EDITED BY MARTIN ROWE
AND FRAN GRANVILLE

designideas

READERS SOLVE DESIGN PROBLEMS

Oscillator has voltage-controlled duty cycle

Luca Bruno, ITIS Hensemberger Monza, Lissone, Italy

The classic Colpitts oscillator circuit in **Figure 1** generates a clock signal with a fixed duty cycle. By replacing CMOS inverter gate IC_{1A} with a voltage comparator (**Figure 2**), you can obtain a more versatile and more useful clock generator. You can set not only the oscillation frequency but also the duty cycle. You must use the comparator in an inverting configuration, which introduces a 180° phase shift. That shift, along with an additional phase shift of 180° from the capacitor-input network, lets the circuit oscillate. The circuit compares the sine wave at the output of the capacitor-filter network with the reference voltage, which drives the output of the threshold comparator high and low.

The network sets the oscillation frequency as follows:

$$f_O=\frac{1}{2\pi\sqrt{LC/2}},$$

where f_O is the oscillation frequency. With a suitable choice of the values of inductors and capacitors, the circuit can oscillate at frequencies as high as 10 MHz.

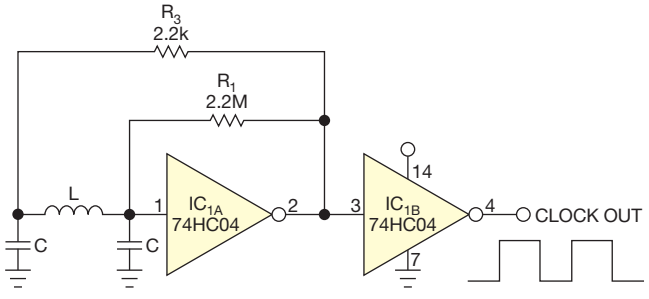

Figure 1 A typical Colpitts oscillator uses an inverter and has a fixed duty cycle.

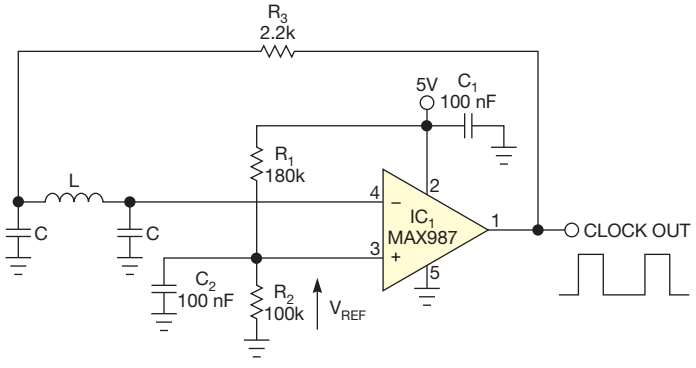

Figure 2 A comparator lets you vary the duty cycle based on resistors R_1 and R_2.

DIs Inside

▶To see all of *EDN*'s Design Ideas, visit www.edn.com/designideas.

The output clock's duty cycle depends on the reference voltage, which you can easily set through the voltage divider comprising R_1 and R_2. Unfortunately, the mathematical relationship between the reference voltage and the duty cycle is nonlinear because the sine wave at the output of the capacitor-input-filter network is not a linear function. Also, its amplitude is not constant but depends on the duty cycle of the output clock. You can easily obtain this mathematical relationship by testing the circuit with an inductance of 10 µH and a capacitance of 10 nF.

You can use any high-speed compara-

TABLE 1 OSCILLATOR DUTY CYCLE BASED ON REFERENCE VOLTAGE

Reference voltage (V)	Duty cycle (%)
0.5	15.2
1	28.3
1.5	37
2	43.5
2.5	50
3	56
3.5	62.6
4	71.5
4.5	85.4

tor with rail-to-rail inputs and outputs, such as the MAX987 from Maxim (www.maxim-ic.com), to achieve a wider input range for the reference voltage. That wider range gives you wider control of the duty cycle, although you can't reach the minimum duty cycle of 0% or the maximum duty cycle of 100%.

