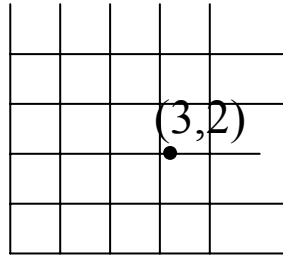


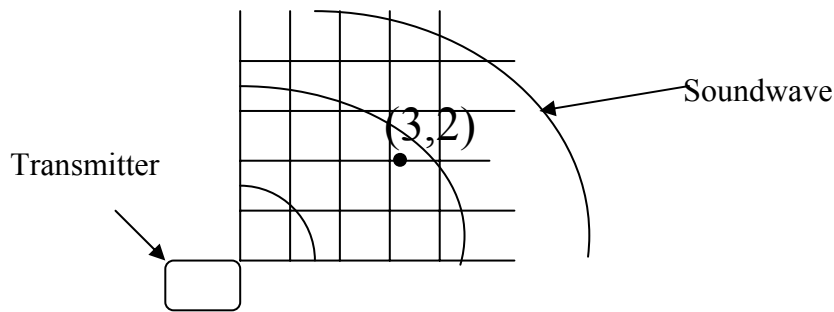
## Using The Ultrasonic Sensor to Determine Location By William Hamlet and William Kusewich

### Introduction

A coordinate axis can be used to represent position on a flat surface. The two axis are the x-axis (representing horizontal position) and the y-axis (representing vertical position). An objects position on a surface may be represented in terms of x and y points. For example, an object may be located at position (3,2) of a flat surface.



A transmitter may be placed at the ordinate (0,0) to determine the distance between the object and the transmitter.



Sound waves may be used as transmitters. Sound waves are formed from the vibration of air molecules and they have all the typical properties of wave (i.e. reflection, refraction, diffraction, etc.). The speed of sound in air is a constant,  $v$ . A formula may be used to calculate the speed of sound in air,

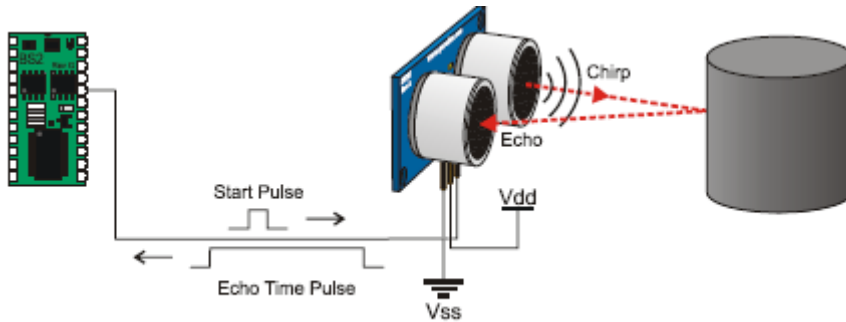
$$\text{speed of sound} = \text{distance traveled} / \text{time elapsed}$$

If you know how long it takes for a wave to travel to an object and reflect back to the starting position, then the distance the wave traveled is

$$\begin{aligned} \text{Distance traveled} &= \text{time elapsed} * \text{speed of sound} \\ D &= \Delta t * v \end{aligned}$$

