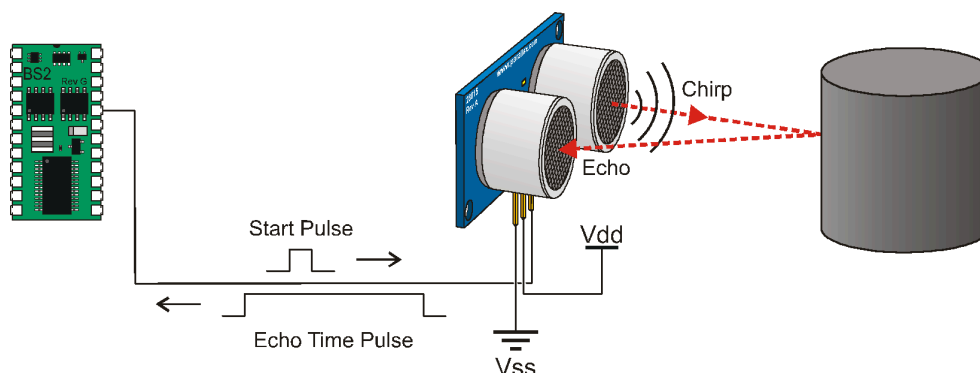


PING))) Ultrasonic Distance Sensor (#28015)

The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 3 cm (1.1 inches) to 3 meters (3.3 yards). It is very easy to connect to nearly any microcontroller; such as the BASIC Stamp®, Parallax Propeller, micro:bit or Arduino. The Parallax PING))) uses an efficient 3-pin interface, requiring only one I/O signal pin along with power and ground.

The default PING))) sensor protocol works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



The Rev E revision of the PING))) sensor keeps the original PING))) protocol by default (after power-on), and adds an additional simplex UART serial mode protocol that streams serial ASCII messages to the microcontroller once activated.

Features

- Range: 3 cm to 3 m (1.1 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse or simplex UART interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2 μ s min, 5 μ s typ.
- Echo pulse: positive TTL pulse, 115 μ s minimum to 18.5 ms maximum.
- Serial mode enable: 3x positive 100 μ s TTL pulses, each with 10 μ s typ. low gap.
- RoHS Compliant

Key Specifications

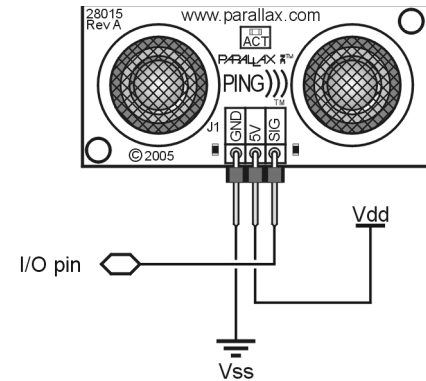
- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse or serial 9600 baud.
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

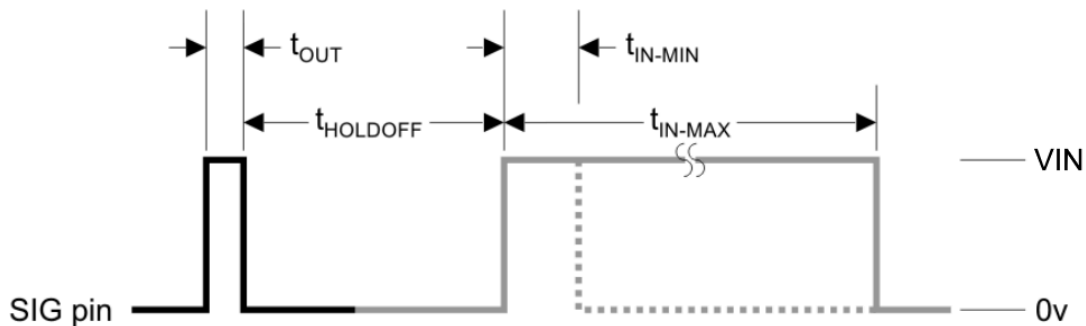
The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged directly into a solderless breadboard, or into a standard 3-wire extension cable (Parallax part #800-00120).



