# SEED TECHNOLOGY INC (SEEEDUINO)
# Ultra Sonic range measurement module
# Model: SEN136B5B

## Introduction

Seeed ultrasonic sensor is non-contact distance measurement module, which is also compatible with electronic brick. It's designed for easy modular project usage with industrial performance.
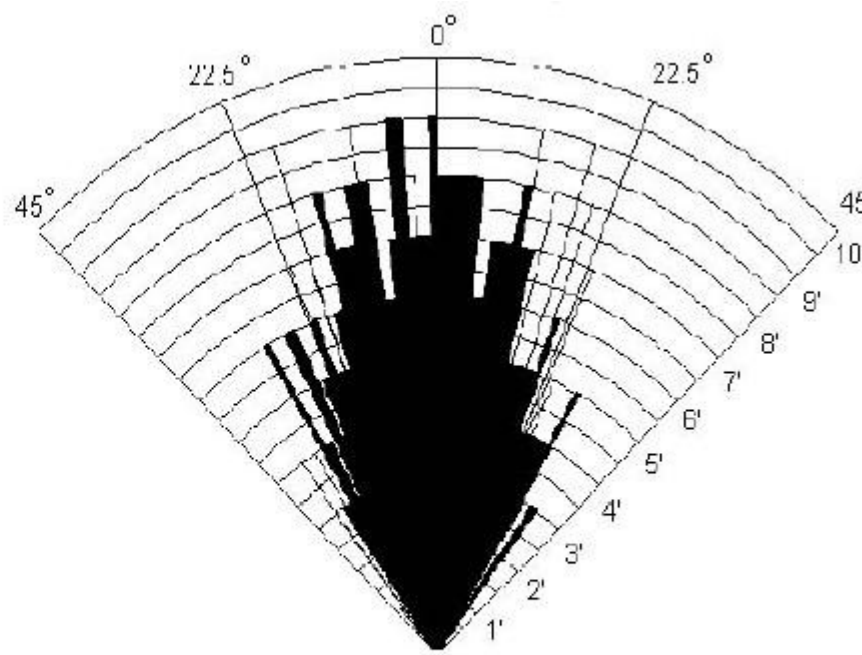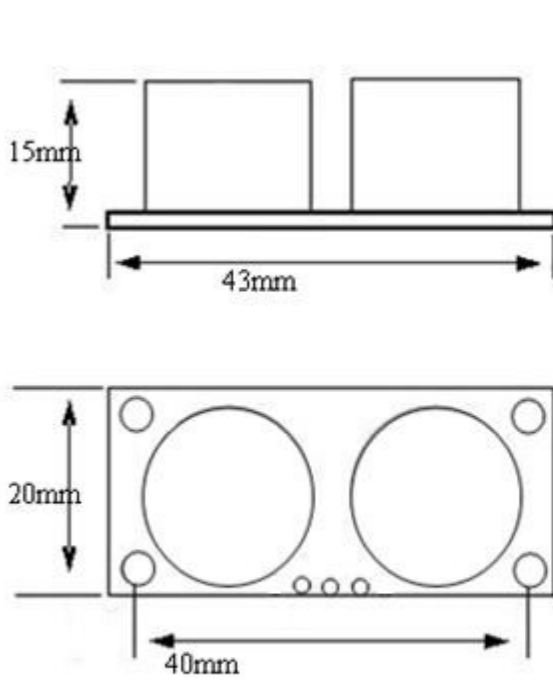


## Features

- Detecting range: 3cm-4m
- Best in 30 degree angle
- Electronic brick compatible interface
- 5VDC power supply
- Breadboard friendly
- Dual transducer
- Arduino library ready

## Specification

| | |
|---|---|
| Supply voltage | 5V |
| Global Current Consumption | 15 mA |
| Ultrasonic Frequency | 40k Hz |
| Maximal Range | 400 cm |
| Minimal Range | 3 cm |
| Resolution | 1 cm |
| Trigger Pulse Width | 10 µs |
| Outline Dimension | 43x20x15 mm |