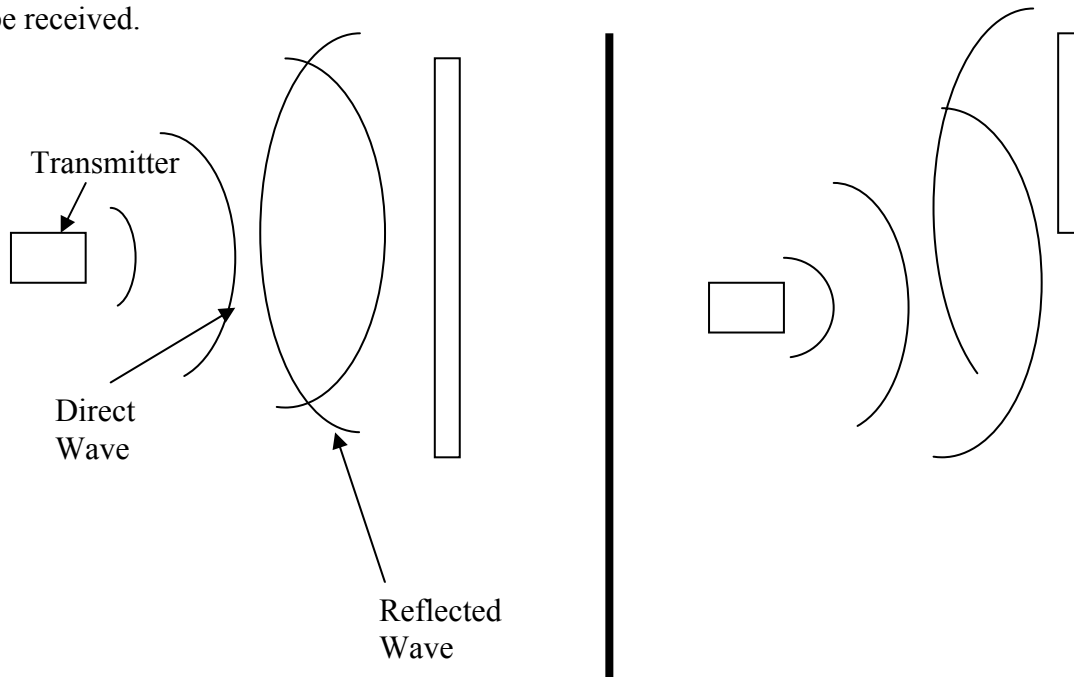
### Primary Objective

The primary objective of this project is to determine an objects position on a coordinate axis using two ultrasonic sensors. An ultrasonic sensor is used to send out a high frequency sound waves and record the time it takes for the reflected sound to return. The sensor then uses the time to calculate the distance the object is from the sensor.



### Background

There are many practical reasons why you would use sensors to locate an object. Most GPS (global positioning systems) use sensors to track objects on the ground. The object may be car, a package, a ship at sea, etc. The Ping ultrasonic sensor uses sound wave to find the distance of an object. When the sound wave hits an object, it reflects off the object. The object may be directly in front of the object or at an angle for the signal to be received.



Direct Reflection

Reflection at an Angle

## **Equipment List**

The primary pieces of equipment are listed below:

1. Ping Ultrasonic Sensor
2. Board of Education circuit boards
3. BASIC stamp microcontroller
4. Metal Stands
5. Servo Motors

The metal stands were constructed out of steel located in the

### ***Ping Ultrasonic Sensor***

The BASIC stamp microcontroller is used to emit a short ultrasonic burst (40kHz for 2μs). The speed of sound is 344.8m/s. Using equation 1,

$$D = (344.8\text{m/s}) * \Delta t$$

Converting micro-seconds into seconds and meters into centimeters, the distance equation becomes,

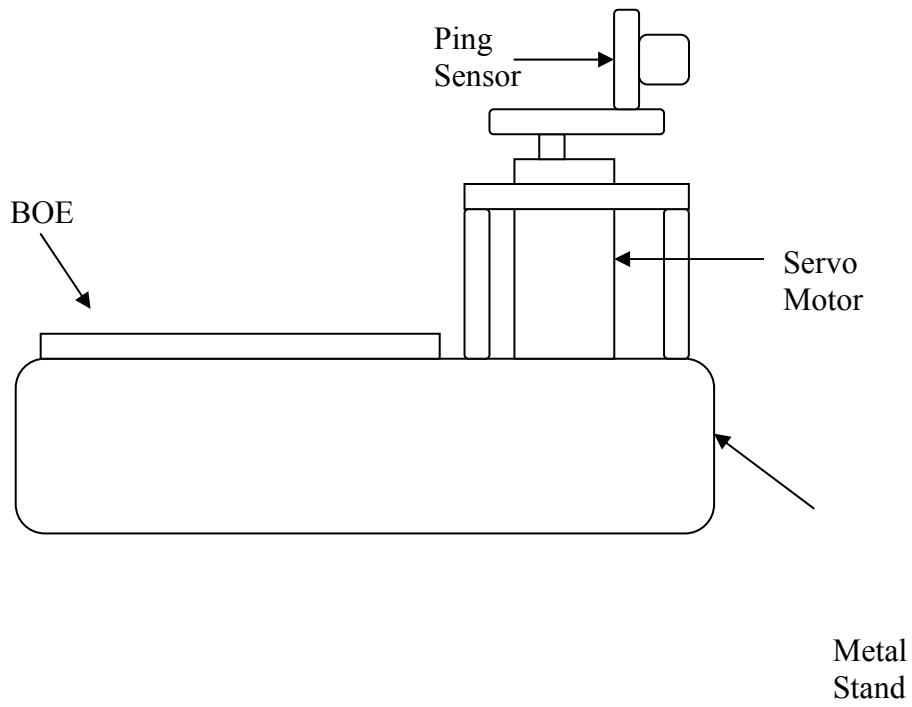
$$D(\text{cm}) = .03448 * \Delta t$$

The ultrasonic sensor cannot detect object more than three meters away.

