

FEATURE ARTICLE

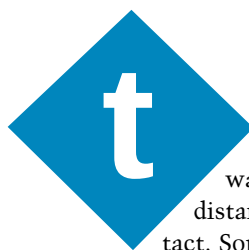
Fabio Piana

The PSoC RangeFinder

A Simple Ultrasonic Distance Meter

Some distances are more difficult to measure than others.

Therefore, Fabio built his own ultrasonic rangefinder, which won First Prize in the Cypress PSoC design contest. It can be used for measuring liquid levels, generic distances, and even the depth of snow.



here are several ways to measure distance without contact. Some products

have infrared light emitters and receivers that determine an object's distance by implementing the optical triangulation method. Other devices have laser-based systems that increase accuracy and precision. For electrically conductive metal objects, the eddy current method is an option, and capacitive sensors that are independent of the metal used in the measured objects can be used.

I decided to use ultrasonic waves. My ultrasonic PSoC RangeFinder measures the amount of time it takes for a pulse of sound to travel to a par-

ticular surface and return. Then, the device calculates the distance based on the estimated speed of sound. In this article, I'll explain how I built this simple ultrasonic distance meter.

Photo 1a is a picture of my PSoC RangeFinder with an LCD. The display is optional, and I removed it for Photo 1b. For this particular application, the only required components are a PSoC microcontroller, two 40-kHz ultrasonic transducers, two resistors, and two capacitors. Similar circuits are typically complicated and expensive, consisting of a large number of integrated circuit and passive components.

Take a look at the RangeFinder's system block diagram in Figure 1. As you can see, it's divisible into three parts: the transmitting section, receiving section, and output section. Each section contains several PSoC blocks. Using the PSoC chip family, all of the digital and analog devices are on-board with the microcontroller.

The RangeFinder has numerous applications. You can use it for the positioning of robots as well as measuring generic distances, liquid levels in tanks, and the depth of snow banks. The device can also serve as a motion detector in production lines where surfaces must not be damaged, or you can use it for educational purposes.

A restricted target angle (it requires a near-perpendicular surface) and large beam, which can create poor resolution, seem to be the RangeFinder's only limitations. Despite these drawbacks, you'll find the device's main features to be extremely useful.

The RangeFinder has a 40-kHz operating frequency, a range of 25 to 200 cm, and 1-cm resolution. In addition, it

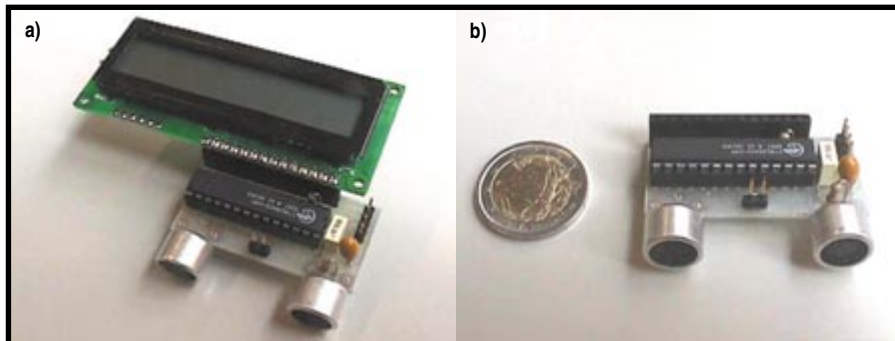


Photo 1a—There are several things to consider when building your own ultrasonic rangefinder, including whether or not to incorporate an LCD. **b**—Take a look at the PSoC RangeFinder without the LCD.

Listing 1—The time base interrupt routine is the heart of the measuring process. Remember, the PGA_1 gain is dynamically reconfigured from one to 16.

```
TimeBaseINT:
//Interrupt rate = 58 µs
push a
inc [time1]
cmp [time1],26 ;25 //35 × 58 ms = 2-ms blank
jc timebase1 //Jump if time 1 < 35
//Test comparator
tst reg[CMP_CR],0b01000000 //Test if comparator = 1
jz timebase1 //If comparator = 0, then jump
//gest comparator = 1 (pong)
mov A,[time1]
mov [range],A //Distance in range
call TimeBase_DisableInt //Disable time base interrupt
timebase1:
//Set PGA_1 gain
mov A,[time1]
cpl A
or A,0b00001111 //Add X8h
call PGA_1_SetGain //Set pga_1 gain in 16 steps
pop A
reti
```

requires only a single 5-VDC power supply and draws just 25 mA (23 mA without the LCD). The device has one PWM output and one TTL-level serial output (9600 bps, 8 bits, 1 stop bit, and no parity). Finally, don't forget the optional 2 × 16 LCD, software calibration, and dynamic receiver stage gain increment.

