# A PharoThings Tutorial

Allex Oliveira

January 10, 2019

Layout and typography based on the `sbabook` LaTeX class by Damien Pollet.

# Contents

# Illustrations

CHAPTER 1

# Lesson 9 - Ultrasonic Sensor (Distance)

In the previous lessons, we learned how to control LEDs and to use a button to interact with LEDs. We learned also how to use I2C sensors to read the temperature, humidity, pressure, and x, y, z-axis. Now let's use a different kind of sensor, that doesn't use I2C protocol.

## 1.1 What do we need?

In this lesson, we will use a setup with an ultrasonic sensor.

### Components

- 1 Raspberry Pi connected to your network (wired or wireless)
- 1 Breadboard
- 1 HC-SR04 sensor
- 1 Resistor (1K ohms)
- 1 Resistor (2K ohms)
- Jumper wires

## 1.2 Experimental theory

Before constructing any circuit, you must know the parameters of the components in the circuit, such as their operating voltage, operating circuit, etc.

### The ultrasonic measure

The ultrasonic sensor that we will use in this tutorial has 4 PINs: GND Ground, VCC 5V, TRIG to send a pulse and ECHO to show how long time the pulse spend to go and back.

### How the HC-SR04 works?

It uses our Raspberry Pi to send a signal using the TRIG, which triggers the sensor to send an ultrasonic pulse. Pulse waves reflect any nearby objects and some are reflected back to the sensor. The sensor detects these return waves and measures the time between the trigger and the returned pulse and sends a 5V signal on the ECHO pin. So is just to see how long time the ECHO pin will stay ON and calculate the distance because we already know the sound speed.

### Limiting the return voltage with resistors

The ECHO signal in sensor HC-SR04 uses 5V. But the Raspberry Pi PINS uses 3.3V. If we send 5V to Raspberry GPIO, it can damage the Rasp. To avoid this let's put 2 resistors to limit the voltage.

## 1.3 Experimental procedure

Now we will build the circuit. This circuit consists of 1 ultrasonic sensor HC-SR04, 1 resistor 1K ohms, 1 resistor 2K ohms and a power supply (the Rasp).

- Connect the Ground PIN from Raspberry in the breadboard blue rail (-). In this experiment we will use the PIN9 (Ground);
- Then connect the 5V (PIN2) pin in the red rail (+).
- Now push the HC-SR04 sensor in the breadboard;
- And insert the jumper wires connecting the sensor leg TRIG in the GPIO0 (PIN11) and the sensor leg ECHO in the breadboard, like the scheme shown in the Figure 1-1;
- The last step is to connect the power 5V and ground (-) wires from the breadboard rails in the sensor VCC + and GND (-) legs.

The Figure 1-1 shows how the electric connection is made.

## 1.4 Connecting remotely

Through your local Pharo image, let's connect in the Pharo image by running on Raspberry, enable the auto-refresh feature of the inspector, and open the inspector. Run this code in your local playground:

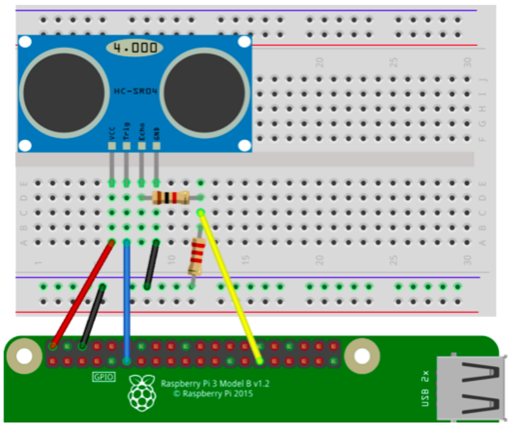

**Figure 1-1** Physical sensors connection.

```
remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip: #[193 51 236
    212] port: 40423)
GTInspector enableStepRefresh.
remoteBoard := remotePharo evaluate: [ RpiBoard3B current].
remoteBoard inspect.
```

## 1.5 Experimental code

In your inspect window (Inspector on a PotRemoteBoard), let's create an instance of the ultrasonic sensor.

```
d := board installDevice: PotHCSR04Device new.
```

As we saw before, we can inspect the remote object to see some properties and methods. Let's use the method `readDistance` to read the distance:

```
d readDistance.
```