When interfacing to a 3.3V microcontroller it is recommended to add a 1K (or up to 4.7K) series resistor between the PING))) SIG pin and the microcontroller I/O pin.



Communication Protocol (Default TTL Pulse Mode)

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.



	Host Device	Input Trigger Pulse	t_{OUT}	2 μ s (min), 5 μ s typical
	PING))) Sensor	Echo Holdoff	$t_{HOLDOFF}$	750 μ s
		Burst Frequency	t_{BURST}	200 μ s @ 40 kHz
		Echo Return Pulse Minimum	t_{IN-MIN}	115 μ s
		Echo Return Pulse Maximum	t_{IN-MAX}	18.5 ms
		Delay before next measurement		200 μ s

To convert the pulse width from time, in μ s, to mm, use the following equation:

$$\text{Distance (mm)} = \text{Pulse Width } (\mu\text{s}) / 1000 \times 171.5$$

To convert the pulse width from time, in μs , to inches, use the following equation:

$$\text{Distance (inches)} = \text{Pulse Width } (\mu\text{s}) / 1000 \times 6.752$$

Communication Protocol (Serial Data Mode)

Serial data mode was added to PING))) modules marked Rev E (or later). The serial protocol works at 9600 baud on a single I/O pin (SIG), and can communicate with 3.3 V* or 5 V TTL or CMOS microcontrollers. The SIG pin will idle high in this mode, at the VIN voltage. Once Serial mode has been activated, the PING))) sensor operates as a simplex UART device.

To switch from the default TTL pulse mode to serial data mode, drive the SIG pin low, then send three high 100 μs pulses with 10 μs (typical) low gaps between. This can often be achieved by transmitting a capital 'I' character if your microcontroller is not able to transmit the TTL high/low pulses.

In Serial data mode, PING))) will continuously read and then transmit measurement data in ASCII format. The value will be in millimeters, and followed by carriage return character (decimal 13). A new value will be transmitted each time the sensor receives a valid reading, and so the time interval will vary slightly depending on the distance of the detected object; approximately once every 25ms for targets < 1m and approximately 50ms for further targets.

To stop serial mode and return to the default PWM mode, simply power-cycle the PING))) sensor.

A simple code example is provided at our OBEX code repository - search for [Object ID #5253](#).

Tip: For any microcontroller the typical code implementation would wait for the I/O pin to go low, wait 10 μs , and then pulse the I/O pin high 3x 100 μs , with a 10 μs low gap between each high pulse. After that, enable serial-rx on the same I/O pin and read the ASCII string data (with carriage-return termination) being sent by the PING))) sensor.

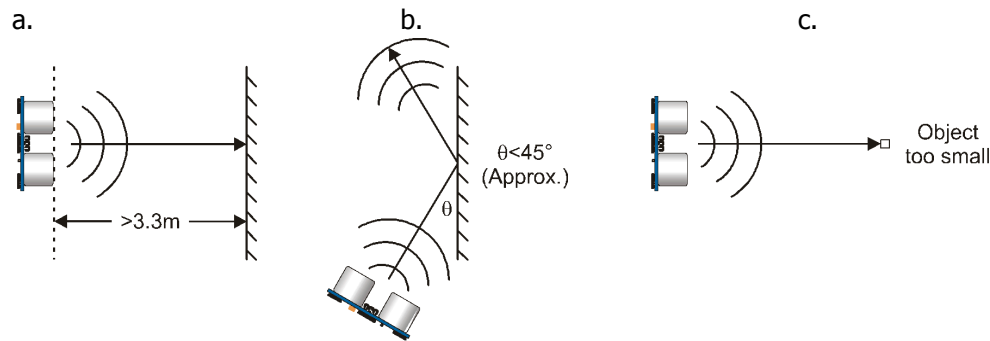
In our example code object #5253 mentioned above, the code has an endless loop that reads the ASCII string, outputs the received data to a serial terminal, then waits for the next ASCII string and repeats itself! This way the user can view the live PING))) distance measurement.