## **Equipment Diagram**

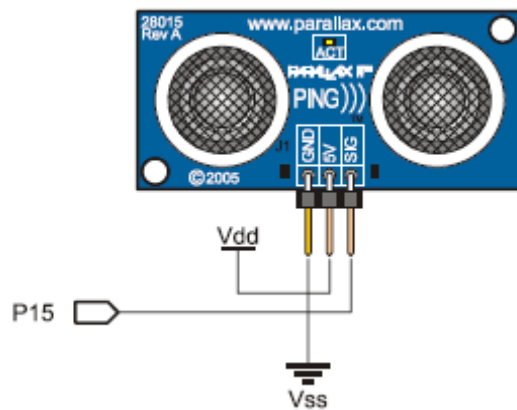
The ultrasonic sensor is placed on a servo motor that rotates in a 90 span. When the object is detected the servo motor stops moving, and the ultrasonic sensor measure the distance from the object. The following picture and diagram shows how the sensor is mounted on the

stand.

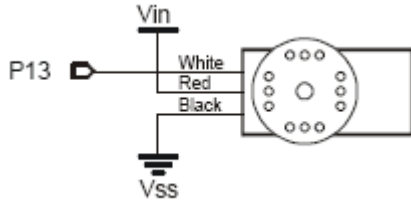


The servo motor and Ping sensor may be attached the board of education using the following schematic.

Ping Connection

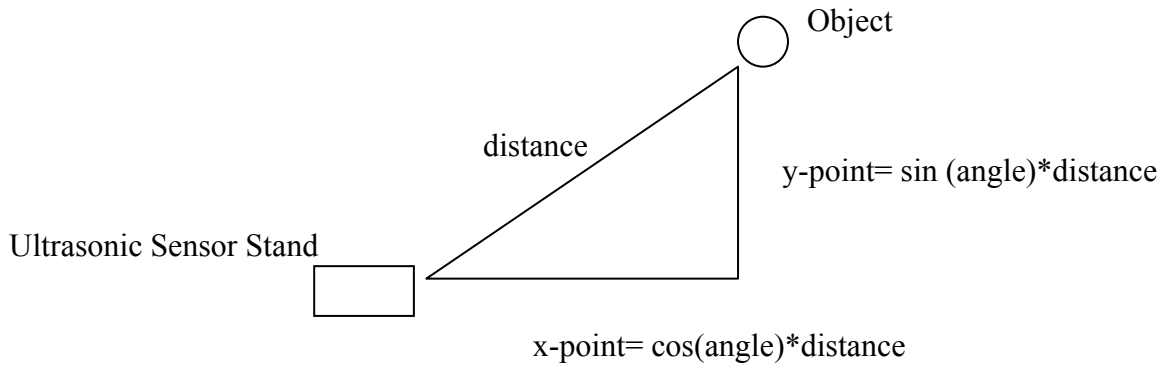


Servo Motor Connection

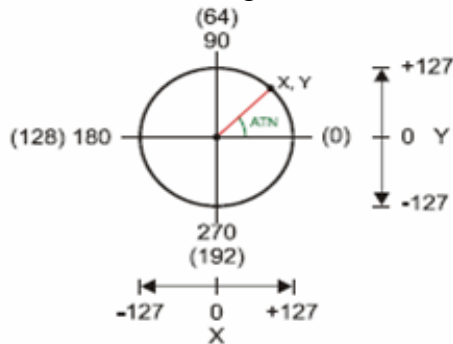


### Experimental Procedure

An one meter by one meter coordinate grid was made, with each square in the grid measuring 10cm by 10cm. The metal stands holding the servo motors and Ping sensor were placed on the ends of the coordinate grid. A wooden block measuring 5cm by 5cm by 10cm was placed at one of the points on the coordinate grid. The servo motors were programmed to turn until the Ping sensor detected the wooden block. When the Ping sensor detected the block, it measured the distance it was from the block. Using the sine and cosine, the x and y points were determined on the coordinate grid.



The pBASIC program language uses the unit brad instead of angle. In pBASIC 180 degrees is equal to 128 brads, and 360 degrees is equal to 256 brads. This scale is used because pBASIC can hold only 256 units of memory. The following diagram shows how brads are related to degrees on the unit circle.



The following program was used to turn the Servo and detect the block.

