

超音波レンジファインダー PING)))™ #28015

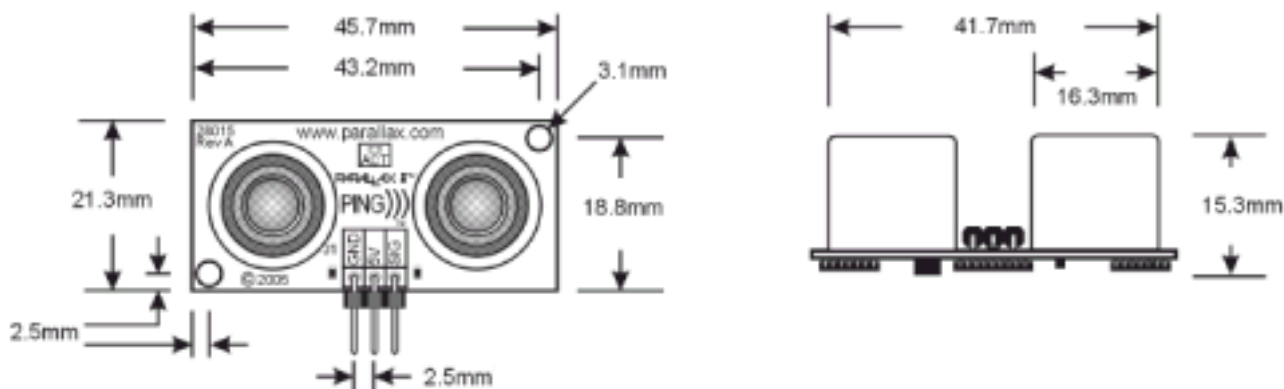
パララックス社の PING)))™ 超音波レンジファインダーは、およそ 3 cm から 3 m までの距離を接触なしで正確に測るものです。ベーシックスタンプ (BASIC Stamp) やジャベリンスタンプ (Javelin Stamp) のマイクロコントローラーに、1 本の I/O ピンだけで簡単に接続が出来ます。

PING センサーは人間の可聴周波数よりはるかに高い超音波のパルスを発射します。そして、前方にある物体に当たり跳ね返ってくる時間を測定する事によって、簡単に距離を計算して出す事が出来ます。

仕様と特長

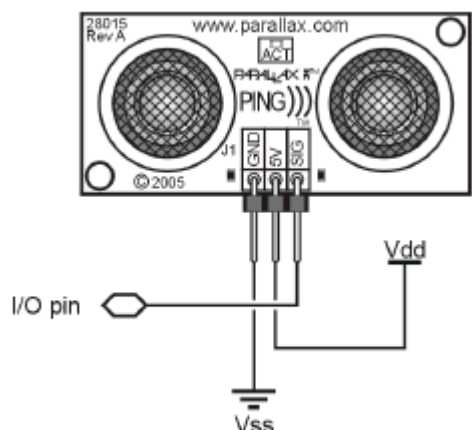
- 測定距離範囲 ; 3 cm から 3.35 m
- 供給電圧 ; 5 V ± 10% (最小電圧 : 4.5V、最大電圧 : 6V)
- 供給電流 ; 30mA (通常) 35mA (最大)
- 入力トリガー ; TTL パルス、2 μsec. (最小) 5 μsec. (通常)
- エコーパルス ; TTL パルス、115 μsec. から 18.5msec.
- 3 ピンインターフェイス (電源プラス用、グランド用、信号用)
- 小さい放射角度
- 単純なパルス入力 / パルス出力による通信
- LED による測定中の表示
- エコーホールドオフ ; トリガーパルスの trailing edge (パルスの後縁) から 350 μsec.
- 発射周波数 ; 200 μsec. の幅で 40KHz

寸法 :



マイクロコントローラーへの接続

PING)))™ センサーには3ピンのヘッダーがついていて、それぞれ電源の5 VDC、グランド、信号用になっています。このヘッダーピンは、ハンダ付の必要のないブレッドボードのようなものに直接差込む事が出来ます。又は、標準のサーボ用の延長ケーブルなどを使う事によって、離れた場所に取り付けることも出来ます。標準的な接続を次に示します。

