Lab Exercise 7: Ultrasonic Sensors

The objective of this lab is to provide you with the experience of using an ultrasonic sensor. Ultrasonic sensors, or sonar sensors, are used in a wide range of applications involving range finding and object detection and avoidance. For example, Polaroid Corporation used sonar sensors in their auto-focusing cameras to detect the distances of objects for focusing. Sonar sensors are excellent sensors to use for mobile robot applications. If a robot needs to navigate through a room filled with obstacles, then it can do it successfully by employing sonar sensors.

How Sonar Sensors Work

Figure 1 below shows a simple schematic of a generic sonar sensor in action. An ultrasonic burst of energy is emitted from the transducer. This is known as a *ping*. The sound waves travel until reflected off of an object. The echoed sound wave returns to the transducer. The echo may be of smaller amplitude, but the carrier frequency should be the same as the ping. An external timer records the time of flight (the time that the sound waves take to travel to and from the object), which can be converted to distance when considering the speed of sound in air. As the transmitted sound waves propagate from the transducer, they spread over a greater range. In other words, the sound waves propagate from the transducer in the shape of a cone of angle θ.

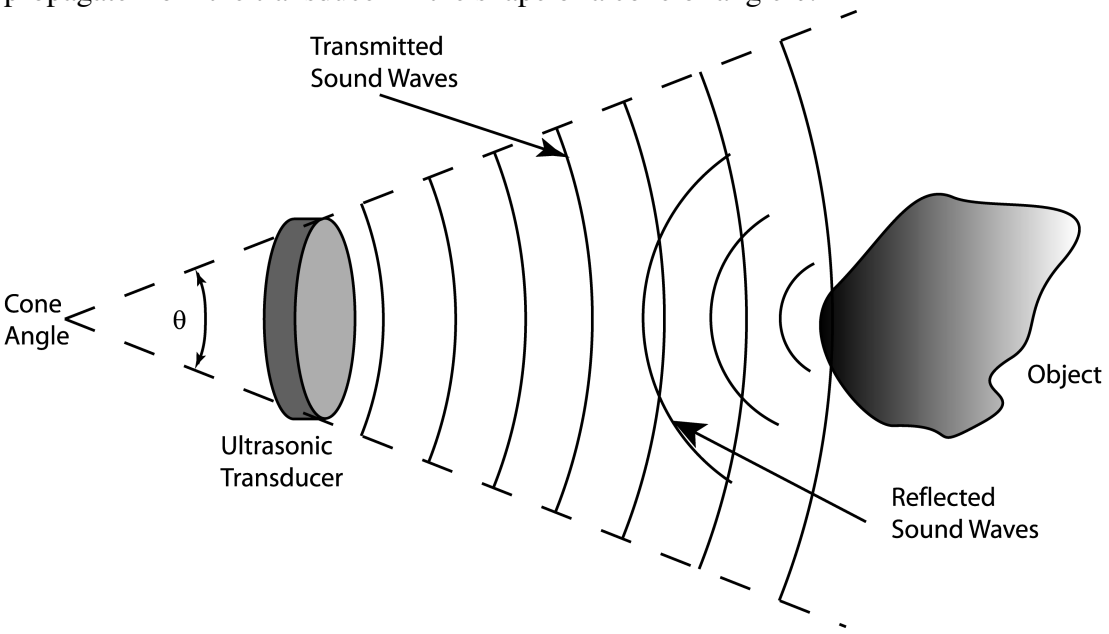

Figure 1. Functionality of a generic sonar sensor.

Limitations of Sonar Sensors

Sonar sensors are not ideal devices. They are limited to resolution, range, and the size of object they can detect. The external timing circuits of some sonar sensor systems are subject to false echoes. Values returned by the sensor may not match the actual distance of the object. One solution is to take an average of your readings. For example, ping three times and take an average. This method seems to reduce the effects of false triggers.

