DESIGNER'S NOTEBOOK

Motor speed control

ROBERT GROSSBLATT

SMALL DC MOTORS, LIKE THOSE USUally found in toys, can be really handy things to keep around. They can be used in a variety of applications where the circuit you're designing has to move something other than just electrons. Those motors are great for control applications or just about anything else you can think of that doesn't require a great deal of precision.

However, those small DC motors have their own set of problems. For one, the motor's speed is notoriously dependent on the applied voltage. But that drawback can be turned into a benefit with the simple addition of a rheostat or potentiometer to make the motor speed variable. However, anyone who's ever tried making a motor-speed control using that principle, soon discovers that it's terribly unreliable, and it's an inefficient way to go about things.

No matter how small the motor, you still have to deal with the fact that they all have a certain amount of inertia, however small. That means that regulating really slow speeds with just a potentiometer is almost impossible. It usually requires that you get the motor going first and then back the potentiometer off until you achieve the desired speed.

Not only that, but if the motor draws a substantial amount of current, you're going to find that standard potentiometers won't be able to handle the power requirements: They'll start smoking and that will be the end of that. More expensive potentiometers can be used, but you'll still have the same low-speed problems. Obviously, there has to be a better way—and there is!

Controlling DC motors

Another way of controlling the speed of a DC motor is shown in Fig. 1. Here instead of controlling the motor by varying the voltage, we simply apply a constant voltage and vary the duty cycle. All that means is that we'll control the amount of time the motor is on and allow the applied voltage to remain constant. By doing things that way, the inertia of the motor can be made to work on our behalf because it will keep the motor turning until the next pulse is applied to "kick" it along. Therefore, how fast the motor turns depends on the controlling oscillator.

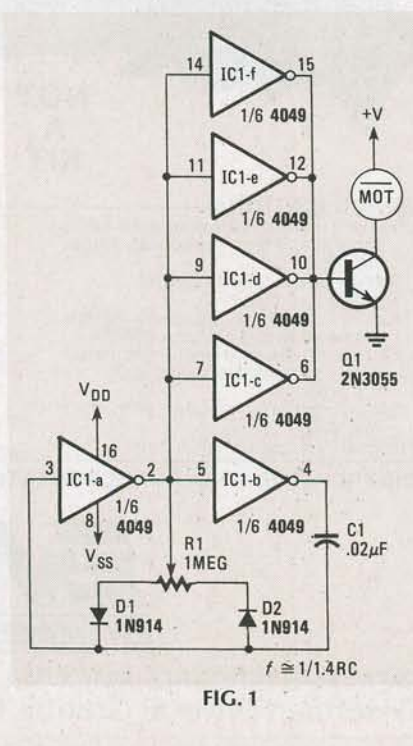

FIG. 1

Now, there are many ways to build an oscillator that can be used to control a motor. We've already seen that oscillator design is a wide open field and just about any combination of circuit building blocks can be used. Transistors, logic gates, 555 timers, and so on can be used to form the basis of a perfectly workable circuit. Each has it's own advantages and disadvantages.

Figure 1 shows an oscillator circuit that may be used in motor-speed control applications: It is by no means the absolute last word in—or the best approach to—solving the problem. It is, however, one way to go about it and is perfectly workable in a wide variety of applications. In any event, that circuit will show you the basic method to follow in designing a circuit that is capable of handling your particular requirements.

Higher precision means using a more precise oscillator and adding a crystal to the circuit to lock-in the frequency. Heavier motors will need a "beefier" output stage than the single transistor shown in Fig. 1. However, that circuit shows the basic approach to follow (you may not find it necessary to go any further).

As shown, two inverters—IC1-a and IC1-b (each ⅙ of a 4049 hex inverter)—are used to make a simple oscillator whose frequency is approximately given by:

$$f = 1/1.4RC$$

Where R is the value of the potentiometer.

The basic circuit shown is one that you've seen a million times and have probably used just as often as a convenient clocking circuit. By adding the two diodes (D1 and D2), we can split the charging of the capacitor and thereby control the positive and negative parts of the cycle—D1 controls the positive and D2 controls the negative.

The time the circuit puts out a high and turns on the motor is controlled by the value of the left part of potentiometer R1 and the low-time is controlled by the right part of potentiometer. Regardless of how the potentiometer is set, the motor will always "see" the maximum voltage and as a result, the motor is less likely to stall at low speeds. You'll also find that the motor will start turning at a much slower speed than it would if the control was achieved by varying the voltage applied to it.

Although the transistor in our example is a 2N3055, any transistor that can handle the power requirements of the motor will do. If the motor is really "chunky," you might have to make a Darlington or add another stage to the output. Ganging the four remaining gates in the IC provides enough power to drive even a 2N3055, however, other applications may require other components. Although you can use any CMOS gate that can be made to oscillate, the 4049 is heftier than most of the others. But then, the final decision of circuit elements is yours, since only you know what your needs are.

There are several improvements on the basic circuit that come to mind almost immediately: They include using a crystal oscillator, adding a keyboard for speed-selection, or adding a digital display (which offers some interesting possibilities since all you have to do is count pulses and do a bit of arithmetic). Just as with all the circuits discussed in this column, remember that what we have described here is only a starting point.

You can elaborate on the circuit as much as you want to make it as versatile as you need. The only thing I ask is that you let me know what you've done—there's a lot of people out there who are interested. **R-E**