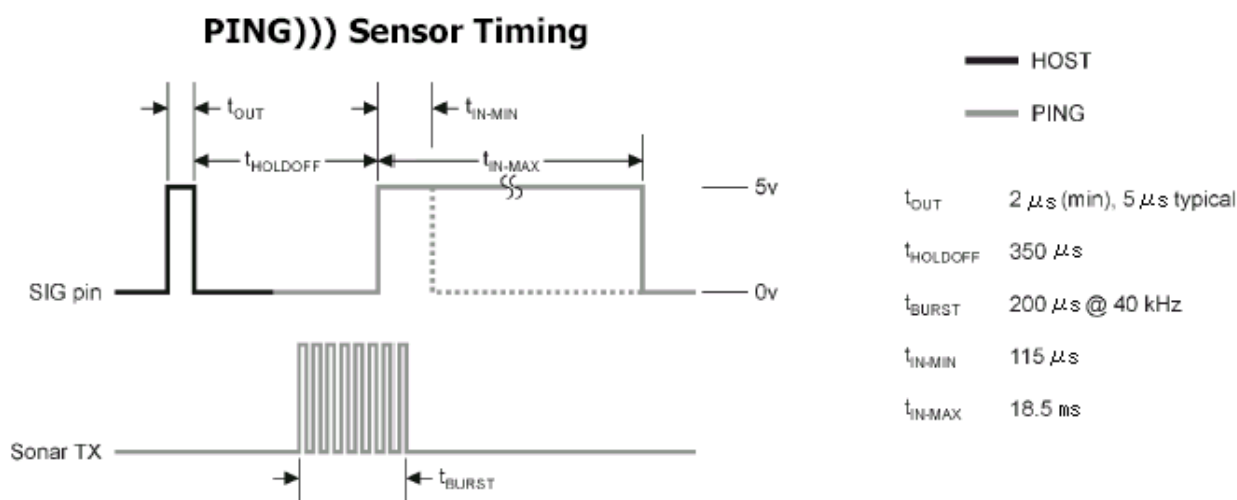


標準的な接続

I/O pin : マイクロコントローラーの I/O ピン
 Vss : 電源のグランド (信号用グランド)
 Vdd : 電源の+ 5 VDC

働き方

Ping センサーは、短い超音波を発射して、その反射の有無を調べる事によって目的物を検出します。ホストマイクロコントローラー（主になるマイクロコントローラー）の制御下（トリガーパルス）で、センサーは短い 40KHz の超音波を発射します。この発射はおよそ秒速 344.42m で空気中を通り抜けます。そして物体に当たるとそれが反射してセンサーに戻ってきます。Ping センサーは、反射してきたものを検出した時、ホストのマイクロコントローラーに出力パルスを送ります。それ故、パルス幅が目的物との距離に相当します。



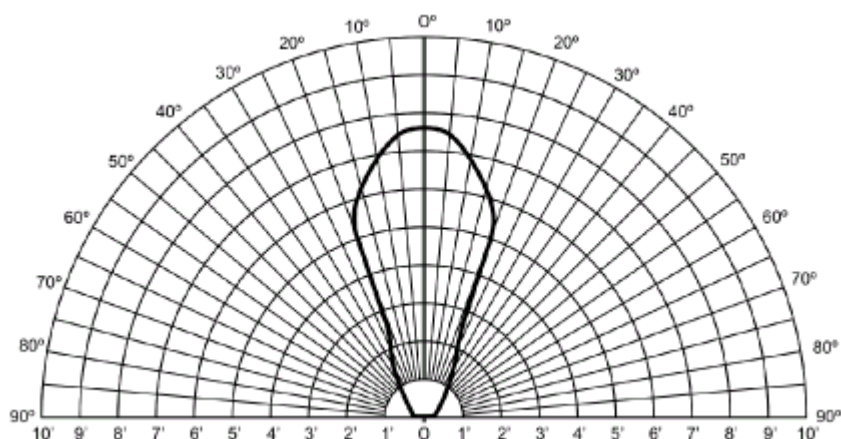
テストデータ

次のテストデータは、Ping))) センサーを使い、ベーシックスタンプのマイクロコントローラーにモジュールを接続し、パララックス社のラボでテストしたものです。テスト面は、リノリウム床でデータに部屋の床からの反射が最小になるように持ち上げられています。全てのテストは、環境の変化がないように室内で通常の室温で実施されました。目標物は常に Ping))) センサーと同じ様に持ち上げられて中心に置かれています。

