Introduction. This application note presents a circuit that allows the BASIC Stamp to measure distances from 1 to 12 feet using inexpensive ultrasonic transducers and commonly available parts.

Background. When the November 1980 issue of *Byte* magazine presented Steve Ciarcia's article *Home in on the Range! An Ultrasonic Ranging System*, computer hobbyists were fascinated. The project, based on Polaroid's SX-70 sonar sensor, allowed you to make real-world distance measurements with your computer. We've always wanted to build that project, but were put off by the high cost of the Polaroid sensor ($150 in 1980, about $80 today).

If you're willing to give up some of the more advanced features of the

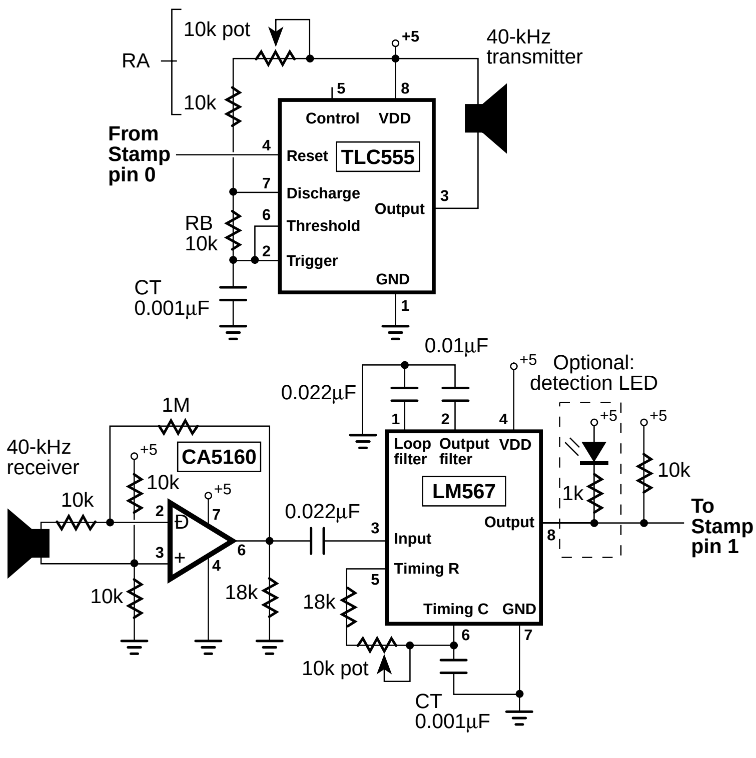

Figure 1. Schematic to accompany program SONAR.BAS.

Polaroid sensor (35-foot range, multi-frequency chirps to avoid false returns, digitally controlled gain) you can build your own experimental sonar unit for less than $10. Figure 1 shows how.

Basically, our cheap sonar consists of two sections; an ultrasonic transmitter based on a TLC555 timer wired as an oscillator, and a receiver using a CMOS op-amp and an NE567 tone decoder. The Stamp controls these two units to send and receive 40-kHz ultrasonic pulses. By measuring the elapsed time between sending a pulse and receiving its echo, the Stamp can determine the distance to the nearest reflective surface. Pairs of ultrasonic transducers like the ones used in this project are available from the sources listed at the end of this application note for $2 to $3.50.

Construction. Although the circuits are fairly self-explanatory, a few hints will make construction go more smoothly. First, the transmitter and receiver should be positioned about 1 inch apart, pointing in the same direction. For reasons we'll explain below, the can housing the transmitter should be wrapped in a thin layer of sound-deadening material. We used self-adhesive felt from the hardware store. Cloth tape or thin foam would probably work as well. Don't try to enclose the transducers or block the receiver from hearing the transmitter directly; we count on this to start the Stamp's timing period. More on this later. For best performance, the oscillation frequency of the TLC555 and the NE567 should be identical and as close to 40 kHz as possible. There are two ways to achieve this. One way is to adjust the circuits with a frequency counter. For the '555, temporarily connect pin 4 to +5 volts and measure the frequency at pin 3. For the '567, connect the counter to pin 5.

