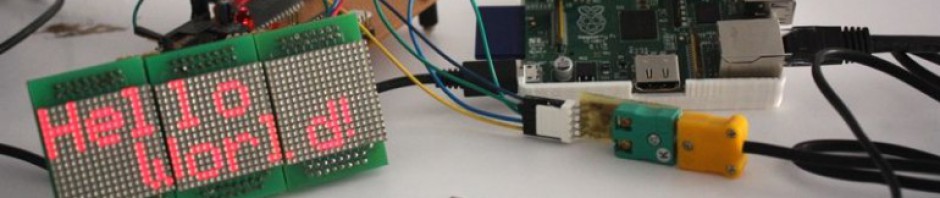
**[Notes on robotics, operating systems and programming](https://arduibots.wordpress.com/)**

*Personal projects on free platforms*

**

[Skip to content](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#content)

* [Home](https://arduibots.wordpress.com/)
* [Fritzing components](https://arduibots.wordpress.com/fritzing/)

[← Arduino-controlled basic mobile robot](https://arduibots.wordpress.com/2013/12/24/arduino-controlled-basic-mobile-robot/)

[Component bluetooth HC-06 for Fritzing →](https://arduibots.wordpress.com/2015/02/26/component-bluetooth-hc-06-for-fritzing/)

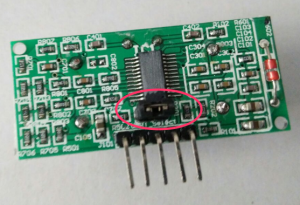
## **[US-100 ultrasonic sensor in serial mode](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/)**

Posted on [12 October, 2014](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/)by [Rafa G](https://arduibots.wordpress.com/author/shindenfudo/)

In a [previous article](https://arduibots.wordpress.com/2013/10/14/comparison-of-ultrasonic-sensors-us-100-and-hc-sr04/) we used the US-100 ultrasonic sensor in pulse width mode, consisting primarily of sending a first start pulse and measure the pulse width of return. The problem of this mode of operation is that fails to take advantage of the characteristics of this sensor, which is able to compensate its calculations from room temperature.

In this article we will study the behavior of US-100 serial mode.

## Settings and messaging of US-100 in serial mode

The serial mode for this sensor is set up via jumper. So, we will ensure that the rear jumper is closed.

In this mode, serial communications take place via those pins marked as Tx and Rx, which would be connected respectively to the transmit and receive pins of our Arduino. The serial port is configured as 9600 8N1.

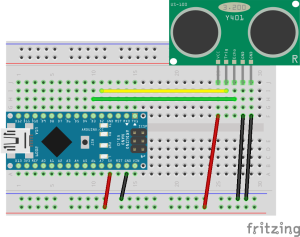
In terms of messaging, this device allows two commands in serial mode:

* 0x55: distance measurement request
* 0x50: temperature measurement request

Response messages are:

* Distance in millimeters, two bytes according to the expression: *MSB \* 256 + LSB*
* Temperature in degrees Celsius, one byte according to the expression: *BYTE – 45*

## Mounting the test circuit

We will make the assembly shown in figure based on the following components:

* [Protoboard](http://dx.com/p/121529?Utm_rid=72096337&amp;Utm_source=affiliate)
* [Arduino Nano V3.0](http://dx.com/p/81877?Utm_rid=72096337&amp;Utm_source=affiliate)
* [US-100](http://dx.com/p/144433?Utm_rid=72096337&amp;Utm_source=affiliate)
* [Jumper wires](http://dx.com/p/118826?Utm_rid=72096337&amp;Utm_source=affiliate)

As you can see in the figure, we connect pins 2 and 3 with the pin “trigger / tx” and “echo / rx” the US-100. The US-100 also requires that the two pins “GND” are shorted and connected to GND in Arduino.

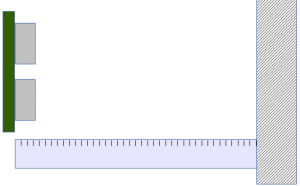
## Arduino programming

We use the following code to check the sensor:

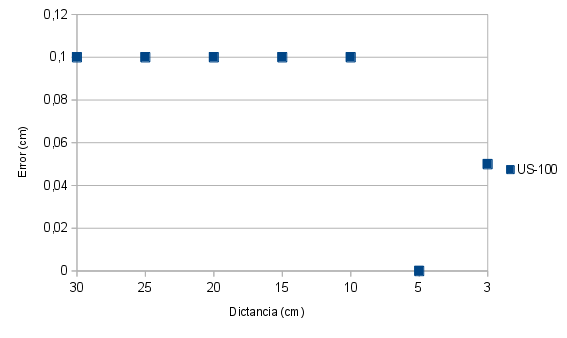
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59 | // Prueba de medidas de distancia y temperatura con US-100 en modo serie  // Por RafaG, 2014    #include <SoftwareSerial.h>;    const int US100\_TX = 2;  const int US100\_RX = 3;    // Instancia nuevo canal serie  SoftwareSerial puertoUS100(US100\_RX, US100\_TX);    unsigned int MSByteDist = 0;  unsigned int LSByteDist = 0;  unsigned int mmDist = 0;  int temp = 0;    void setup() {  Serial.begin(9600);  puertoUS100.begin(9600);  }    void loop() {    puertoUS100.flush(); // limpia el buffer del puerto serie  puertoUS100.write(0x55); // orden de medición de distancia    delay(500);    if(puertoUS100.available() >= 2) // comprueba la recepción de 2 bytes  {  MSByteDist = puertoUS100.read(); // lectura de ambos bytes  LSByteDist = puertoUS100.read();  mmDist = MSByteDist \* 256 + LSByteDist; // distancia  if((mmDist > 1) && (mmDist < 10000)) // comprobación de la distancia dentro de rango  {  Serial.print("Distancia: ");  Serial.print(mmDist, DEC);  Serial.println(" mm");  }  }    puertoUS100.flush(); // limpia el buffer del puerto serie  puertoUS100.write(0x50); // orden de medición de distancia    delay(500);  if(puertoUS100.available() >= 1) // comprueba la recepción de 1 byte  {  temp = puertoUS100.read(); // lectura del byte  if((temp > 1) && (temp < 130)) // comprobación de rango válido  {  temp -= 45; // corrige offset de 45º  Serial.print("Temperatura: ");  Serial.print(temp, DEC);  Serial.println(" ºC.");  }  }    delay(500);  } |
| --- | --- |

Notice that we use the library **SoftwareSerial** in order to enable pins 2 and 3 as a new serial port for our Arduino. Otherwise, the code simply sends commands on distance measurement and temperature, showing the results.

## Performing the tests and results

Once done the previous assembly -and propperly coded our Arduino-, our *breadboard* with the sensor installed is placed in front of a solid surface, using a ruler to reference the measures, such as is indicated in the figure:

By reference to a centimeter ruler, five measurements are taken for each of the reference distances (30, 25, 20, 15, 10, 5 and 3 cm). The errors of the mean for each group relative to the reference measurements are shown in the following graph:

It can be seen that the measurement error remains within the manufacturer’s specifications. For small distances the error becomes negligible.

Advertisements

Report this ad

Report this ad

### **Compártelo:**

* [Twitter](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?share=twitter&nb=1)
* [Facebook1](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?share=facebook&nb=1)
* [Google](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?share=google-plus-1&nb=1)

### **Related**

****

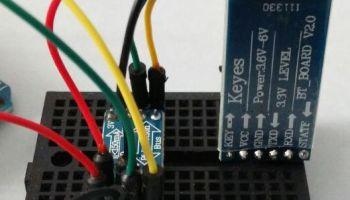
#### [Comparison of ultrasonic sensors US-100 and HC-SR04](https://arduibots.wordpress.com/2013/10/14/comparison-of-ultrasonic-sensors-us-100-and-hc-sr04/)

In "Arduino"



#### [Arduino-controlled basic mobile robot](https://arduibots.wordpress.com/2013/12/24/arduino-controlled-basic-mobile-robot/)

In "Arduino"



#### [Communicating our Arduino with an Android phone via bluetooth](https://arduibots.wordpress.com/2015/02/26/communicating-our-arduino-with-an-android-phone-via-bluetooth/)

In "Android"

This entry was posted in [Arduino](https://arduibots.wordpress.com/category/arduino/) and tagged [Arduino](https://arduibots.wordpress.com/tag/arduino/), [distance sensor](https://arduibots.wordpress.com/tag/distance-sensor/). Bookmark the [permalink](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/).

[← Arduino-controlled basic mobile robot](https://arduibots.wordpress.com/2013/12/24/arduino-controlled-basic-mobile-robot/)

[Component bluetooth HC-06 for Fritzing →](https://arduibots.wordpress.com/2015/02/26/component-bluetooth-hc-06-for-fritzing/)