テスト 1

センサーの高さ：101.6 cm (40 inch)

目標物：直径 8.9 cm (3.5 inch) シリンダー状で 121.9 cm (4 ft) の長さで真っ直ぐに立っている。

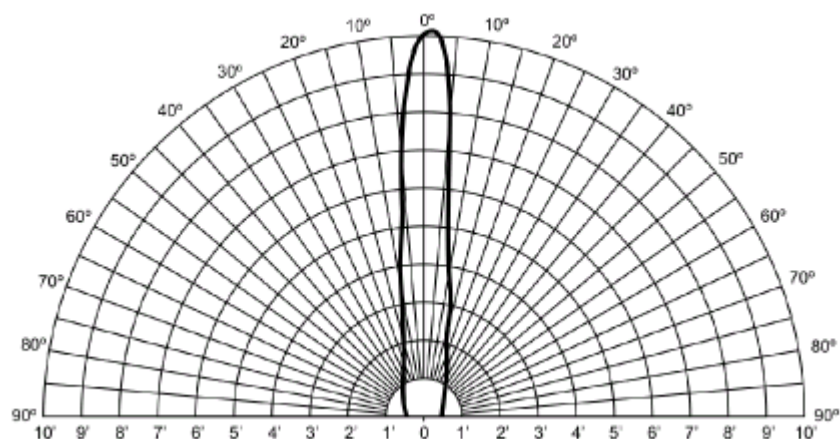


テスト 2

センサーの高さ：101.6 cm (40 inch)

目標物：30.5 cm x 30.5 cm (12 in. x 12 in.) のダンボールで、2.5 cm の棒に取り付けたもの。

目標物は、センサーの取付け板に平行に置かれたもの。



プログラム例：ベーシックスタンプ2 マイクロコントローラー

これから説明するプログラム例は、ベーシックスタンプ2 マイクロコントローラー(以下 BS2 という)で、PING)))センサーを使うためのものです。接続したモジュールに適した条件付編集技術を使う事によって、このプログラムは BS2 モジュールのどの種類でも働きます。

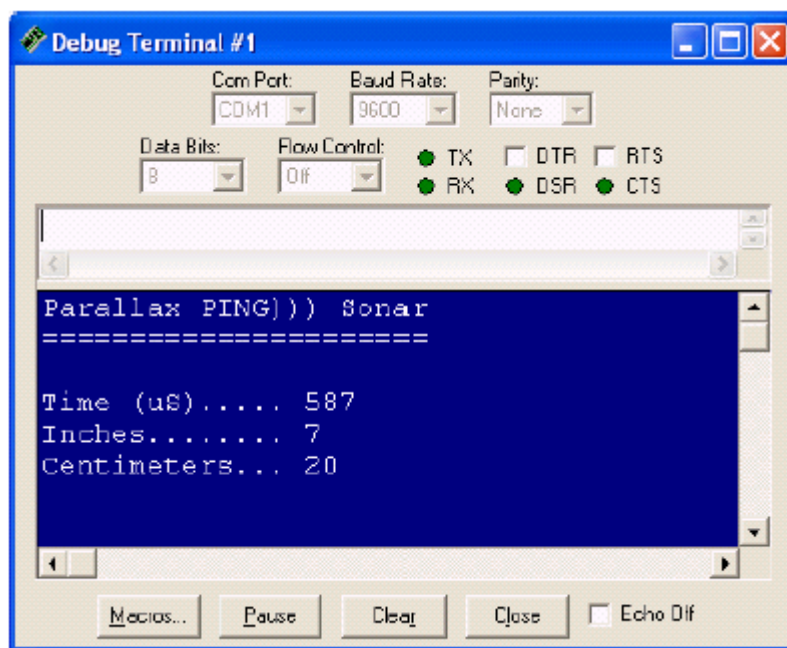
このプログラムの中心は、**Get_Sonar** サブルーティンです。このサブルーティンは、0 (ゼロ) を I/O に設定したピンに出力とする事から始まります。0 (ゼロ、又はロー); これは PING)))センサーをトリガーするのに必要な、ロー、ハイ、ロー (low - high - low ; 0 - 1 - 0) の完全な **PULSOUT** を発生させるものです。(I/O ピンがどのような状態か分からないため、0 (ロー) にします。)トリガーパルスが終わった後、センサーは超音波を発射する前に約 200 μ 秒の間、待ちます。これは、BS2 に次のインストラクションの準備をさせるためです。インストラクション **PULSIN** は、目標物までの距離に相当するパルス幅の 1 (ハイ) の状態の時間を測定するために使われます。

