sound to made the road trip **Echo** will be high.

The logical steps to get a measure would be:

1. Set the Input Capture Unit to measure the high period of a square wave on the **Echo** pin;
2. Pull up **Trig** signal for at least 10 us;
3. Convert the measured time in a distance using the previous formula.

### 4 A simple demo

In this first demo we will use a STM32 Nucleo-64 F401RE and ChibiOS in order to detect an obstacle and print the distance to it on the ST-Link virtual COM port. To do this we will use `chprintf()` already explained in [Hello ChibiOS](#) and the ICU driver already used with [DHT11](#) and [IR Remote](#).

#### 4.1 Connections

The PINs are connected as follow:

1. VCC, to 5V;
2. Trig, to PB10;
3. Echo, to PA8;
4. GND, to ground.

## 4.2 Demo explained

Taking a look to *main.c* we can see how this demo works. After few initializations the code starts to capture square waves on PA8.

```
#define ICU_TIM_FREQ          1000000
#define M_TO_CM              100.0f
#define SPEED_OF_SOUND       343.2f

static float lastdistance = 0.0;

static void icuwidthcb(ICUDriver *icup) {

    icucnt_t width = icuGetWidthX(icup);
    lastdistance = (SPEED_OF_SOUND * width * M_TO_CM) / (ICU_TIM_FREQ * 2);
}

static ICUConfig icucfg = {
    ICU_INPUT_ACTIVE_HIGH,
    ICU_TIM_FREQ, /* 1MHz ICU clock frequency. */
    icuwidthcb,
    NULL,
    NULL,
    ICU_CHANNEL_1
}
```

As said, the information is encapsulated in the width of the squarewave: computing the previous explained formula it is possible to get distance starting from width. The distance will be stored in a variable which is used by main.

```
while (true) {
    /* Triggering */
    palWriteLine(LINE_TRIGGER, PAL_HIGH);
    chThdSleepMicroseconds(10);
    palWriteLine(LINE_TRIGGER, PAL_LOW);
#ifdef ANSI_ESCAPE_CODE_ALLOWED
    chprintf(chp, "\033[2J\033[1;1H");
#endif
}
```

```
chprintf(chp, "Distance: %.2f cm\n\r", lastdistance);
chThdSleepMilliseconds(100);
}
```

The main loop triggers the measurement providing a rising-edge on **Trig** and prints the last measure on the STLink Virtual COM port.

## 5 HC-SR04 and HD44780: mixing two demos

In the article [How to drive a HD44780 with I2C backpack with a STM32](#) we have shown how to use an LCD 16x2 based on HD44780 provided of an I2C backpack. It is possible to create a more interesting demo including the LCD library in order to print the distance on the display instead of the Virtual COM port.

### 5.1 Connections

The PINs are connected as follow:

1. **VCC** (HC-SR04) and **VCC** (HD44780), to 5V;
2. **GND** (HC-SR04) and **GND** (HD44780), to Ground;
3. **Trig** (HC-SR04), to PB10;
4. **Echo** (HC-SR04), to PA8;
5. **SDA** (HD44780), to PB9;
6. **SCL** (HD44780), to PB8.

### 5.2 Demo explained

Lets take a look to the main loop

```
lcdStart(&LCDD1, &lcdfcg);
lcdWriteString(&LCDD1, "Distance", 0);

while (true) {
    /* Triggering */
    palWriteLine(LINE_TRIGGER, PAL_HIGH);
    chThdSleepMicroseconds(10);
    palWriteLine(LINE_TRIGGER, PAL_LOW);

    chsnprintf(string, sizeof(string), "%.2f cm", lastdistance);
    lcdWriteString(&LCDD1, string, 40);
    chThdSleepMilliseconds(200);
}
```

We can note few things:

- `chprintf()` is replaced by a `lcdWriteString()`;
- first line is printed only once before the loop since it doesn't changes;
- `chsnprintf()` is a custom implementation of `snprintf()`;
- printing the last distance we add some spaces to avoid misprints when the string size decrease.

## 6 Downloads

These demos have been tested under ChibiOS 16.1.7 and work with each ChibiOS 16.1.x release:



RT-STM32F401RE-NUCLEO-HC-SR04 (15.54 kB)



RT-STM32F401RE-NUCLEO-HC-SR04+HD44780-PCF8574 (23.28 kB)

## Taglist

[STM32](#) [ChibiOS](#) [STM32F4](#) [STM32 Nucleo](#) [Counter Unit](#) [HD44780](#) [HC-SR04](#)

## Be the first to reply at Detecting obstacles using an ultrasonic sensor HC-SR04

### Leave a Reply

Enter your comment here...

Powered by [Wordpress](#) • PLAY Embedded - Copyright © 2014-2017 Rocco Marco Guglielmi e Salvatore Dello Iacono

u