If you don't have a counter, you'll have to use ±5-percent capacitors for the units marked CT in the '555 and '567 circuits. Next, you'll need to adjust the pots so that the timing resistance is as close as possible to the following values. For the '555: Frequency = 1.44 / ((RA + 2∗RB)∗ CT), which works out to $40 \times 10^3 = 1.44 / ((16 \times 10^3 + 20 \times 10^3) \times 0.001 \times 10^{-6})$.

Measure the actual resistance of the 10k resistors labeled RA and RB in the figure and adjust the 10k pot in the RA leg so that the total of the equation RA + 2∗RB is 36k. Once the resistances are right on, the frequency of oscillation will depend entirely on CT. With 5-percent tolerance, this puts you in the ballpark; 38.1 to 42.1 kHz.

For the ’567 the math comes out like so: Frequency = 1/(1.1*R*CT); $40x10^3 = 1/(1.1 \times 22.73x10^3 \times 0.001x10^{-6})$

Adjust the total resistance of the 18k resistor and the pot to 22.73k. Again, the actual frequency of the ’567 will depend on CT. With 5-percent tolerance, we get the same range of possible frequencies as for the ’555; 38.1 to 42.1 kHz.

1

Once you get close, you can fine-tune the circuits. Connect the LED and resistor shown in the figure to the ’567. Temporarily connect pin 4 of the ’555 to +5 volts. When you apply power to the circuits, the LED should light. If it doesn’t, gradually adjust the pot on the ’555 circuit until it does. When you’re done, make sure to reconnect pin 4 of the ’555 to Stamp pin 0. Load and run the program in the listing. For a test run, point the transducers at the ceiling; a cluttered room can cause a lot of false echoes. From a typical tabletop to the ceiling, the Stamp should return `echo_time` values in the range of 600 to 900. If it returns mostly 0s, try adjusting the RA pot very, very slightly.

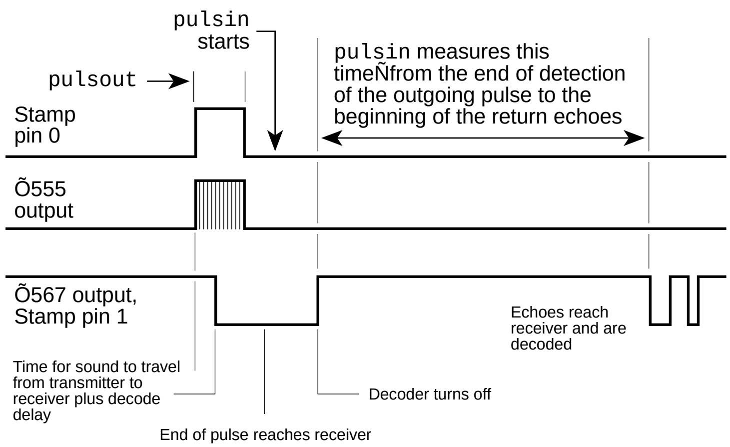

Figure 2. Timing diagram of the sonar-ranging process.

How it works. In figure 1, the TLC555 timer is connected as a oscillator; officially an *astable multivibrator*. When its reset pin is high, the circuit sends a 40-kHz signal to the ultrasonic transmitter, which is really just a specialized sort of speaker. When reset is low, the ’555 is silenced.

In the receiving section, the ultrasonic receiver—a high-frequency microphone—feeds the CA5160 op amp, which amplifies its signal 100 times. This signal goes to an NE567 tone decoder, which looks for a close match between the frequency of an incoming signal and that of its internal oscillator. When it finds one, it pulls its output pin low.

Figure 2 illustrates the sonar ranging process. The Stamp activates the ’555 to send a brief 40-kHz pulse out through the ultrasonic transmitter. Since the receiver is an inch away, it hears this initial pulse loud and clear, starting about 74 µs after the pulse begins (the time required for sound to travel 1 inch at 1130 feet per second). After the ’567 has heard enough of this pulse to recognize it as a valid 40-kHz signal, it pulls its output low.

After `pulsout` finishes, the transmitter continues to ring for a short time. The purpose of the felt or cloth wrapping on the transmitter is to damp out this ringing as soon as possible. Meanwhile, the Stamp has issued the `pulsin` command and is waiting for the ’567 output to go high to begin its timing period. Thanks to the time required for the end of the pulse to reach the receiver, and the pulse-stretching tendency of the ’567 output filter, the Stamp has plenty of time to catch the rising edge of the ’567 output.