* When interfacing to a 3.3V microcontroller it is recommended to add a 1K (or up to 4.7K) series resistor between the PING))) SIG pin and the microcontroller I/O pin.

Practical Considerations for Use

Object Positioning

The PING))) sensor cannot accurately measure the distance to an object that: a) is more than 3 meters away, b) that has its reflective surface at a shallow angle so that sound will not be reflected back towards the sensor, or c) is too small to reflect enough sound back to the sensor. In addition, if your PING))) sensor is mounted low on your device, you may detect sound reflecting off of the floor.



Target Object Material

In addition, objects that absorb sound or have a soft or irregular surface, such as a stuffed animal, may not reflect enough sound to be detected accurately. The PING))) sensor will detect the surface of water, however it is not rated for outdoor use or continual use in a wet environment. Condensation on its transducers may affect performance and lifespan of the device.

Air Temperature

Temperature has an effect on the speed of sound in air that is measurable by the PING))) sensor. If the temperature (°C) is known, the formula is:

$$C_{\text{air}} = 331.5 + (0.6 \times T_c) \text{ m/s}$$

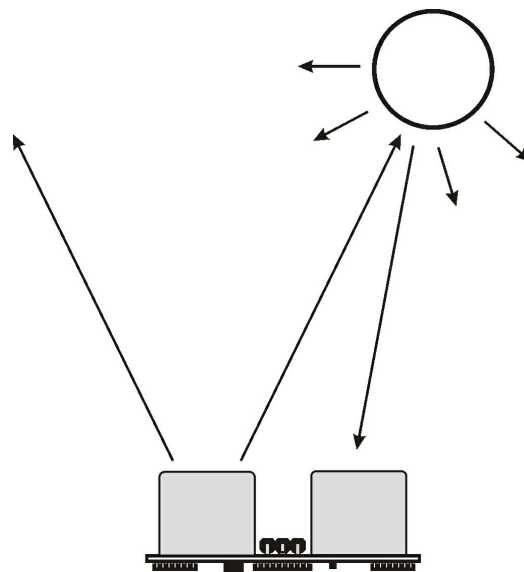
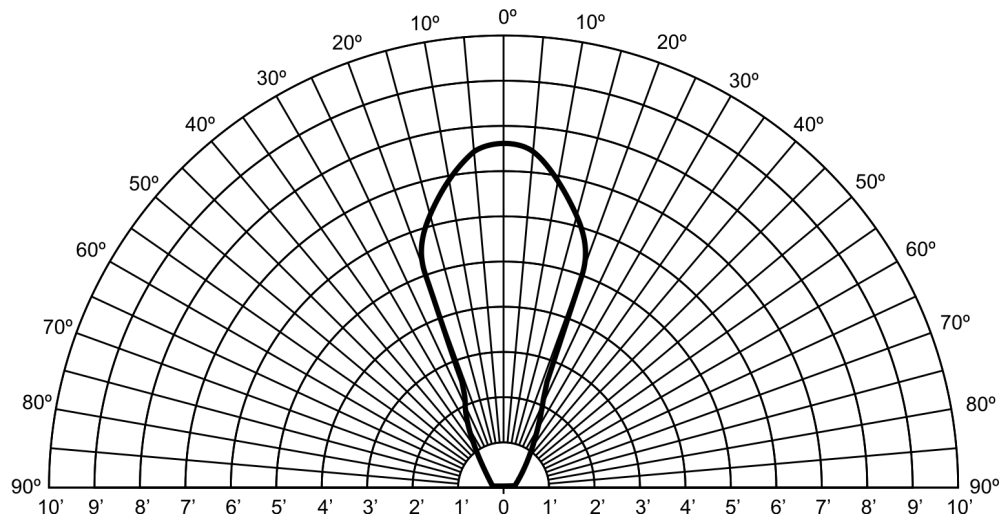
The percent error over the sensor's operating range of 0 to 70 °C is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the example BS2 program given in the Example Programs section below). Percent error and conversion constant calculations are introduced in Chapter 2 of *Smart Sensors and Applications*, a Stamps in Class text available for download from the 28029 product page at www.parallax.com.

Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

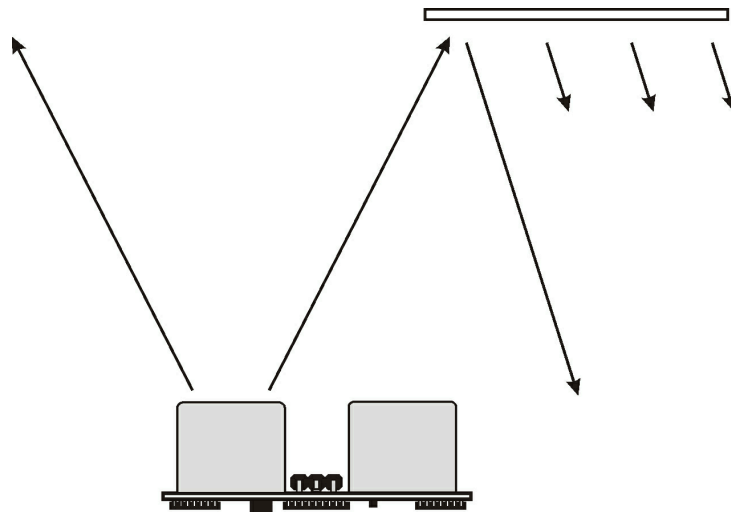
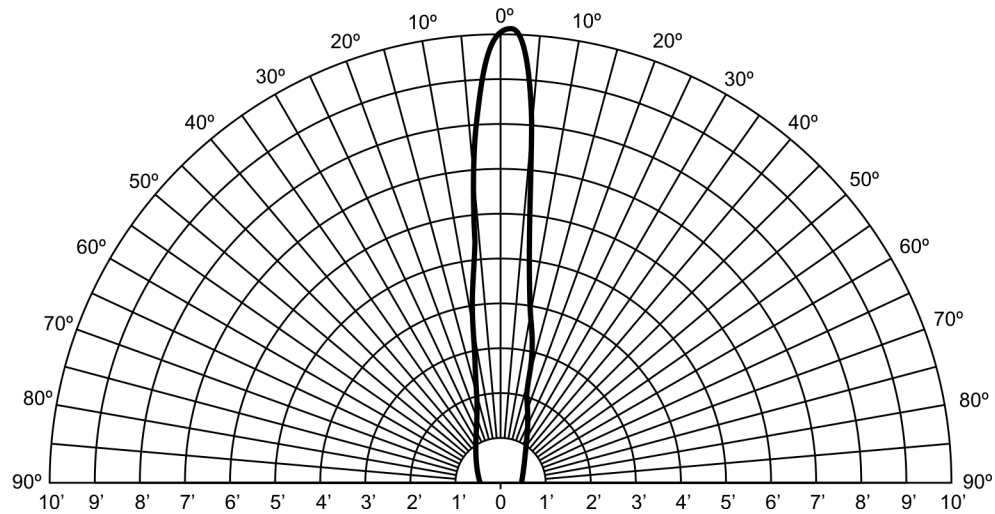
Test 1

Sensor Elevation: 40 in. (101.6 cm)
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation



Test 2

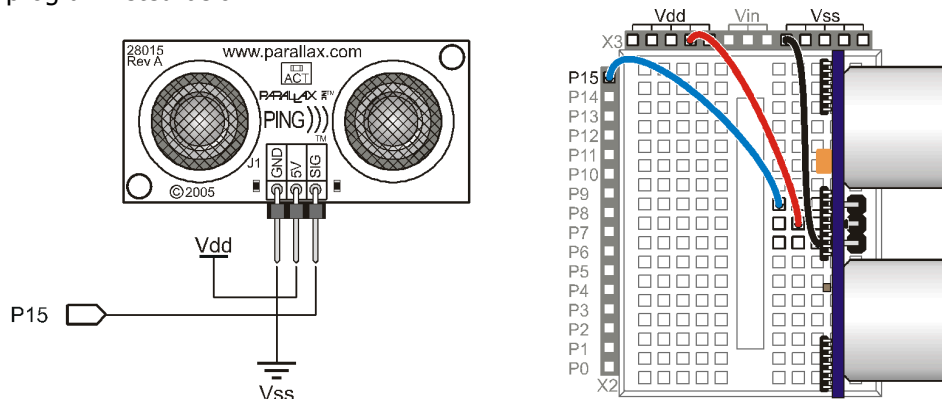
Sensor Elevation: 40 in. (101.6 cm)
Target: 12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole
Target positioned parallel to backplane of sensor