PULSIN で得られた値 (raw value) は、BS2 モジュールの種類によって分解度が異なりますので、そのモジュールに適したものに目盛り変換しなければいけません。得られた値がマイクロ秒 (μ 秒) に変換された後パルスは往復しているためそれを 2 で割らなければいけません。その rawDist (Word 変数) に保存された値は、マイクロ秒で表された目標物までの距離ということになります。

マイクロ秒からセンチメートルに変換するには簡単な数学で出来ます。一般に言われている音速は気温 21 の時、毎秒 1130 フィート (約 344.42 m) です。(音速: 大気中では、気温 0 、1 気圧で秒速 331.45 m で気温 1 上昇する毎に 0.61m ずつ増す。広辞苑より) 従って、1 秒間に 13,560 インチ (約 34442.4 cm) ですから、1 インチ (2.54 cm) にかかる時間は、73.746 μ 秒となります。どのように得られた値を、フローティングポイント数値 (floating-point number) の 73.746 で割り算をしたら良いのでしょうか?

73.746 で除算する代わりに、0.01356 で乗算する事が出来ます。ベーシックスタンプ利用に慣れていない方は、ジレンマに感じるかもしれませんが、特別なオペレータ「**」というものが用意されていて、このような計算が簡単に出来るようになっています。「**」オペレータは、1/65,536 の単位で数値を掛ける事が出来ます。オペレータ「**」のパラメータを見つけ、それから 0.01356 に 65,536 を掛け合わせます。結果は、888.668 ですが、私たちは 889 と四捨五入した結果を使います。