MICRO CONFIGURATION

Several PSoC microcontroller resources were used in this project. I applied the following PSoC digital resources: five 8-bit counters, an 8-bit serial transmitter, two 8-bit PWMs, and a digital inverter. I implemented the following analog resources: one programmable gain amplifier (PGA), two two-pole passband filters, a programmable threshold comparator, and a reference multiplexer. In addition, I used the EEPROM and LCD toolbox software modules.

Photo 2 is a screen shot of the PSoC Designer with the Device Editor showing the placement of the PSoC's digital and analog resources. As you can see, a large number of resources are required. All of the digital blocks and several analog blocks are employed.

The software modules include the LCD toolbox user module and a complete set of library routines that allows you to write numbers and strings to a two-line LCD using standard Hitachi HD44780 commands. An EEPROM emulation library allows you to store important data in flash memory space as if it were a physical EEPROM.

PRINCIPLES

The transmission section is based on four digital blocks allocated in the DBA00, DBA01, DBA02, and DBA03 blocks. An 8-bit counter (called "TimeBase") provides a 17,240-Hz time base frequency. Sound velocity depends on ambient air temperature, which is calculated for an air temperature value of 22°C:

$$V_{\text{SOUND IN AIR}} = 331.4 + (0.6 \times T) \text{ m/s}$$

where T is equal to the air temperature (°C). So, for a middle value of 22°C:

$$V_{\text{SOUND IN AIR}} = 331.4 + (0.6 \times 22) = 344.6 \text{ m/s} = 34,460 \text{ cm/s}$$

For a round-trip ping, you have:

$$f_{\text{TimeBase}} = \frac{V_{\text{SOUND IN AIR}}}{2} = 17.23 \text{ kHz}$$

And with a 24-MHz MCU clock:

$$f_{\text{TimeBase}} = \frac{24 \text{ MHz}}{116} = \frac{2 \text{ MHz}}{116} = 17.24 \text{ kHz}$$

An 8-bit counter (F40kHz) drives the ultrasonic transmitter and a digital inverter (F40kHz_inv). The phase of the voltage applied to the positive and negative terminals of the sensor has been shifted 180°, so two times the supply voltage is applied to the sensor.

The 40-kHz transmission enables an 8-bit counter (called "Meter") that increments one step per centimeter. The Meter clock input is TimeBase (17,240 Hz).

The ultrasonic receiver's negative terminal is connected to the analog ground reference (AGND—pin 25, P02) provided by RefMux_1, a reference multiplexer allocated in the ACA03 block. The ultrasonic receiver's positive terminal is connected to an amplification chain based on a programmable gain amplifier (PGA_1) and two two-pole passband filters (BPF2_1 and BPF2_2). The first passband filter is designed for a 40-kHz center frequency and a correspondent gain of 33 dB. The second filter is also designed for a 40-kHz center frequency,

but it has a 10-dB passband gain. Because of the discrete value of the capacitors integrated in the switched-capacitor analog blocks, the real frequency response is different from the nominal one.

Figure 2 shows the BPF2_1 and BPF2_2 frequency responses. The BPF2 output is sent to the programmable threshold comparator (CMPR_1). When a 40-kHz

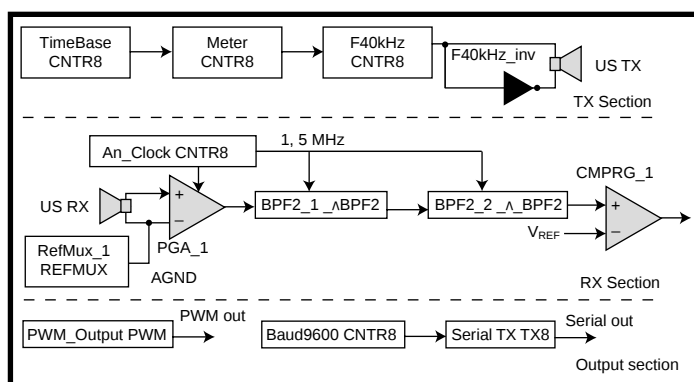


Figure 1—There are three distinct sections in my PSoC RangeFinder project: digital blocks, analog blocks, and transducers.

