

Arduino Courses

Ultrasonic Sensor

November 2019

Lorenzo Prosseda
lerokamut@gmail.com

POLITECNICO OPEN
unix LABS



Released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International license.

1. SENSORS

Sensors are electrical devices which listen to the physical world: they can do this by converting the energy related to a specific phenomenon into electrical signals, that will be fed into the microcontroller of the Arduino board.

In fact, microcontrollers *listen* to sensors and use the information they provide to take decisions or control actuators, which are instead devices that convert electrical energy into physical energy (light, movement, heat, ...).

Practically speaking, the output of a sensor is some value of electric voltage, which can be analogue (it can acquire every possible value in a real interval) or digital (the sensor output voltage will settle around some discrete values from an interval).

2. ULTRASOUNDS

So called sound waves which frequencies are beyond the upper limit of the audible spectrum ($> 20kHz$); they are just sound waves that humans cannot hear.

We are interested in the application of ultrasounds for object detection and distance measurement: we chose the HC-SR04 sensor to achieve the latter, and below we will see how to hook it up to an Arduino compatible board.

2.1. HC-SR04. This sensor is capable of measuring from 2cm to 40cm, with an error of 3mm; these numbers are theoretical, and only valid in the best case scenario, that is a big, flat and smooth surface in front of the sensor.

The principle behind the sensor is the emission of a series of high frequency sound pulses, and the detection of the same pulses from nearby the source; the time between send and detection will be used to compute the approximate distance of a surface in front of the sensor.

The sensor connection bindings (*pinout*) follows:

Ultrasonic sensor	↔	Arduino board
VCC	↔	+5V
GND	↔	GND
Trig	↔	GPIO5
Echo	↔	GPIO6

You may use jumper wires to connect the sensor to the specified Arduino pins directly, or you could better cable manage your project with the help of a breadboard.

You could find more information about this sensor by looking up its datasheet: that is a document with all the specifications regarding this sensor's constraints and requirements in order to use it in your project. Here it is a [link](#) to it.

2.2. Ultrasonic_hc_sr04 library. We shall use a *library* to manage to quickly get up and running with our sensor: Patrick Bobbinik has made available his own through the Arduino IDE Library manager; we will look for the library named `Ultrasonic_hc_sr04` with version number `0.5.0`.

According to the [official guide](#), we can add a library to the IDE by following the steps below:

- (1) click the *Sketch* drop down menu;
- (2) hover the *Include Library* menu item;
- (3) click the *Manage libraries* sub-menu item;
- (4) wait for the cache to be reloaded;
- (5) type the library or the sensor name in the text field;
- (6) in case of multiple results, look for a suitable one;
- (7) select the corresponding entry and select a version from the drop-down menu;
- (8) hit the *Install* button.

3. THE SKETCH

The following code snippet will be enough to use the sensor through the previously addressed library:

```
#include "Ultrasonic.h"

#define TRIG_PIN 5
#define ECHO_PIN 6

Ultrasonic mySensor(TRIG_PIN, ECHO_PIN);

void setup() {
  Serial.begin(115200);
  while(!Serial) {
    delay(10);
  }
}

void loop() {
  mySensor.measure();
  Serial.println(mySensor.get_cm(), 0);
  delay(50);
}
```

During this talk we will use an Arduino Leonardo, so you should better select it from the boards manager. To look at the distance measured by the sensor, we can use the serial monitor or the serial plotter of the Arduino IDE.

In the code snippet above, the baud rate for the serial communication is set to 115200; you can find the snippet at the following [link](#).