That’s why we have to damp the ringing of the transmitter. If the transmitter were allowed to ring undamped, it would extend the interval between the end of `pulsout` and the beginning of `pulsin`, reducing the minimum range of the sonar. Also, if the ringing were allowed to gradually fade away, the output of the ’567 might chatter between low and high a few times before settling high. This would fool `pulsin` into a false, low reading.

On the other hand, if we prevented the receiver from hearing the transmitter at all, `pulsin` would not get a positive edge to trigger on. It would time out and return a reading of 0.

Once pulsin finds the positive edge that marks the end of the NE567's detection of the outgoing pulse, it waits. Pulsin records this waiting time in increments of 10 µs until the output of the '567 goes low again, marking the arrival of the first return echo. Using debug, the program displays this delay on your PC screen.

To convert this value to distance, first remember that the time pulsin measures is the round-trip distance from the sonar to the wall or other object, and that there's an offset time peculiar to your homemade sonar unit. To calibrate your sonar, carefully measure the distance in inches between the transmitter/receiver and the nearest wall or the ceiling. Multiply that number by two for the roundtrip, then by 7.375 (at 1130 feet/second sound travels 1 inch in 73.746 µs; 7.375 is the number of 10-µs pulsin units per inch). Now take a Stamp sonar reading of the distance. Subtract your sonar reading from the calculated reading. That's the offset.

1

Once you have the offset, add that value to pulsin's output before dividing by 7.375 to get the round-trip distance in inches. By the way, to do the division with the Stamp's integer math, multiply the value plus offset by 10, then divide by 74. The difference between this and dividing by 7.375 will be about an inch at the sonar's maximum range. The result will be the round-trip distance. To get the one-way distance, divide by two.

Modifications. The possibilities for modifications are endless. For those who align the project without a frequency counter, the most beneficial modification would be to borrow a counter and precisely align the oscillator and tone decoder.

Or eliminate the need for frequency alignment by designing a transmitter oscillator controlled by a crystal, or by the resonance of the ultrasonic transducer itself.

Try increasing the range with reflectors or megaphone-shaped baffles on the transmitter and/or receiver.

Soup up the receiver's amplifier section. The Polaroid sonar unit uses variable gain that increases with the time since the pulse was transmitted to compensate for faint echoes at long distances.

Make the transmitter louder. Most ultrasonic transmitters can withstand inputs of 20 or more volts peak-to-peak; ours uses only 5.

Tinker with the tone decoder, especially the loop and output filter capacitors. These are critical to reliable detection and ranging. We arrived at the values used in the circuit by calculating reasonable starting points, and then substituting like mad. There's probably still some room for improvement.

Many ultrasonic transducers can work as both a speaker and microphone. Devise a way to multiplex the transmit and receive functions to a single transducer. This would simplify the use of a reflector or baffle.

Parts sources. Suitable ultrasonic transducers are available from All Electronics, 1-800-826-5432. Part no. UST-23 includes both transmitter and receiver. Price was $2 at the time of this writing. Marlin P. Jones and Associates, 1-800-652-6733, stock #4726-UT. Price was $3.95 at the time of this writing. Hosfelt Electronics, 1-800-524-6464, carries a slightly more sensitive pair of transducers as part no. 13-334. Price was $3.50 at the time of this writing.

Program listing. This program may be downloaded from our Internet ftp site at ftp.parallaxinc.com. The ftp site may be reached directly or through our web site at http://www.parallaxinc.com.

```
' Program: SONAR.BAS
' The Stamp runs a sonar transceiver to measure distances
' up to 12 feet.

Symbol  echo_time = w2          ' Variable to hold delay time

setup:  let pins = 0            ' All pins low
        output 0                ' Controls sonar xmitter
        input 1                 ' Listens to sonar receiver

ping:   pulsout 0,50            ' Send a 0.5-ms ping
        pulsin 1,1,echo_time    ' Listen for return
        debug echo_time         ' Display time measurement
        pause 500               ' Wait 1/2 second
        goto ping               ' Do it again.
```