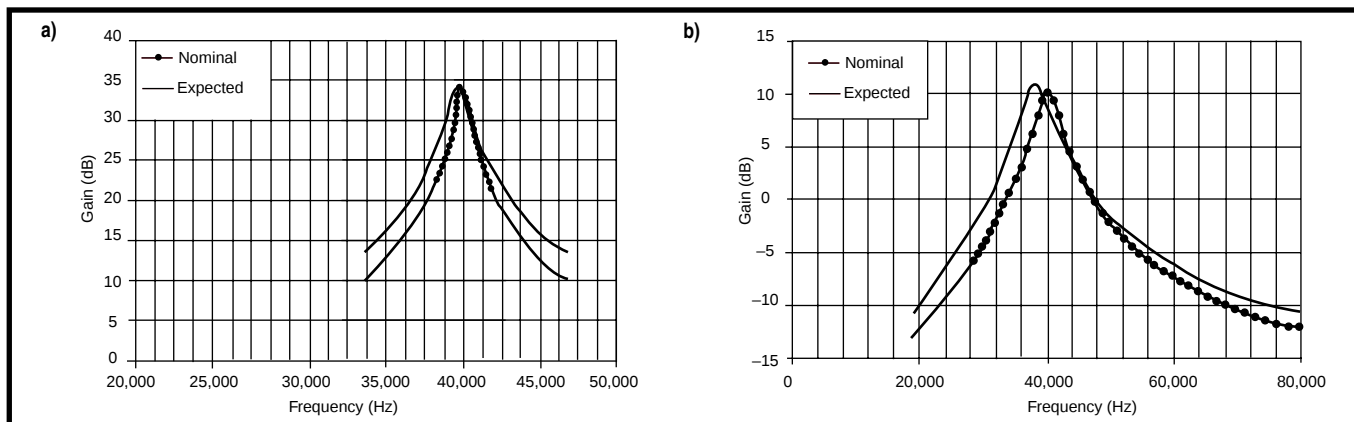


Figure 2a—This is the BPF2_1 frequency response graph. **b**—The BSF2_2 frequency response graph is a little different. For the filter design, I used the appropriate Cypress filter design folder in Excel.

signal is received, the CMPPRG_1 output is high logic level, so the software can read it.

There are two different output devices in the output section: an 8-bit pulse width modulator and an 8-bit serial transmitter. In the former, the frequency of the rectangular signal that's available on the PWM output (P2.7, pin 5) is approximately 780 Hz; its pulse width is proportional to a measured distance value (5 μ s per centimeter). The latter is a standard TTL logic-level serial output. Transmission parameters are 9600 bps, 8 data bits, 1 stop bit, and no parity bit. An 8-bit counter, Baud9600, provides a 9600-Hz time base for the serial transmitter block, SerialTX, with its output on P2.4 (pin 22). If you need RS-232 interfacing, you can use an external TTL-level RS-232 converter (e.g., the common, inexpensive MAX232). This means that you can also interface the RangeFinder to a PC or microcontroller-based system with a standard RS-232 port.

THE HARDWARE

The RangeFinder's circuitry is quite simple (see Figure 3). The most important part is U1, which is a CY8C26443 PSoc microcontroller. U1 does all of the work with its internal analog and digital blocks.

Two capacitors, C1 and C2, suppress high- and low-frequency noise on the 5-V supply line. R1 is a 100-k Ω resistor that holds the DC input voltage of the receiving stage to AGND (2, 5 V). R2 regulates the LCD contrast. I

selected this value for a good contrast in the LCDs used in prototypes. It can be changed to adjust the contrast using different LCD modules. In addition, a 10-k Ω trimmer with the wiper to the contrast input and other pins connected to 5 V and GND can replace it. This allows for decent LCD contrast regulation.