センチメートルに変換するには、上と同じ様な方法を使います。そのプログラムの結果を次に示します。



```

=====
'
' File..... Ping Demo.BS2
' Purpose.... Demo Code for Parallax Ping Sonar Sensor
' Author..... Parallax, Inc.
' E-mail..... support@parallax.com
' Started....
' Updated.... 03 FEB 2005
'
' {$STAMP BS2}
' {$PBASIC 2.5}
'
=====

' -----[ Program Description ]-----
'
' This program demonstrates the use of the Parallax PING))) sensor and then
' converting the raw measurement to English (inches) and Metric (cm) units.
'
' Sonar Math:
'
' At sea level sound travels through air at 1130 feet per second. This
' equates to 1 inch in 73.746 uS, or 1 cm in 29.034 uS).
'
' Since the PING))) sensor measures the time required for the sound wave to
' travel from the sensor and back. The result -- after conversion to
' microseconds for the BASIC Stamp module in use -- is divided by two to
' remove the return portion of the echo pulse. The final raw result is
' the duration from the front of the sensor to the target in microseconds.
'
' -----[ I/O Definitions ]-----

Ping          PIN      15
    
```

```
' -----[ Constants ]-----
#SELECT $STAMP
#CASE BS2, BS2E
  Trigger    CON    5          ' trigger pulse = 10 uS
  Scale      CON    $200       ' raw x 2.00 = uS
#CASE BS2SX, BS2P
  Trigger    CON    13
  Scale      CON    $0CD       ' raw x 0.80 = uS
#CASE BS2PE
  Trigger    CON    5
  Scale      CON    $1E1       ' raw x 1.88 = uS
#ENDSELECT

RawToIn     CON    889         ' 1 / 73.746 (with **)
RawToCm     CON    2257        ' 1 / 29.034 (with **)

IsHigh      CON    1          ' for PULSOUT
IsLow       CON    0

' -----[ Variables ]-----

rawDist     VAR    Word        ' raw measurement
inches      VAR    Word
cm          VAR    Word

' -----[ Initialization ]-----

Reset:
  DEBUG CLS,
    "Parallax PING))) Sonar", CR,          ' setup report screen
    "===== ", CR,
    CR,
    "Time (uS)..... ", CR,
    "Inches..... ", CR,
    "Centimeters...  "

' -----[ Program Code ]-----

Main:
  DO
    GOSUB Get_Sonar          ' get sensor value
    inches = rawDist ** RawToIn ' convert to inches
    cm = rawDist ** RawToCm   ' convert to centimeters

    DEBUG CRSRXY, 15, 3,          ' update report screen
      DEC rawDist, CLREOL,
      CRSRXY, 15, 4,
      DEC inches, CLREOL,
      CRSRXY, 15, 5,
      DEC cm, CLREOL

    PAUSE 100
  LOOP
END
```

```
' -----[ Subroutines ]-----
'
' This subroutine triggers the Ping sonar sensor and measures
' the echo pulse.  The raw value from the sensor is converted to
' microseconds based on the Stamp module in use.  This value is
' divided by two to remove the return trip -- the result value is
' the distance from the sensor to the target in microseconds.

Get Sonar:
Ping = IsLow           ' make trigger 0-1-0
PULSOUT Ping, Trigger ' activate sensor
PULSIN Ping, IsHigh, rawDist ' measure echo pulse
rawDist = rawDist */ Scale ' convert to uS
rawDist = rawDist / 2     ' remove return trip
RETURN
=====

'
' File..... Ping Demo.BS1
' Purpose.... Demo Code for Parallax Ping Sonar Sensor
' Author..... Parallax, Inc.
' E-mail..... support@parallax.com
' Started....
' Updated.... 03 FEB 2005
'
' {$STAMP BS1}
' {$PBASIC 1.0}
'
' =====

' -----[ Program Description ]-----
'
' This program demonstrates the use of the Parallax PING))) sensor and then
' converting the raw measurement to English (inches) and Metric (cm) units.
'
' Sonar Math:
'
' At sea level sound travels through air at 1130 feet per second.  This
' equates to 1 inch in 73.746 uS, or 1 cm in 29.034 uS).
'
' Since the PING))) sensor measures the time required for the sound wave to
' travel from the sensor and back.  The result -- after conversion to
' microseconds for the BASIC Stamp module in use -- is divided by two to
' remove the return portion of the echo pulse.  The final raw result is
' the duration from the front of the sensor to the target in microseconds.

' -----[ I/O Definitions ]-----

SYMBOL Ping           = 7

' -----[ Constants ]-----

SYMBOL Trigger       = 1           ' 10 uS trigger pulse
SYMBOL Scale         = 10          ' raw x 10.00 = uS

SYMBOL RawToIn       = 889         ' 1 / 73.746 (with **)
SYMBOL RawToCm       = 2257        ' 1 / 29.034 (with **)

SYMBOL IsHigh        = 1           ' for PULSOUT
SYMBOL IsLow         = 0
```

```
' -----[ Variables ]-----
SYMBOL rawDist      = W1          ' raw measurement
SYMBOL inches       = W2
SYMBOL cm           = W3

' -----[ Program Code ]-----

Main:
  GOSUB Get Sonar          ' get sensor value
  inches = rawDist ** RawToIn ' convert to inches
  cm = rawDist ** RawToCm   ' convert to centimeters

  DEBUG CLS              ' report
  DEBUG "Time (uS)..... ", #rawDist, CR
  DEBUG "Inches..... ", #inches, CR
  DEBUG "Centimeters... ", #cm

  PAUSE 500
  GOTO Main

END

' -----[ Subroutines ]-----

' This subroutine triggers the Ping sonar sensor and measures
' the echo pulse.  The raw value from the sensor is converted to
' microseconds based on the Stamp module in use.  This value is
' divided by two to remove the return trip -- the result value is
' the distance from the sensor to the target in microseconds.

Get Sonar:
  LOW Ping          ' make trigger 0-1-0
  PULSOUT Ping, Trigger ' activate sensor
  PULSIN Ping, IsHigh, rawDist ' measure echo pulse
  rawDist = rawDist * Scale ' convert to uS
  rawDist = rawDist / 2 ' remove return trip
  RETURN
```