The Devantech SRF04 Ultrasonic Range Finder

The sonar we will be using in the lab and on the project is the Devantech SRF04 Ultrasonic Range Finder. An important feature that distinguishes itself from the generic sonar sensor in Figure 1 is that it has a separate transmitter and receiver. This allows for the smallest minimum detectable distance. The frequency of the ping is 40 kHz. The reason for a 40 kHz frequency sound wave is to reduce the chances of false echoes. For example, it is unlikely that a 40 kHz sound wave will come from any other source other than the actual ultrasonic sensor itself. The external timing circuit looks for a 40 kHz return signal to identify it as an echo. An advantage of the SRF04 (over the Polaroid units) is that the frequency generating circuit is integral to the printed circuit board supporting the transducers. This minimizes the amount of wiring needed to utilize the sensor. More information on the SRF04 can be found by clicking on the links on the class web page associated with this lab exercise, which take you to www.acroname.com, distributor of the SRF04, as well as many other useful robotics related devices. Following are direct links to the SRF04 information pages:

http://www.acroname.com/robotics/parts/R93-SRF04.html
http://www.acroname.com/robotics/info/examples/srf04-3/srf04-3.html

Laboratory Exercise

Equipment Needed: Handy Board microcontroller and ultrasonic sensor.

1. Interface the sonar sensor to the Handy Board as shown by Figure 2 below. Turn the Handy Board on and connect it to the computer. Start Interactive C (IC), noticing that SONAR.C is loaded with IC. SONAR.C sends a ping and measures the time to return. An integer is returned, which is the time of flight in internal clock ticks. This is the number of CPU clock cycles (or ticks) that pass while the signal travels to the object and then returns. In interactive mode, try typing `printf("%d\n",sonar());` at the prompt several times to see this.

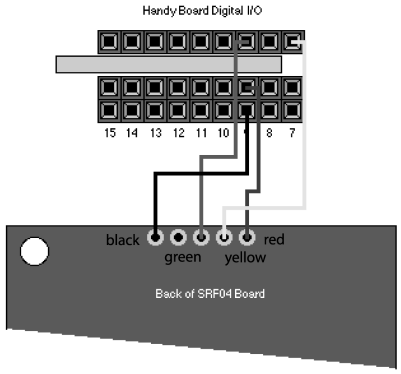

Figure 2. Interfacing the SRF04 with the Handy Board.
Figure used with permission from Acroname
(http://www.acroname.com/robotics/info/examples/srf04-3/srf04-3.html)

2. Before the sonar can be used, the Handy Board must first be initialized to work with the unit. This is accomplished by calling the function `void sonar_init()`. In interactive mode, try typing `sonar_init()`; once, and then type the command `printf("%d/n",sonar())`; at the prompt several times to see how the sonar works. Notice that the returned integer is proportional to the distance to the nearest object as you move the object or sonar around.
3. write the following program to average the time of flight readings.
4. Use the program in figure 3 to measure the distance to an object (such as a book) at several locations orthogonal to the sonar unit. Record the data and use it to determine a calibration coefficient that correlates clock ticks with centimeters using a linear regression (hint: use an x-y chart with a trend line in EXCEL or the polyfit command in MATLAB). What is your calibration coefficient:

   Calibration coefficient: _____ (clock ticks per cm)

Figure 3

5. Now modify your program to convert the average time of flight of the ultrasonic waves into cm. Make the above code a subroutine by changing the name of the main function and adding a statement that returns an integer. (Hint: the main function

```
int i;            /* a loop counter */
int num;          /* the number of readings to average */

int ticks;        /* the # of clock ticks returned for */
                  /* a single reading */
int ticksAvg;     /* the average of num readings */

void main() {
   num = 1;       /* set the number of readings to average */
   sonar_init();              /* initialize the sonar function */
                              /* must be called before sonar() */
   printf("press start\n");
   while(1) {                 /* loop continually */
      if(start_button()) {    /* take num readings every time */
                              /* start is pushed */
         ticksAvg = 0;        /* initialize to 0 */
         for(i=0;i<num;i++) { /* take num readings */
            ticks = sonar() / num;  /* dividing first then summing */
                                    /* will avoid an overflow error*/
            if(ticks > 0) {   /* occasionaly the sonar() function */
                              /* fails and returns a neg. value, */
                              /* discard it */
               ticksAvg += ticks;   /* sum ticks to get average */
            }
         }
         printf("ticks = %d\n",ticksAvg); /* print the result */
      }
   }
}
```

should look like: int YOURNAME(){code}). Write a new main function in the same C file that calls the subroutine and performs the conversion. Save a copy of your code to use in your report.