Example Programs

BASIC Stamp 2

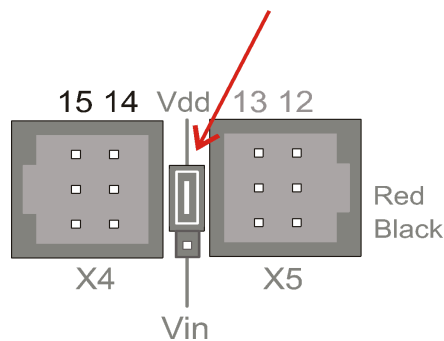
This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp® 2 via the Board of Education® breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example BASIC Stamp program listed below.



Extension Cable and Port Cautions for the Board of Education

If you are connecting your PING))) sensor to a Board of Education platform using an extension cable, follow these steps:

1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown. Then plug the cable into the port, matching the wire color to the labels next to the port.
4. If your Board of Education servo ports do not have a jumper, **do not use them with the PING))) sensor**. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the cable directly to the breadboard with a 3-pin header as shown above. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

Example Program: PingMeasureCmAndIn.bs2

This program for the BASIC Stamp 2 displays distance measurements in both inches and centimeters in the BASIC Stamp Debug Terminal. The example program can be downloaded from the 28015 product page at www.parallax.com. The BASIC Stamp Editor software, which includes the Debug Terminal, is a free download from www.parallax.com/basicstampsoftware.

```
' Smart Sensors and Applications - PingMeasureCmAndIn.bs2
' Measure distance with Ping))) sensor and display in both in & cm

' {$STAMP BS2}
' {$PBASIC 2.5}

' Conversion constants for room temperature measurements.
CmConstant    CON    2260
InConstant     CON    890

cmDistance    VAR    Word
inDistance     VAR    Word
time          VAR    Word

DO

  PULSOUT 15, 5
  PULSIN 15, 1, time

  cmDistance = cmConstant ** time
  inDistance = inConstant ** time

  DEBUG HOME, DEC3 cmDistance, " cm"
  DEBUG CR, DEC3 inDistance, " in"

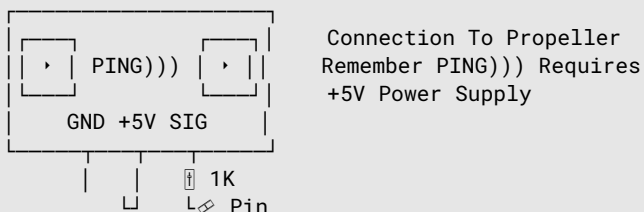
  PAUSE 100

LOOP
```


Propeller Microcontroller

```
{
*****
*      Ping))) PWM Object V1.1      *
*      (C) 2006 Parallax, Inc.      *
* Author:  Chris Savage & Jeff Martin *
* Started: 05-08-2006               *
*****
```

Interface to Ping))) sensor and measure its ultrasonic travel time. Measurements can be in units of time or distance. Each method requires one parameter, Pin, that is the I/O pin that is connected to the Ping)))'s signal line.



```
-----REVISION HISTORY-----
v1.1 - Updated 03/20/2007 to change SIG resistor from 10K to 1K
}}
```