The propagation delay, T_{PD}, of the comparator introduces a further phase shift of value $\Delta\phi=2\pi f_O T_{PD}$, where $\Delta\phi$ is the phase shift. The capacitor-input network compensates for the phase shift, slightly reducing the oscillation frequency. For safe operation of the circuit, you should vary the reference voltage by 0.5 to 4.5V. The duty cycle varies from approximately 15 to 85% (**Table 1**). You can produce a bipolar output signal if you use a dual power supply. EDN

Generate noisy sine waves with a sound card

José M Miguel, RF-Electronics Ltd, Barcelona, Spain

Testing audio-noise-reduction circuits, PLLs (phase-locked loops), and audio-frequency filters may require a noisy sine wave, one that is summed with white noise. Using a typical computer sound card, free software, and an external amplifier circuit, you can create a noisy sine wave.

Free Generatosaur software from Wavosaur (www.wavosaur.com) turns your sound card into a low-frequency wave generator. It lets you independently choose amplitude, frequency, and waveform for the left and the right channels. The Generatosaur's user interface is a dialogue-box-style control panel (see **figure** at www.edn.com/110120dia). If you select a sine wave for the left channel and a white noise for the right channel, you then need only to use an amplifier to add the signals. **Figure 1** shows the complete circuit.

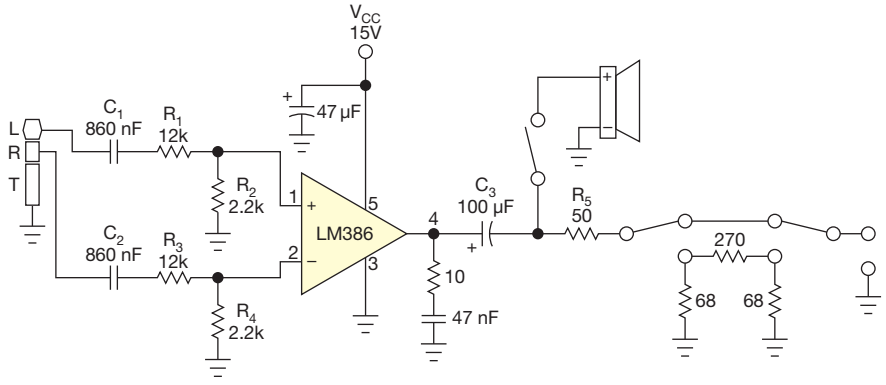

Figure 1 Generatosaur is free software that turns your sound card into a low-frequency wave generator.

The differential amplifier employs a National Semiconductor (www.national.com) LM386 audio power amplifier with a supply voltage of 15V. The output of the LM386 has a self-centered quiescent voltage that is half the power-supply voltage and that requires a blocking capacitor, C_3. Resistor R_5 sets the output impedance to 50Ω. You need the voltage dividers R_1/R_2 and R_3/R_4 because the output-voltage range for a standard sound card is 0 to 2V. Taking into account that the voltage gain of the LM386 amplifier is internally set to 20 and that its output voltage range is 7V, you need an attenuation factor, K, of 7/(2×20) in each amplifier input. The circuit also includes a selectable 20-dB attenuator that you can invoke with the two DPDT (double-pole/double-throw) switches.

Another **figure**, also available at www.edn.com/110120dia, shows a 450-Hz sine wave with a 10-dB SNR (signal-to-noise ratio). The root-mean-square noise voltage of this signal is 0.5V measured on an oscilloscope and following the tangential method. If you need to hear the generated noisy signal, connect a loudspeaker to the output of IC LM386. EDN

Decode a quadrature encoder in software

Sid Levingston, Gentec-EO, Lake Oswego, OR

Quadrature encoders work in many applications to determine displacement and direction of mechanical travel. They vary in design, but they all do the same thing: supply a set of square waves 90° out of phase. **Figure 1** shows the typical output signals.

The encoder rotates clockwise when Channel A leads Channel B. If Channel B leads Channel A, the encoder is rotating counter clockwise. By counting the pulses and the direction of rotation, you can find the position of the encoder. Although ICs can decode quadrature encoders, you can easily and less expensively have the processor decode the signal. The signals from Channel A and Channel B go through a Schmitt trigger if necessary, but many encoders and processors include this trigger internally. The signals are then applied to two I/O pins on the processor that support edge-triggered inter-