Without an LCD, the circuit draws approximately 23 mA from a 5-V power supply. If you use an LCD, the current consumption is 25 mA. The optional LCD is a standard 2 $\times$ 16 model.

THE SOFTWARE

Figure 4 is a flowchart depicting the microcontroller's software. The main program sets up the analog and digital blocks before testing JP1 to determine

Normal or Calibration mode. If JP1 is shorted, the control program runs the Normal mode procedure. Otherwise, the calibration routine is executed.

In Normal mode, the software continuously runs the transmitted ultrasonic burst (ping). After a blanking time, it waits for the returned ultrasonic signal (pong). The time between the start of a transmitted burst and the start of a received burst is proportional to the distance between the RangeFinder and the obstacle. By polling the comparator bus register, the software measures this time and stores it in a RAM location. Finally, the range value is written to the LCD (if present), sent to the serial interface, and the PWM duty cycle is set to a value that's proportional to the distance.

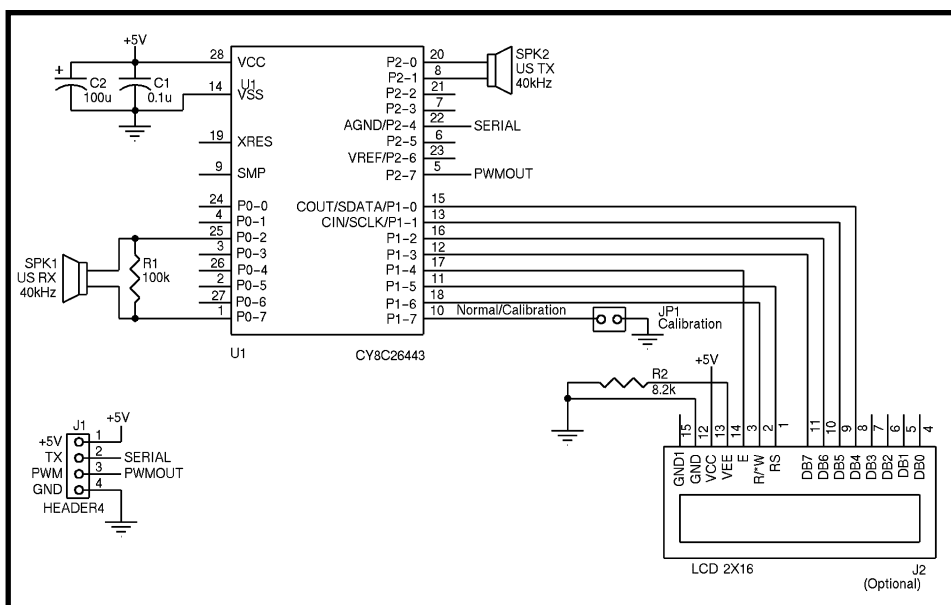


Figure 3—Building the circuitry for the PSoc RangeFinder isn't complicated. All of the functions are concentrated in the C48C26443 PSoc microcontroller, which is clearly the most important part of the design.

Gain	Value
PGA_A_G16_0	08h
PGA_A_G8_00	18h
PGA_A_G5_33	28h
PGA_A_G4_00	38h
PGA_A_G3_20	48h
PGA_A_G2_67	58h
PGA_A_G2_27	68h
PGA_A_G2_00	78h
PGA_A_G1_78	88h
PGA_A_G1_60	98h
PGA_A_G1_46	A8h
PGA_A_G1_33	B8h
PGA_A_G1_23	C8h
PGA_A_G1_14	D8h
PGA_A_G1_06	E8h
PGA_A_G1_00	F8h

Table 1—The PSoC's PGA gain values are used to set the PGA_1 gain during the measuring routine. For more information, study the PGA_A module datasheet from Cypress.

The software in Calibration mode is similar to that in Normal mode; however, the measured value is compared with the constant value 50, and the resultant offset is stored in nonvolatile EEPROM and used to calculate the measured range in Normal mode.