CON

```
TO_IN = 73_746      ' Inches
TO_CM = 29_034      ' Centimeters
```

PUB Ticks(Pin) : Microseconds | cnt1, cnt2

'Return Ping)))'s one-way ultrasonic travel time in microseconds

```
outa[Pin]~          ' Clear I/O Pin
dira[Pin]~~         ' Make Pin Output
outa[Pin]~~         ' Set I/O Pin
outa[Pin]~          ' Clear I/O Pin (> 2 µs pulse)
dira[Pin]~          ' Make I/O Pin Input
waitpne(0, |< Pin, 0) ' Wait For Pin To Go HIGH
cnt1 := cnt         ' Store Current Counter Value
waitpeq(0, |< Pin, 0) ' Wait For Pin To Go LOW
cnt2 := cnt         ' Store New Counter Value
Microseconds := (|(cnt1 - cnt2) / (clkfreq / 1_000_000)) >> 1 ' Return Time in µs
```

PUB Inches(Pin) : Distance

'Measure object distance in inches

```
Distance := Ticks(Pin) * 1_000 / TO_IN      ' Distance In Inches
```

PUB Centimeters(Pin) : Distance

'Measure object distance in centimeters

```
Distance := Millimeters(Pin) / 10          ' Distance In Centimeters
```

PUB Millimeters(Pin) : Distance

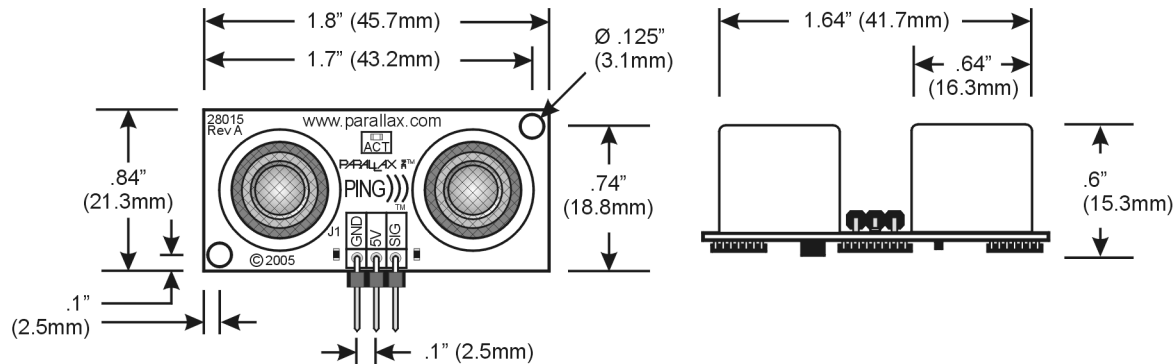
'Measure object distance in millimeters

```
Distance := Ticks(Pin) * 10_000 / TO_CM     ' Distance In Millimeters
```

The ping.spin object is used in an example project with the Parallax 4 x 20 Serial LCD (#27979) to display distance measurements. The complete Project Archive can be downloaded from the Propeller Object Exchange at <https://obex.parallax.com>. Search for object #2290.

When interfacing to a 3.3V microcontroller, such as the Parallax Propeller, it is recommended to add a 1K (or up to 4.7K) series resistor between the PING))) SIG pin and the microcontroller I/O pin.

Dimensions



Resources and Downloads

For additional example code downloads and links to videos, tutorials and robotics projects that use the PING))) Ultrasonic Distance Sensor, visit www.parallax.com and search "28015."

PING))) LCD Demo: <https://obex.parallax.com/obex/ping-demo/>

PING))) Serial Demo: <https://obex.parallax.com/obex/ping-serial-demo/>

The free Propeller Tool programming software can be downloaded from www.parallax.com/propellertool

Product Change Notice

Rev A: original release

Rev B: resonator added to the SX-28 co-processor circuit. No changes to functionality

Rev C: SX-28 co-processor changed to PIC16F57. No changes to functionality.

Rev D: Manufacturing adjustment only. No changes to functionality.

Rev E: Added serial mode. The sensor continues to operate by default in standard PING))) PWM mode, but can be user configured to UART serial mode, which transmits the measurement data at 9600 baud.

Revision History

Version 3.0: Update for release of Ping RevE. Added new UART serial mode protocol information.

Version 2.0: Added revision history. Removed Javelin Stamp examples. Added URLs for programming software. Added Product Change Notice section with PCB revision information.