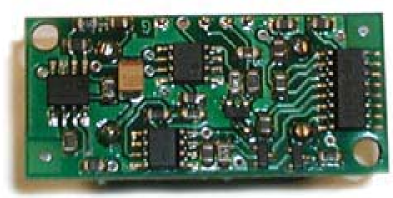
Department of Electrical & Computer Engineering

Application of the Devantech SRF04 Ultrasonic Range Finder

Author: Keunsuk Lee
Date: November 15, 2002



Executive Summary

This document entails the author's experience, knowledge, and application of use of ultrasonic range sensors. In summary, Ultrasonic sensors use reflected or transmitted ultrasonic waves to detect the presence or absence of a target component. The output is trivial in respect that the sensor merely detects whether the target is or is not within the design detection range. The application usages of the sonar sense are wide and vast. 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.

Objectives

The objective of this document is to provide the reader with the experience and know-how of using an ultrasonic sensor based off the authors experiences. Before going into implementation instructional notes, it is necessary that one understands the basic functionality of a sonar sensor. Followed by the introductory to sonar sensors, the reader will be subject to the author's experimental analysis, implementation, and concluding remarks of the Devantech SRF04 Ultrasonic Ranger Finder. In aiding the reader to possibly reproduce or optimize the author's work, the use of diagrams, source code with explanations, and schematics is also provided for the user's endeavors.

Key Words

Microcontroller, Ultrasonic Range Sensor, PIC16F874

I. Introduction: How Sonar Sensors Work

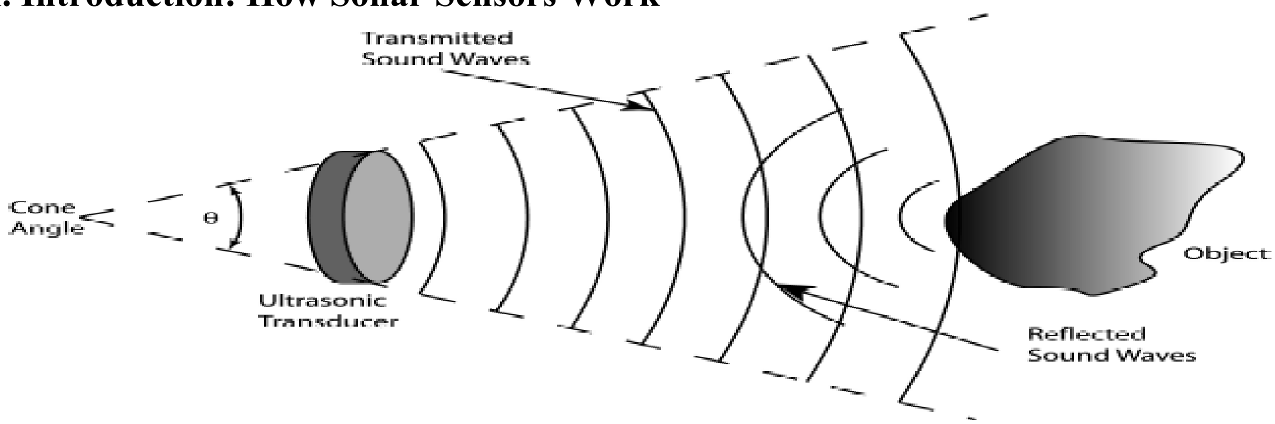


Figure 1. Functionality of a generic sonar sensor

Figure 1 above shows a basic diagram of a generic sonar sensor in operation. 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 then 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. The Devantech SRF04 Ultrasonic Range Finder is unique in that it has a separate transmitter and receiver. This allows for the smallest minimum detectable distance. The frequency of the ping is 40kHz. 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 40kHz return signal to identify it as an echo. An advantage of the SRF04 is that the frequency generating circuit is integrated into the printed circuit board supporting the transducers. This minimizes the amount of wiring needed to utilize the sensor. The SRF04 offers precise ranging information from roughly 3cm to 3 meters. This range and minimal power requirements, 5 volts, make this an ideal ranger for robotics applications.