プログラム例：ジャベリンスタンプマイクロコントローラー (Javelin Stamp Microcontroller)

このクラス (class) は PING))) センサー使用のために幾つかの方法を実施します。

```
package stamp.peripheral.sensor;

import stamp.core.*;

/**
 * This class provides an interface to the Parallax PING))) ultrasonic
 * range finder module.
 * <p>
 * <i>Usage:</i><br>
 * <code>
 *   Ping range = new Ping(CPU.pin0);           // trigger and echo on P0
 * </code>
 * <p>
 * Detailed documentation for the PING))) Sensor can be found at: <br>
 * http://www.parallax.com/detail.asp?product_id=28015
 * <p>
 *
 * @author Jon Williams, Parallax Inc. (jwilliams@parallax.com)
 * @version 1.0 03 FEB 2005
 */
public final class Ping {

    private int ioPin;

    /**
     * Creates PING))) range finder object
     *
     * @param ioPin PING))) trigger and echo return pin
     */
    public Ping (int ioPin) {
        this.ioPin = ioPin;
    }

    /**
     * Returns raw distance value from the PING))) sensor.
     *
     * @return Raw distance value from PING)))
     */
    public int getRaw() {

        int echoRaw = 0;

        CPU.writePin(ioPin, false);           // setup for high-going pulse
        CPU.pulseOut(1, ioPin);              // send trigger pulse
        echoRaw = CPU.pulseIn(2171, ioPin, true); // measure echo return

        // return echo pulse if in range; zero if out-of-range
        return (echoRaw < 2131) ? echoRaw : 0;
    }
}
```

```

/*
 * The PING))) returns a pulse width of 73.746 uS per inch. Since the
 * Javelin pulseIn() round-trip echo time is in 8.68 uS units, this is the
 * same as a one-way trip in 4.34 uS units. Dividing 73.746 by 4.34 we
 * get a time-per-inch conversion factor of 16.9922 (x 0.058851).
 *
 * Values to derive conversion factors are selected to prevent roll-over
 * past the 15-bit positive values of Javelin Stamp integers.
 */

/**
 * @return PING))) distance value in inches
 */
public int getIn() {
    return (getRaw() * 3 / 51);           // raw * 0.058824
}

/**
 * @return PING))) distance value in tenths of inches
 */
public int getIn10() {
    return (getRaw() * 3 / 5);           // raw / 1.6667
}

/*
 * The PING))) returns a pulse width of 29.033 uS per centimeter. As the
 * Javelin pulseIn() round-trip echo time is in 8.68 uS units, this is the
 * same as a one-way trip in 4.34 uS units. Dividing 29.033 by 4.34 we
 * get a time-per-centimeter conversion factor of 6.6896.
 *
 * Values to derive conversion factors are selected to prevent roll-over
 * past the 15-bit positive values of Javelin Stamp integers.
 */

/**
 * @return PING))) distance value in centimeters
 */
public int getCm() {
    return (getRaw() * 3 / 20);         // raw / 6.6667
}

/**
 * @return PING))) distance value in millimeters
 */
public int getMm() {
    return (getRaw() * 3 / 2);         // raw / 0.6667
}
}

```

この簡単なデモは、ジャベリンスタンプでPING)))超音波レンジファインダークラスの使い方を示しています。

```
import stamp.core.*;
import stamp.peripheral.sensor.Ping;

public class testPing {

    public static final char HOME = 0x01;

    public static void main() {

        Ping range = new Ping(CPU.pin0);
        StringBuffer msg = new StringBuffer();

        int distance;

        while (true) {
            // measure distance to target in inches
            distance = range.getIn();

            // create and display measurement message
            msg.clear();
            msg.append(HOME);
            msg.append(distance);
            msg.append(" \n");
            System.out.print(msg.toString());

            // wait 0.5 seconds between readings
            CPU.delay(5000);
        }
    }
}
```