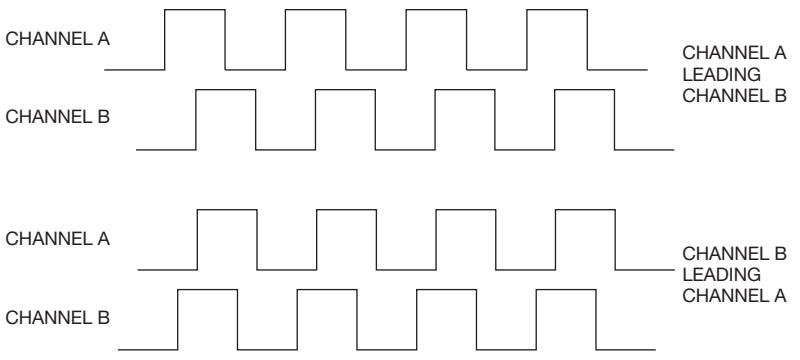

Figure 1 A quadrature encoder produces pulses that are 90° out of phase.

rupts. The code in the interrupt handler implements a standard decoder algorithm, but all algorithms typically follow these steps:

1. Set up a state table like the one in **Table 1**. The table wraps around from State 3 to State 0.
2. Initialize a counter.
3. Measure the current state of Channel A and Channel B. Find that state in the table and set a pointer to it.
4. Enable the interrupts.

In the interrupt handler, use the following steps:

1. Read the state of Channel A and Channel B.
2. If the state is the one preceding the pointer, decrement the counter.
3. If the state is the one following the pointer, increment the counter.
4. Set the pointer to the new state.
5. Clear the interrupt.

This method requires that a state table exists, that the previous state remain, and that the handler determine on each interrupt which of four states exists and then make a decision based on two possible conditions. The handler accomplishes this task with a four-case switch, in which each case has two if conditions.

Now, consider what happens in the real world. If the I/O pin generates an interrupt on a rising edge, then when the interrupt happens, that channel goes from low to high. Therefore, there's no reason to read the state of the pin that interrupted. The other channel did not interrupt because the signals are 90° out of phase. So, to determine the current state, you need only to read the state of the pin that didn't initiate the interrupt. The state of the unchanged low or high signal tells which way the encoder rotated. If it is low, then the interrupting pin is leading. If it is high, then the interrupting pin is trailing. You can use these facts to implement an efficient interrupt handler with no state table and no memory of a previous state.

The code in **Listing 1** was tested on an MSP430F processor connected to an Encoder Products (www.encoder.com) model 15T. The encoder monitored the position of a linear stage. The stage traveled 85 mm and could be tracked with resolution of 5 microns.

The define statements (highlighted in red) make the code more readable. The initPort() function (highlighted in blue) sets up the rising-edge interrupt on channels A and B. The final piece is the interrupt handler (highlighted in green). Note that it contains only six lines of code compared with the 20 or 30 lines it would take to implement the traditional method of decoding the channels. EDN

TABLE 1 LOGIC STATES OF QUADRATURE ENCODER

	Channel A	Channel B
State 0	0	0
State 1	1	0
State 2	1	1
State 3	0	1

LISTING 1 QUADRATURE-ENCODER CODE

```
#define CHA BIT0
#define CHB BIT1

initPort()
{
  P1SEL = 0;
  P1DIR = 0;
  P1OUT = 0;
  P1IES = 0;    // rising edge.
  P1IE = CHA + CHB;    // interrupt on CH A or CH B rising edge
}

#pragma vector = PORT1_VECTOR
__interrupt void port1_ISR (void)
{

  if(P1IFG & CHA )  // who interrupted? If A then A is high
   {
     (P1IN & CHB) ? gStepCount--  : gStepCount++ ;   // test B
      P1IFG &= ~CHA;    // clear interrupt
   }

  if(P1IFG & CHB ) // who interrupted? If B then B is high

   {
     (P1IN & CHA) ? gStepCount++  : gStepCount-- ;  // test A
      P1IFG &= ~CHB;     // clear  interrupt
   }
}
```

Power an LED driver using off-the-shelf components

TA Babu, Chennai, India

High-power LEDs challenge electronics engineers to design accurate and efficient, yet simple, driver circuits. Conventionally, driving high-power strings with accurate current requires dedicated switching regulators. Choosing a discrete driver circuit requires an understanding of LED lighting to make the best trade-off. This Design Idea describes a simpler and equally good way to employ the ubiquitous 555 IC.