II. Connections

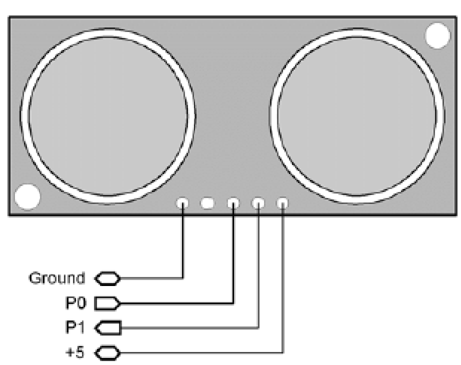


Figure 2. pin connections

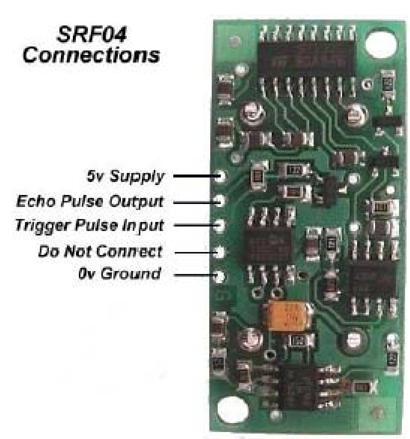


Figure 3. pin connections

The SR04 requires four connections to operate. First is the power and ground lines. The SR04 requires a 5v power supply capable of handling roughly 50mA of continuous output. For testing purposes, the author provides this power through a Hewlett Packard E3630A Triple Output DC Power Supply. The remaining two wires are the trigger pulse in and the echo pulse out. These two lines shall be connected directly to register ports of any desired microcontroller. The basic operation between these I/O lines is as followed: Echo pulse line will output measured distance in the form of a varied length square wave form upon the initial pulsing of the input trigger line. In other words, the designer must manually trigger the sonar sensor to send out an echo pulse to detect object distances. As for the main control unit, the author has chosen the PIC16F874 as the ideal microcontroller to use, utilizing the PIC's register port B digital I/O registers for input and output of the two main SRF04 control lines. Below, Figure 4 depicts the basic schematic of how one should go about connecting the SRF04 to a PIC16F874 microcontroller and oscillator.

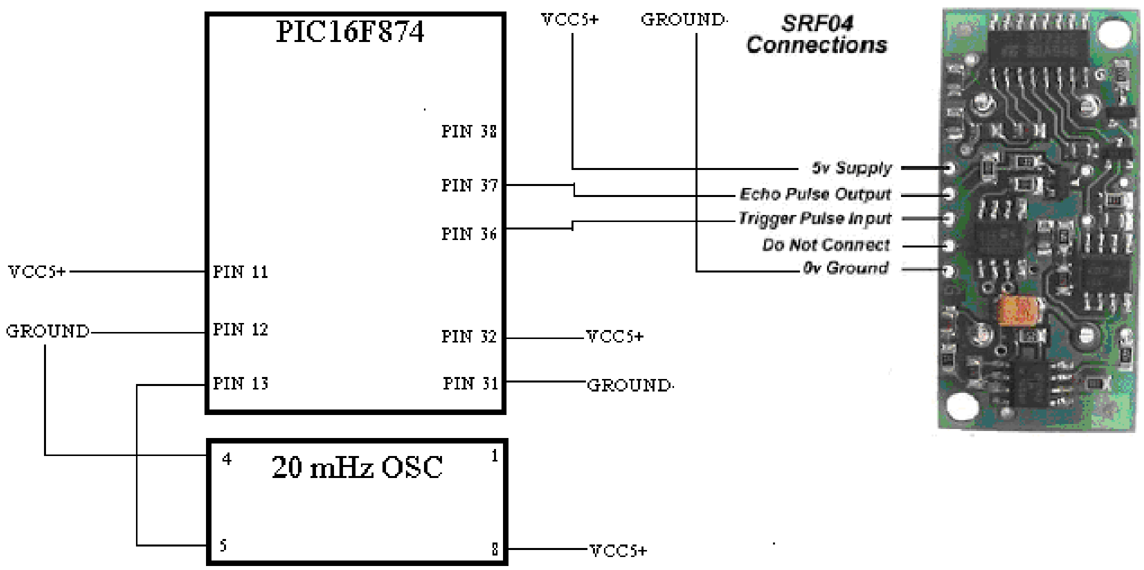


Figure 4. SRF04 interfaced to a PIC16F874 Microcontroller

The 20mHz oscillator provides as a function generator to create the required wave pulses that will operate the PIC and the SRF04. One could analogize the oscillator as the main bloodline for the entire circuit. In terms of difficulty of construction of this circuit, looking at the diagram, it is trivial. The only soldering that is required are the four main connections to the SRF04. The rest can be interfaced via a proto-board or breadboard.