## ***Computer Program***

```
' Summit Project: Finding x-y coordinates using distance input from an
' ultrasonic device.
' Test the servo at three different position signals.
' {$STAMP BS2}
' {$PBASIC 2.5}
counter VAR Word
cmDistance = 120 'Initialize the length to be greater than what we
                  'are looking for.

HIGH 13
PAUSE 200

SEROUT 13, 84,[22, 12]
DEBUG "Position counter clockwise to 9 o'clock or 0 degrees. ",CR

Pulses VAR Word
Pulses = 826 ' 672 initial value of pulses
CmConstant CON 2000 'should be 2260
InConstant CON 890
cmDistance VAR Word
cmDistance = 120
inDistance VAR Word
time VAR Word
brads VAR Word
X VAR Word
Y VAR Word
brads = 22 ' 1 is normal starting point binary radians
```

PULSOUT 0, Pulses ' signal to servo pulses determine servo pos
PAUSE 20
DO WHILE ((Pulses < 1200) AND (cmDistance > 95)) 'can not do row 9
Pulses = Pulses + 22 'increment the pulses

PULSOUT 1,5 'ultrasound pin 1 signal out
PULSIN 1,1, time
cmDistance = (CmConstant ** time) '+ 21 add to correct for length
inDistance = InConstant ** time
'increment the brads
Y = (SIN(brads)* cmDistance) /120 'should divide by 128 but not
accurate
Y = Y + 5 'no rounding up
Y = Y / 10
IF brads < 18 THEN
X = (COS(brads)*cmDistance)/ 120 'should be 128 but results not
calibrated
X = X + 3 'rounding but it raises 1 too much
X = X / 10
ELSE
X = (COS(brads)*cmDistance)/ 110 'should be 128 but results not
calibrated
X = X + 5 'rounding but it raises 1 too much
X = X / 10
X = X - 1
Y = (SIN(brads)* cmDistance) /110 'should divide by 128 but not
accurate
Y = Y + 5 'no rounding up
```

```

    Y = Y / 10
    IF (brads > 34) THEN
        X = (COS(brads)*cmDistance)/ 110 'should be 128 but results
not calibrated
        X = X + 5 'rounding but it raises 1 too much
        X = X / 10
        X = X
        Y = (SIN(brads)* cmDistance) /110 'should divide by 128 but
not accurate
        Y = Y + 5 'no rounding up
        Y = Y / 10
        Y = Y - 1
    ENDIF

ENDIF

DEBUG CR, " cmDistance = ", DEC3 cmDistance
DEBUG " brads = ", DEC3 brads," X = ", DEC X, " Y = ",DEC3
Y, CR
DEBUG " time = ", DEC5 time
PAUSE 50
DEBUG " Pulses = ", DEC5 Pulses, CR
FOR counter = 1 TO 150
    PULSOUT 0, Pulses ' signal to servo pulses determine servo pos
    PAUSE 20
NEXT
SEROUT 2, 84, [128, "Hypenuse = ",DEC3 cmDistance]
SEROUT 2, 84, [148, "X = ", DEC3 X, " Y ",DEC3 Y]
'PAUSE 20
brads = brads + 3 'change for test
LOOP
END

```

'Program that measures cmDistance in cm

```

'DO
'PULSOUT 9, 5
'PULSIN 9, 1, time
'cmDistance = CmConstant ** time
'DEBUG HOME, DEC3 time, CR
'DEBUG DEC3 cmDistance, " cm"
'PAUSE 100
'LOOP

```

## Results

The following table shows the results of testing the ultrasonic sensor using a wooden cylinder. A dark circle represents has been accurately detected with the proper x-y coordinates given. A clear circle represents the object was not detected or gave incorrect x-y coordinates. The table indicates the ultrasound sensor was completely accurate for 70% of the measurements. When the results are considered for trials that are within one unit of the location tested (i.e. the gray circles added to the black circles) the results are accurate 88% of the time.

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Y=8 |     |     |     |     |     |     |     |     |     |
| Y=7 | ●   | ●   | ●   | ●   | ○   | ●   | ○   | ○   | ○   |
| Y=6 | ●   | ●   | ●   | ●   | ●   | ●   | ●   | ●   | ○   |
| Y=5 | ●   | ●   | ●   | ●   | ●   | ○   | ●   | ●   | ●   |
| Y=4 |     | ●   | ●   | ●   | ●   | ●   | ○   | ●   | ●   |
| Y=3 |     | ●   | ●   | ●   | ●   | ●   | ○   | ○   | ○   |
| Y=2 |     | ●   | ●   | ●   | ●   | ●   | ○   | ○   | ○   |
| Y=1 |     | ●   | ●   | ●   | ●   | ●   | ○   | ○   | ○   |
| Y=0 |     | ●   | ●   | ●   | ●   | ●   | ○   | ○   | ○   |
|     | X=0 | X=1 | X=2 | X=3 | X=4 | X=5 | X=6 | X=7 | X=8 |

## Future Works

- ◆ Create another unit, and have both units move along their respective axes until the object is located.
- ◆ Place sensors at end of x-y coordinates respectively. Solve two equations using the distance formula simultaneously for x and y.
- ◆ Have locators find a moving object.