In the converter circuit in **Figure 1**, IC_1's pins 2 and 6 connect together, which lets the device retrigger itself on each cycle. Thus, it operates as a free-running oscillator. During each cycle, capacitor C_2 charges up through timing resistor R_1 and discharges through resistor R_2. The capacitor charges up to two-thirds of the power-supply voltage, the upper comparator limit, which $0.693(R_1C_2)$ determines, and discharges itself down to one-third the power-supply voltage, the lower comparator limit, which $0.693(R_2C_2)$ determines. The total time period, T, is $0.693(R_1+R_2)C_2$.

ONCE THE VOLTAGE DROP REACHES THE BASE-EMITTER THRESHOLD OF TRANSISTOR Q_2, IT STARTS CONDUCTING.

During the on time, transistor Q_1 conducts and stores the energy in inductor L_1. When it stops conduction, the stored energy transfers to capacitor C_3 through Schottky diode D_1.

You can use the following **equations** to calculate the inductor value. The selection of an inductor depends on input voltage, output voltage, maximum current, switching frequency, and availability of standard inductor values. Once you know the inductance, you can choose the diode and the capacitor.

MOSFET Q_1 determines the duty cycle, according to the following **equation**:

$$D=1-\frac{V_{INMIN}\times\eta}{V_{OUT}},$$

where V_{INMIN} is the minimum input voltage, V_{OUT} is the desired output voltage, and η is the efficiency of the converter, estimated at 80%.

The average inductor current is

$$I_{LAVG}=\frac{I_O}{1-D},$$

where I_{LAVG} is the average inductor current and I_O is the output current.

The peak inductor current is

$$I_{LPEAK}=I_{LAVG}+\frac{\Delta I_L}{2},$$

where I_{LPEAK} is the peak inductor current and ΔI_L is the change in inductor current.

Assume that the change in inductor current is 25% over the average current. You can compute inductor L_1 as

$$L=(V_{IN}\times D)/(F_{OSC}\times\Delta I_L),$$

where F_{OSC} is the oscillator frequency. The inductor's saturation-current rating should be greater than the peak current.

To ensure constant illumination, you must monitor the current through the LED. Resistor R_3 senses the output current. Once the voltage drop across this resistor reaches the base-emitter threshold of transistor Q_2, it starts conducting, and this conduction reduces the on time of the 555 timer.

The following **equation** thus sets the LED current:

$$I_{LED}=\frac{0.6V}{R_{SENSE}},$$

where I_{LED} is the LED's current and R_{SENSE} is the sense resistance.

The minimum and maximum input and output voltages for this circuit are 10.5 and 15V, respectively. The LED string's voltage and current are 21V and 350 mA, respectively. The 6W LED driver can find numerous applications, including battery-operated portable lighting, solar-operated garden lighting, automotive lighting, bike headlights, and underwater lights. Driving high-power LED strings with standard off-the-shelf components simplifies your design without sacrificing performance. EDN

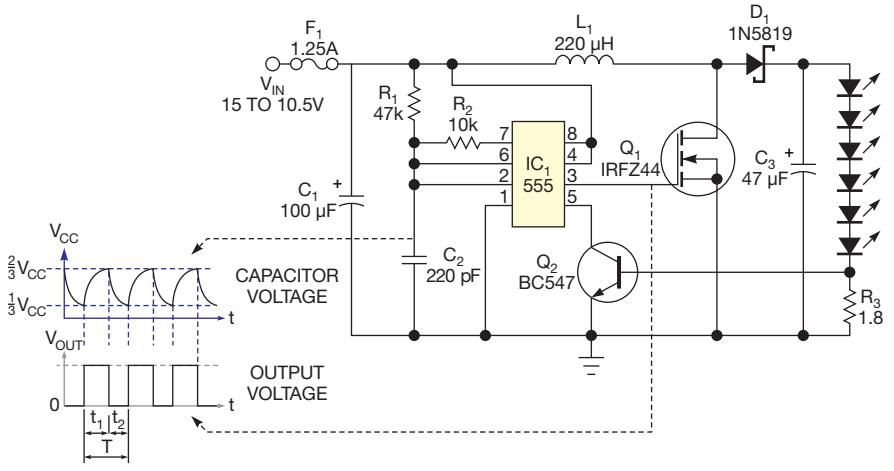

Figure 1 A free-running 555 timer provides the stimulus for driving a series of LEDs.