

Introduction to Pulse Width Modulation

[Michael Barr](#) - August 31, 2001

Introduction to Pulse Width Modulation

A look at a powerful technique for controlling analog circuits with a microprocessor's digital outputs.

Pulse width modulation (PWM) is a powerful technique for controlling analog circuits with a microprocessor's digital outputs. PWM is employed in a wide variety of applications, ranging from measurement and communications to power control and conversion.

Analog circuits

An analog signal has a continuously varying value, with infinite resolution in both time and magnitude. A nine-volt battery is an example of an analog device, in that its output voltage is not precisely 9V, changes over time, and can take any real-numbered value. Similarly, the amount of current drawn from a battery is not limited to a finite set of possible values. Analog signals are distinguishable from digital signals because the latter always take values only from a finite set of predetermined possibilities, such as the set {0V, 5V}.

Analog voltages and currents can be used to control things directly, like the volume of a car radio. In a simple analog radio, a knob is connected to a variable resistor. As you turn the knob, the resistance goes up or down. As that happens, the current flowing through the resistor increases or decreases. This changes the amount of current driving the speakers, thus increasing or decreasing the volume. An analog circuit is one, like the radio, whose output is linearly proportional to its input.

As intuitive and simple as analog control may seem, it is not always economically attractive or otherwise practical. For one thing, analog circuits tend to drift over time and can, therefore, be very difficult to tune. Precision analog circuits, which solve that problem, can be very large, heavy (just think of older home stereo equipment), and expensive. Analog circuits can also get very hot; the power dissipated is proportional to the voltage across the active elements multiplied by the current through them. Analog circuitry can also be sensitive to noise. Because of its infinite resolution, any perturbation or noise on an analog signal necessarily changes the current value.

Digital control

By controlling analog circuits digitally, system costs and power consumption can be drastically reduced. What's more, many microcontrollers and DSPs already include on-chip PWM controllers, making implementation easy.

In a nutshell, PWM is a way of digitally encoding analog signal levels. Through the use of high-resolution counters, the duty cycle of a square wave is modulated to encode a specific analog signal level. The PWM signal is still digital because, at any given instant of time, the full DC supply is either fully on or fully off. The voltage or current source is supplied to the analog load by means of a repeating series of on and off pulses. The on-time is the time during which the DC supply is applied to the load, and the off-time is the period during which that supply is switched off. Given a sufficient bandwidth, any analog value can be encoded with PWM.

Figure 1 shows three different PWM signals. Figure 1a shows a PWM output at a 10% duty cycle. That is, the signal is on for 10% of the period and off the other 90%. Figures 1b and 1c show PWM outputs at 50% and 90% duty cycles, respectively. These three PWM outputs encode three different analog signal values, at 10%, 50%, and 90% of the full strength. If, for example, the supply is 9V and the duty cycle is 10%, a 0.9V analog signal results.

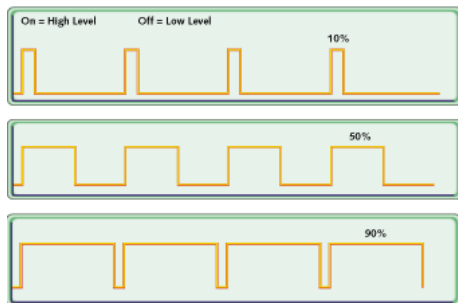


Figure 1: PWM signals of varying duty cycles

Figure 2 shows a simple circuit that could be driven using PWM. In the figure, a 9V battery powers an incandescent lightbulb. If we closed the switch connecting the battery and lamp for 50ms, the bulb would receive 9V during that interval. If we then opened the switch for the next 50ms, the bulb would receive 0V. If we repeat this cycle 10 times a second, the bulb will be lit as though it were connected to a 4.5V battery (50% of 9V). We say that the duty cycle is 50% and the modulating frequency is 10Hz.



Figure 2: A simple circuit

Most loads, inductive and capacitive alike, require a much higher modulating frequency than 10Hz. Imagine that our lamp was switched on for five seconds, then off for five seconds, then on again. The duty cycle would still be 50%, but the bulb would appear brightly lit for the first five seconds and off for the next. In order for the bulb to see a voltage of 4.5 volts, the cycle period must be short relative to the load's response time to a change in the switch state. To achieve the desired effect of a dimmer (but always lit) lamp, it is necessary to increase the modulating frequency. The same is true in other applications of PWM. Common modulating frequencies range from 1kHz to 200kHz.

Hardware controllers

Many microcontrollers include PWM controllers. For example, Microchip's PIC16C67 includes two, each of which has a selectable on-time and period. The duty cycle is the ratio of the on-time to the period; the modulating frequency is the inverse of the period. To start PWM operation, the data sheet suggests the software should:

- Set the period in the on-chip timer/counter that provides the modulating square wave.
- Set the on-time in the PWM control register.
- Set the direction of the PWM output, which is one of the general-purpose I/O pins.
- Start the timer.
- Enable the PWM controller.

Although specific PWM controllers do vary in their programmatic details, the basic idea is generally the same.

Communication and control

One of the advantages of PWM is that the signal remains digital all the way from the processor to the controlled system; no digital-to-analog conversion is necessary. By keeping the signal digital, noise effects are minimized. Noise can only affect a digital signal if it is strong enough to change a logical-1 to a logical-0, or vice versa.

Increased noise immunity is yet another benefit of choosing PWM over analog control, and is the principal reason PWM is sometimes used for communication. Switching from an analog signal to PWM can increase the length of a communications channel dramatically. At the receiving end, a suitable RC (resistor-capacitor) or LC (inductor-capacitor) network can remove the modulating high frequency square wave and return the signal to analog form.

PWM finds application in a variety of systems. As a concrete example, consider a PWM-controlled brake. To put it simply, a brake is a device that clamps down hard on something. In many brakes, the amount of clamping pressure (or stopping power) is controlled with an analog input signal. The more voltage or current that's applied to the brake, the more pressure the brake will exert.

The output of a PWM controller could be connected to a switch between the supply and the brake. To produce more stopping power, the software need only increase the duty cycle of the PWM output. If a specific amount of braking pressure is desired, measurements would need to be taken to determine the mathematical relationship between duty cycle and pressure. (And the resulting formulae or lookup tables would be tweaked for operating temperature, surface wear, and so on.)

To set the pressure on the brake to, say, 100psi, the software would do a reverse lookup to determine the duty cycle that should produce that amount of force. It would then set the PWM duty cycle to the new value and the brake would respond accordingly. If a sensor is available in the system, the duty cycle can be tweaked, under closed-loop control, until the desired pressure is precisely achieved.

PWM is economical, space saving, and noise immune. And it's now in your bag of tricks. So use it.

Michael Barr is editor in chief of *ESP*. He is also the author of *Programming Embedded Systems in C and C++* (O'Reilly, 1999) and an adjunct faculty member at the University of Maryland.

E-mail him at mbarr@cmp.com.

[Return to September 2001 Table of Contents](#)