TimeBase_int is the interrupt subroutine for the TimeBase 8-bit counter. This is the most important portion of code (see Listing 1). When time1 is greater than the value of blank time (blank time prevents false echoes caused by the lateral

receiving of transmitted 40-kHz bursts), the software tests the logical value of the comparator. If a pong is received, the comparator output logic level is one, the time1 value is stored in RAM location range, and the TimeBase interrupt is disabled. As a result, the value stored in the range location represents the measured distance.

If the comparator logic level output is equal to zero, then the PGA_1 gain is dynamically incremented in 16 steps from one to 16 by modifying the corresponding gain register; therefore, the far echoes are much more amplified.

As you can see in Table 1, the PGA_1 gain increment is not linear. This is not a problem, because more amplification is required for long distances. As you can see in Listing 1, the PGA_1_SetGain routine is called to change the amplifier gain and the value stored in the relative configuration register.

PRACTICAL CONSTRUCTION

Constructing the circuit isn't difficult. It will take you just a few minutes. You may download the single-sided PCB design from the *Circuit Cellar* ftp site. The corresponding component layout is available on the ftp site, as well.

First, you have to mount the two wire links. The links are followed by the microcontroller socket, two resistors (mounted vertically), and two capacitors. Pay attention to the polarity of C2, which is a tantalum electrolytic capacitor. Follow with the two headers and jumper. J1 is a four-pin male header for the power supply and output connections. J2 is a 14-pin female header for the LCD connection.

The two ultrasonic transducers are soldered directly to the copper-sided padstacks. These common 40-kHz ultrasonic transducers are the kind used in car alarms. Make sure you

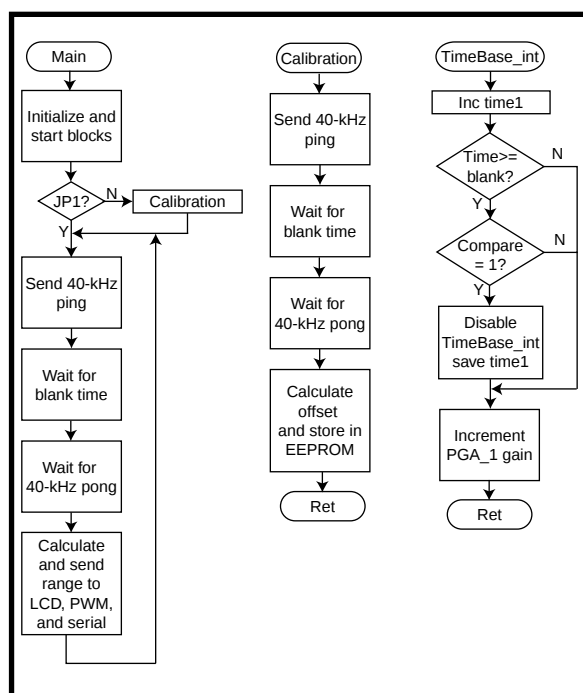


Figure 4—A flowchart will help you understand the software associated with this project. Note that there are three relevant routines: the main program, the calibration routine, and the TimeBase 8-bit counter interrupt subroutine.

position the two ultrasonic transducers correctly. Usually, the negative terminal is soldered to the case for good noise immunity. You can make the connection to the LCD by soldering a 14-pin male connection to the LCD module with a flat ribbon cable or a set of individual flexible wires. After a final inspection of the populated board, you can insert a programmed PSoC microcontroller into its 28-pin socket.

POWER AND CALIBRATION

This circuit needs a 5-V regulated and filtered power supply. Because of the reduced circuit size, the voltage regulator is not included. If an external 5-V power source isn't available and all you have is an unregulated 8-V supply, you can use a three-legged voltage regulator (i.e., 78L05 at 5 V–100 mA maximum) and some capacitors to increase ripple rejection and transient behavior.

You must calibrate the unit before it's installed. The calibration procedure is simple. First, place the

rangefinder 50 cm in front of a perpendicular, flat obstacle (e.g., a wall or wood panel). Second, remove jumper JP1 and power up the circuit by connecting the 5-V regulated power supply. Finally, insert jumper JP1 and power down the circuit. At this point the circuit is calibrated and the offset value is stored in EEPROM. The value will be read from the software at each circuit power-up and used in the distance calculating operation.