III. Trigger/Echo Pulse Timing and Object Distance Measurements

There are a couple of requirements for the input trigger and output pulse generated by the SRF04. The input line should be held logic low and then brought high for a minimum of 10 usec to initiate the sonic pulse. In order to generate the trigger pulse with the PIC, an output high pulse must be generated via code plus an added subroutine for the required 10 usec delay for the trigger pulse + 10 usec additional delay for the 8 cycles between the falling edge of the trigger and the rising edge of the echo. The echo pulse is generated on the falling edge of this input trigger. The ranger's receive circuitry is held in a short blanking interval of a minimum of 100usec to avoid noise from the initial ping and then it is enabled to listen for the echo. The echo line is low until the receive circuitry is enabled. Once the receive circuitry is enabled, the falling edge of the echo line signals either an echo detection or a time out detection which signals that there is no object or an object out of sensing range (3 cm – 3 meters). For safe measure, the author had made sure the trigger pulse would not trigger again for another 50 msec. These delays were also coded into the PIC microcontroller. For measuring distance the PIC can be coded to start timing on the falling edge of the trigger input and end timing on the falling edge of the echo line. This duration determines the distance to the first object the echo is received from. If no object is detected, the echo pulse will timeout and return an echo at approximately 36 msec. For actual viewing of the trigger input and the echo pulse output, the author had connected output lines from PIN 37 and PIN 36 of the PIC and displayed them on a PHILIPS PM3365 Oscilloscope in digital mode.