### **6 Responses to *US-100 ultrasonic sensor in serial mode***

1. ****stoduk** *says:*
2. [18 February, 2016 at 10:07 pm](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#comment-23)
3. In your comparison of HC-SR04 and US-100, the US-100 had errors up to 8.5cm – with the error being roughly linearly increased with distance. [the HC-SR04 has slightly smaller errors, but comparable]
4. In this review are you saying that in serial mode the US-100 gives drastically better accuracy? The max error is “0,1 cm” – is that 1mm? If so, do you think that is because the temperature compensation is coming in to play? Was this test done with the same unit from the previous test?
5. TIA
6. Like
7. [Reply](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?replytocom=23#respond)
   * [](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?replytocom=23#respond)[**Rafa G**](http://minibots.wordpress.com/) *says:*
   * [18 February, 2016 at 10:19 pm](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#comment-24)
   * Temperature compensation is very important to improve the accuracy but, in this case, it’s even more important the fact that the calcs are made by the same sensor and not the Arduino.
   * Like
   * [Reply](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?replytocom=24#respond)
     + [](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/?replytocom=24#respond)**stoduk** *says:*
     + [18 February, 2016 at 10:48 pm](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#comment-25)
     + OK, guess that makes sense – looking at the NewPing library it is polling every 24us and claims that limits the accuracy. It always seemed odd to have the SR04 be smart enough to do the trigger/echo dance, when it could just report back the measurement directly. I should have realised the cheapest module was a dud, and gone for the US-100 🙂
     + Thanks!
     + Like
     + **stoduk** *says:*
     + [10 March, 2016 at 2:42 pm](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#comment-26)
     + Hi,
     + I’ve got a US-100 now and the results are impressive. I’ve got it sat at a distance of 672mm from the floor, fixed to my desk pointing downwards.
     + Using it in the dumb mode (ie. sending ping, measuring echo) I see the measurement is much higher than it should be (~795mm, so 123mm). I also see much more variability in the numbers – from 792 to 798mm.
     + Using it in the serial mode I see the measurement is both more accurate (672mm exactly) and far more stable (it barely changes, occasionally I’ll see it jump up by 1mm). The stability isn’t as simple as the US-100 faking it by doing some smoothing on the values it returns – fetching readings every 100ms, if I wave my hand past the sensor I will see one lower number with perfectly accurate readings either side, so it responds quickly.
     + This backs up all your findings I think, and shows how much better the US-100 is over the more basic models.
     + The two flaw of US-100 for me are:
     + 1) it is a different size to the HC-SR04 (board is ~1mm less long, so the mounting holes are at a different distance, and the ultrasonic tx/rx devices have a different separation). Only bad if you wanted to use this as a drop in replacement where you have an existing housing or mounting to fit it to.
     + 2) The serial data returned is rather dumb, so it wouldn’t be hard to trip up – specifically the data returned doesn’t identify what it value it is returning (or which byte is the start value). This would be more of a problem if you wanted to get distance \*and\* temperature readings – otherwise you should be able to avoid problems with some heavy handed pauses to get in step with the US-100’s output.
     + Cheers,
     + Anthony
     + Like
     + **stoduk** *says:*
     + [10 March, 2016 at 3:29 pm](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#comment-27)
     + [wish I could edit comments!]
     + Update to my last post: there was a bug in the test using trigger/echo mode, that meant the result was scaled by 57/50. This means it was returning a value of (795 \* 50 / 57 = ) 697mm. So it is still overestimating the distance, but by more like 25mm. The comments about variability still stand (6mm from lowest to highest reading).
     + Like
     + [**stoduk**](http://fsmilestest.wordpress.com/) *says:*
     + [19 July, 2016 at 9:52 pm](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/#comment-37)
     + Ages ago I wrote a library for using US-100 in serial mode, I’ve finally put is somewhere so others can benefit. Mentioning here as I figured people might come across this page when they are investigating what to buy.
     + <https://github.com/stoduk/PingSerial>
     + Like

### **Leave a Reply**

### **Entradas recientes: Notas sobre robótica, electrónica, sistemas operativos y programación**

#### [Componente sensor de detección de obstáculos por infrarrojos de tipo tubular para Fritzing](https://minibots.wordpress.com/2018/01/01/componente-sensor-de-deteccion-de-obstaculos-por-infrarrojos-para-fritzing/)

#### [Generación automática del archivo de palabras clave para nuestras librerías de Arduino](https://minibots.wordpress.com/2017/12/31/generacion-automatica-del-archivo-de-palabras-clave-para-nuestras-librerias-de-arduino/)

#### [Control de sensor de ultrasonidos con Nanpy en configuración maestro/esclavo de ArchLinux y Arduino](https://minibots.wordpress.com/2017/12/09/control-de-sensor-de-ultrasonidos-con-nanpy-en-configuracion-maestro-esclavo-de-archlinux-y-arduino/)

#### [ArchLinux y Arduino en configuración maestro/esclavo con Nanpy](https://minibots.wordpress.com/2017/12/09/archlinux-y-arduino-en-configuracion-maestro-esclavo-con-nanpy/)

#### [Ejemplo de utilización del protocolo MQTT con ESP8266](https://minibots.wordpress.com/2017/11/26/ejemplo-de-utilizacion-del-protocolo-mqtt-con-esp8266/)

**[Notes on robotics, operating systems and programming](https://arduibots.wordpress.com/)**

*[Blog at WordPress.com.](https://wordpress.com/?ref=footer_blog)*

* [Follow](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/)

[:)](https://arduibots.wordpress.com/2014/10/12/us-100-ultrasonic-sensor-in-serial-mode/)