Figure 5 shows the text displayed on the LCD in Normal and Calibration modes. This text is easy to change, because it's stored as ASCII strings.

Normal mode															
P	S	o	C		r	a	n	g	e	f	i	n	d	e	r
D	i	s	t	a	n	c	e			N	N	N		c	m

Calibration mode															
5	0	c	m		c	a	l	i	b	r	a	t	i	o	n
P	I	a	c	e		J	P	1		j	u	m	p	e	r

Figure 5—The LCD text that appears in Normal and Calibration mode is fairly easy to change. Note that NNN represents the measured value.

You may download the strings from the main.asm file on the *Circuit Cellar* ftp site.

THINGS TO CONSIDER

I used the 28-pin CY8C26443 microcontroller because of its availability; however, you can use the eight- (if you don't use the LCD) or 20-pin versions for a smaller PCB. With the LCD connected, you can display useful

information while setting up and calibrating your own PSoC RangeFinder.

It's easy to change the program if you're thinking about modifying the device's behavior. Because any one of several obstacles can surface at any time, the ultrasonic distance meter can make mistakes during the measurement process. However, if you want to prevent measurement errors, you can modify the software to calculate the average value of several measurements and discard the measured values that are out of range.

The measurement process is based on a typical air temperature of 22°C.

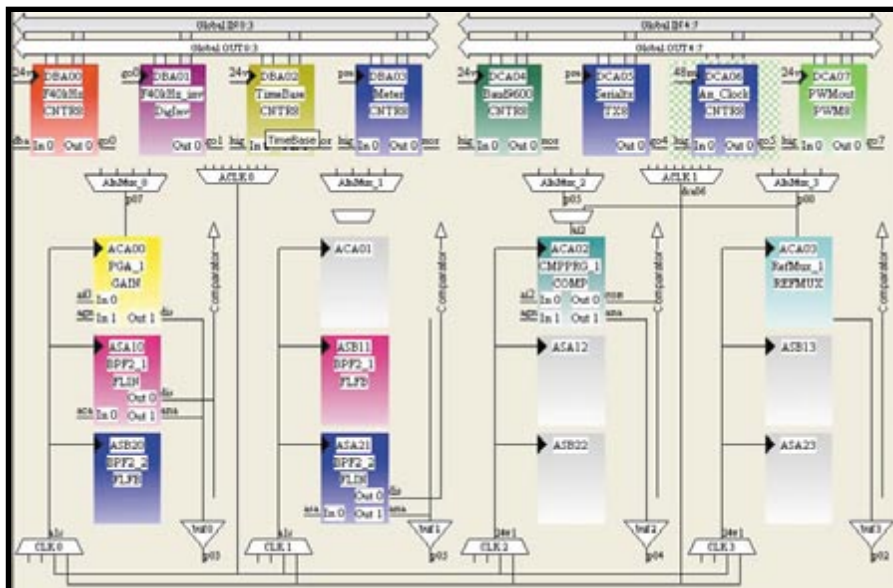


Photo 2—Note the placement of the PSoC's resources. For this application, all of the digital blocks are required.

This can be limiting when the specific application involves a wide operating temperature range.

If you use a 28- or 20-pin integrated circuit, there will be a lot of free pins in the PSoC microcontroller. It would be an improvement to implement an external integrated tempera-

ture sensor to measure the air temperature. Because of its low accuracy ($\pm 20^{\circ}\text{C}$), the internal FlashTemp module isn't suitable for the job. But, if you want to increase the precision, you can calculate and compensate for the different sound velocities at different air temperatures. 

Fabio Piana has been teaching electronics and systems automation for the last 15 years in several technical schools in Italy. In addition, he's an electronics design consultant. You may reach him at fabiopiana@tiscali.it.

PROJECT FILES

To download the code, go to ftp.circuitcellar.com/pub/Circuit_Cellar/2003/150/.

RESOURCES

Cypress MicroSystems, "CY8C25122, CY8C26233, CY8C26443, CY8C26643 Device Data Sheet," CMS10002A-R3.15, 2002.

——, "PSoC Designer, PGA_A Module Data Sheet," 2002.

SOURCE

CY8C26443 PSoC Microcontroller
Cypress MicroSystems, Inc.
(877) 751-6100
www.cypressmicro.com