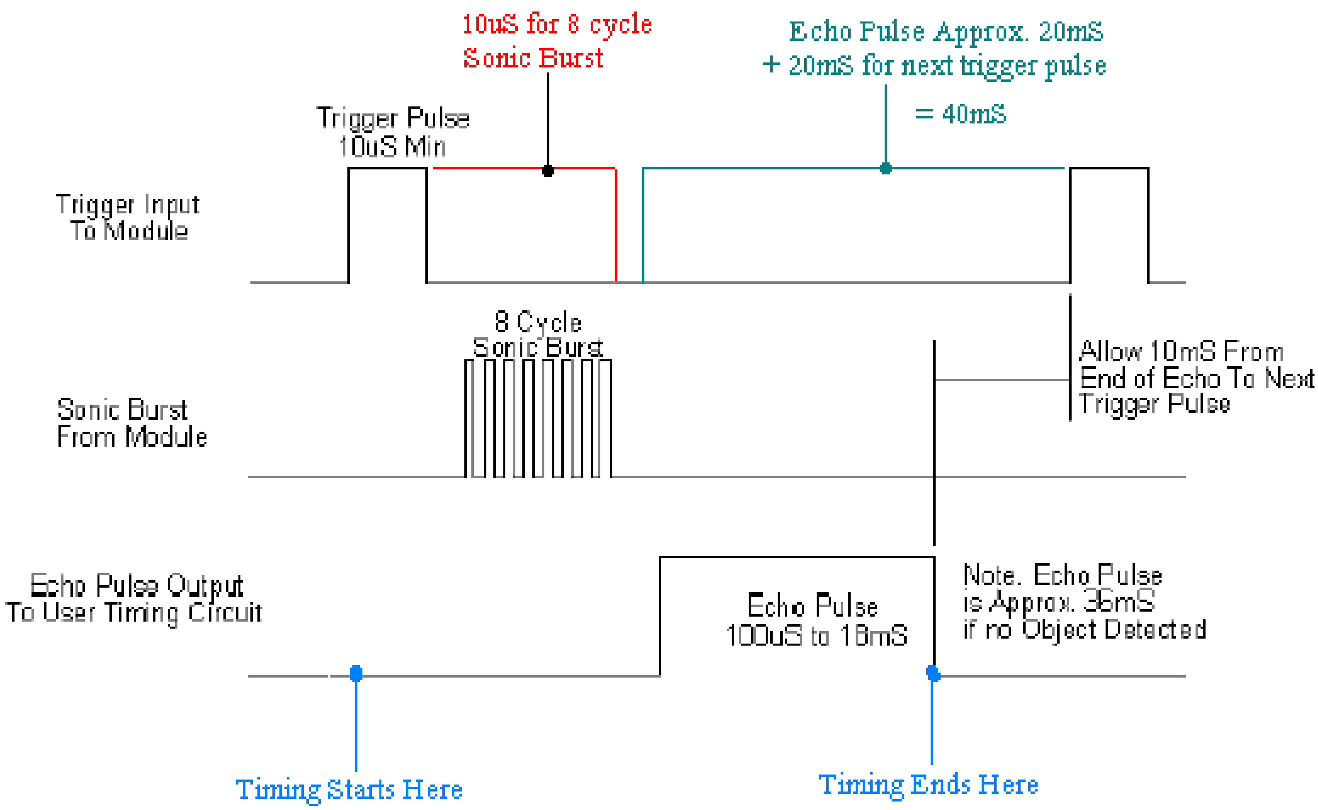


Figure 5. Timing Specifications

Based on actual waveforms of the echo output, the following table below has been devised. It is quite obviously clear that there is a working relationship and even ratio between the measured detection distances and the corresponding measured output waveforms. The average ratio calculated is 1 in. per 166 usec. In relation to coding for the PIC, this relation can be used in calculating actual distances.

Test Range #	Measured Detection Distance	Echo Output Hi Time
1	3 IN.	500 us.
2	6 IN.	1000 us.
3	9 IN.	1500 us

Table 1. Experimental Test Values/Distances/Times

IV. Example Source Code and Explanation

The following code routine has not been shown in it's entirety. What has been revealed to the reader are the basic setup procedures for getting the SRF04 into operation. The following code has been written utilizing the PIC assembly language instruction set.

```
;*****************************************************************
; Main Program
;*****************************************************************
```

(1)
We begin are main code by calling subroutines BANK0 and BANK1. These subroutines are initializations that clears port register B and initializes which bits will be used for input and out put.

```
Start
           call  BANK0
           call  BANK1
           goto  Main
```

(2)
We can now start by pulsing hi the input trigger for 20us then bringing it back low.

```
Main       nop
Loop       bsf   PORTB,3    ;trigger hi
           call  Delay_20us ;let the trigger pulse for 20us
           bcf   PORTB,3    ;trigger lo = triggers an echo
                            ;pulse out from the sensor
```

(3)
The input trigger causes the SRF04 to pulse out an echo. From here we can begin timing between the echo pulse out and echo pulse in. From this information we can calculate and approximate distance of detection.

```
           call  pulse_out  ;check for range sensor output pulse
           call  Delay_20ms ;delay for 20ms
           call  Delay_20ms  ;+ 20ms
           call  Delay_10ms  ;+ 10ms
           goto  Loop
pulse_out
           btfsc PORTB,4    ;if rising edge
           goto  pulse_in   ;then get ready for the pulse_in
           goto  pulse_out  ;else keep on checking for rising edge
pulse_in
           btfss PORTB,4    ;if falling edge
           goto  Done       ;then return to main program
           nop
           goto  pulse_in   ;and keep on checking for falling edge
```

(4)
These following are subroutines for calculating particular delay values of 20us,20ms, and 10ms. They can be reused as many times as the coder wishes to, where ever he/she may require a delay.

```
Delay_20us
          call WaitA
          retlw 00
Delay_20ms
          call WaitB
          retlw 00
Delay_10ms
          call WaitC
          retlw 00
WaitA
          btfsc d1,4
          goto  Done
          incf  d1,1
          goto  WaitA
WaitB
          btfsc d2,7
          call  reset_counter
          incf  d2,1
          btfsc multiplier,7
          goto  Done
          goto  WaitB
WaitC
          btfsc d2,7
          call  reset_counter
          incf  d2,1
          btfsc multiplier,5
          goto  Done
          goto  WaitC
reset_counter
          movlw d'0'
          movwf d2
          incf  multiplier,1
          retlw 00
Done
          movlw d'0'
          movwf d1
          movlw d'0'
          movwf d2
          movlw d'0'
          movwf multiplier
          retlw 00

          end
```

V. Conclusion

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 methods of this application note may be followed as is or modified accordingly at the experimenter's discretion.

VI. References

http://www.acroname.com

http://content.honeywell.com/sensing/prodinfo/ultrasonic/

http://www.diyelectronics.com/Accessories/URF.html

https://www.zagrosrobotics.com/index.asp?NewNewProductsPage.asp~main