6. Use your program to determine the maximum distance the sonar is capable of detecting. Use an object with a large surface area such as a book. Determine the minimum distance the sensor is capable of measuring. Campare uur results with the data supplied by Acroname by clicking on the links on the lab hondout page.

   Minimum Distance: _____ (cm) Maximum Distance _____ (cm)

7. Using a small object, such as an empty soda can or floppy disk, determine the cone angle, as shown in Figure 1. Collect data at several distances from the sonar unit so that you can make a plot representing the cone angle for your report. Compare your data with that supplied by Acroname (see the web links described earlier).

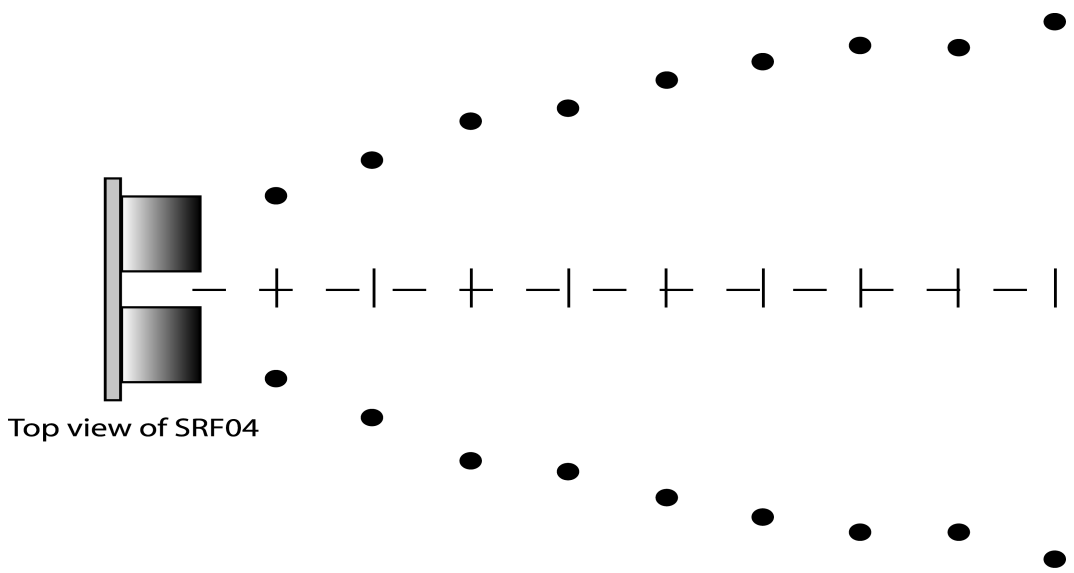

Figure 4. Measuring the cone angle of the SRF04

Cone Angle: _____ (degrees)

8. Experiment more with the sonar sensor and determine some problems it may have. Consider the following questions that are at the end of the lab: Can the sonar module determine the exact location of an object? What about size? How could you use the sensor to determine the size of an object? Can the sensor respond to quick moving objects?

9. Write a program that will make a CutieBot locate and approach an object. The CutieBot must be placed with the SRF04 facing away from the object at a distance of approximately one-meter. The CutieBot must then scan the area for the object. Boolean operators may come in handy here to accommodate a reasonable range. When the CutieBot has found the object, have it approach the object but then stop when it is approximately 10-cm from the object.

Questions:

1. Based on your calibration coefficient, how long is a clock tick? What is the processor frequency? Hint: treat the clock tick as a period, T, where the frequency of the CPU, f, is f = 1 / T. Note that the speed of sound in air is 343 m/s in air at 20° C.

2. Can the sonar sensor determine the size of an object? How could you use the sensor to determine the size of an object?

3. What is the approximate angular and radial resolution of the Devantech sonar

4. Can the sensor respond to quick moving objects? Why or why not?

5. Describe the strategy that you used to home in on the object with the CutieBot.

6. How does your CutieBot code accommodate for the possibility that the object may disappear from the sonar cone as the CutieBot approaches?