*Mechanic Dimensions*
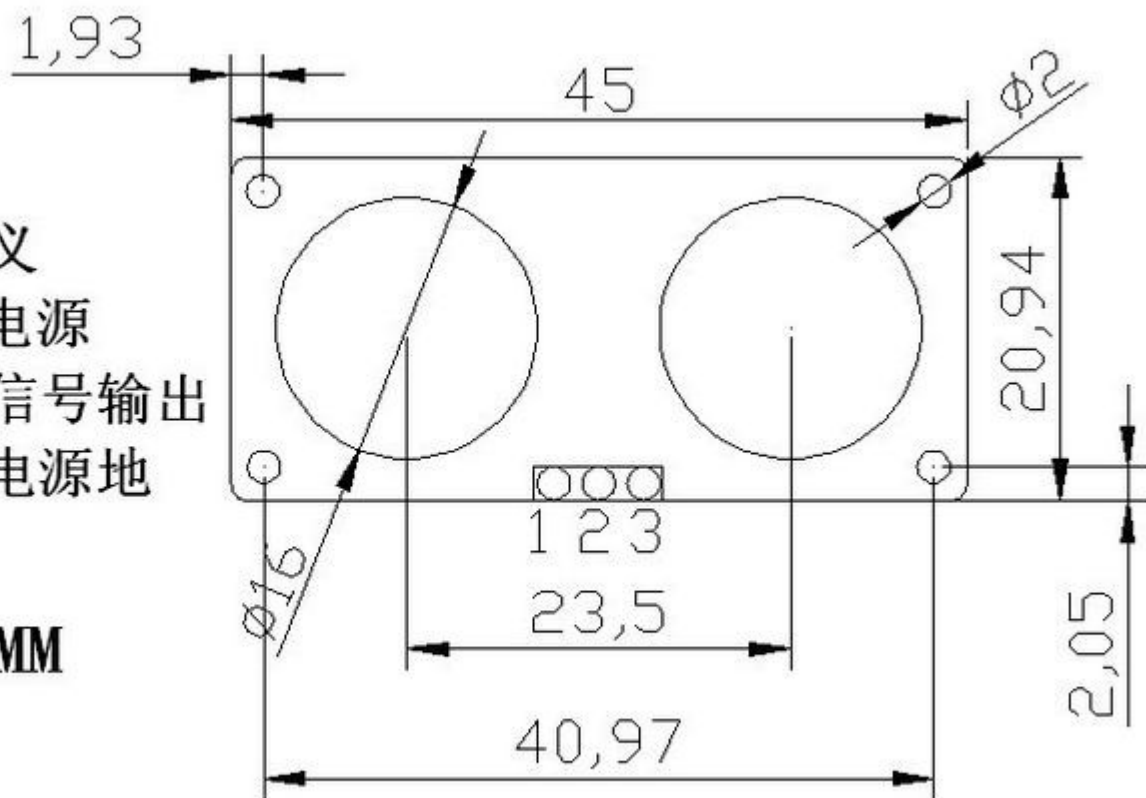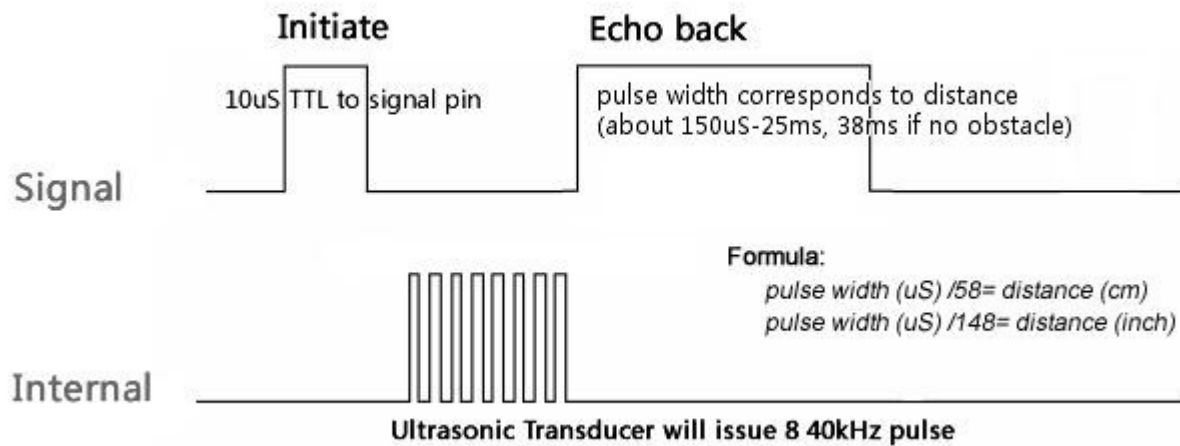


Practical test of performance,
Best in 30 degree angle

接口定义
1:VCC 电源
2:OUT 信号输出
3:GND 电源地

单位:MM

*Usage*

## Hardware Installation



A short ultrasonic pulse is transmitted at the time 0, reflected by an object. The senor receives this signal and converts it to an electric signal. The next pulse can be transmitted when the echo is faded away. This time period is called cycle period. The recommend cycle period should be no less than 50ms. If a 10µs width trigger pulse is sent to the signal pin, the Ultrasonic module will output eight 40kHz ultrasonic signal and detect the echo back. The measured distance is proportional to the echo pulse width and can be calculated by the formula above. If no obstacle is detected, the output pin will give a 38ms high level signal.

## Programming

Includes important code snippet. Demo code like :

```
/* Ping))) Sensor

   This sketch reads a PING))) ultrasonic rangefinder and returns the
   distance to the closest object in range. To do this, it sends a pulse
   to the sensor to initiate a reading, then listens for a pulse
   to return.  The length of the returning pulse is proportional to
   the distance of the object from the sensor.

   The circuit:
    * +V connection of the PING))) attached to +5V
    * GND connection of the PING))) attached to ground
    * SIG connection of the PING))) attached to digital pin 7

   http://www.arduino.cc/en/Tutorial/Ping

   created 3 Nov 2008
   by David A. Mellis
   modified 30 Jun 2009
   by Tom Igoe

   This example code is in the public domain.

 */

// this constant won't change.  It's the pin number
// of the sensor's output:
const int pingPin = 7;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}
```

```
void loop()
{
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);
}

long microsecondsToInches(long microseconds)
{
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second).  This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```

## *FAQ*

Please list your question here:

## *Support*

If you have questions or other better design ideas, you can go to our forum or wish to discuss.

## *Version Tracker*

| Revision | Descriptions | Release |
|---|---|---|
| v1.0 | Seeed Ultrasonic Sensor | May 14, 2010 |

## *Resources*

The resources need to be downloaded, like Eagle file, Demo code, project or other datasheet.

## *Licensing*

This documentation is licensed under the Creative Commons Attribution-ShareAlike License 3.0 Source code and libraries are licensed under GPL/LGPL, see source code files for details.

## *External Links*

Links to external webpages which provide more application ideas, documents/datasheet or software libraries