

Pulse-width modulation for microcontroller servo control

NATHANIEL PINCKNEY

ANIMATRONICS ARE USED in the movie industry to bring aliens, dinosaurs, and man-eating sharks to life. Many mechanical devices are used to impart realism to these puppets. Large motors and hydraulics are employed to control limbs, necks, and other areas requiring high torque. Small motors are used to create facial expressions, control eyes, and provide fluidity of movement to add life to these robots.

A common motor used in animatronics is a servo motor. Instead of providing constant rotation, like most motors, servos are used for precise angular positioning but are often limited to only 180° of maximum rotation. Dozens of servos are used to control the facial expressions of a robotic puppet. Even a simple android might have five servos controlling just the eyes. The latex skin or muscles of an android's face are connected to the servo's shaft so that the servo can contort the face into different expressions. Servos are also common in devices such as radio-controlled cars to control steering, radio-controlled airplanes to control rudders, or even in the cruise control systems of cars. Servos are ideal for applications requiring absolute positioning of a motor shaft.

Microcontrollers are an excellent and inexpensive device for controlling servos. In order to properly control a servo with a microcontroller, it is necessary to learn a few techniques, such as properly generating a control signal for a servo, which the rest of this article will cover.

Servos may be purchased in a prepackaged form, or you can create



your own servo with a few common components. Internally, a servo can be thought of as a direct current (dc) motor (which rotates an external motor shaft but provides no way to determine the amount of rotation) with a built-in controller. The control circuitry compares an angular position, determined by a control signal, to the current position of the motor shaft (as shown in Fig. 1). The motor shaft's angular position is often determined by a potentiometer, which is rotated by the motor shaft. A potentiometer is a three-terminal resistor whose center connection has variable resistance, usually controlled by a slider or dial. The potentiometer acts as a variable voltage divider. The voltage from the center connection of the potentiometer represents the angular position the motor shaft is in. Other methods to determine angular position and rotation

exist for bigger servos, but a potentiometer is the most common for small servos. The built-in controller generates an internal signal from the voltage controlled by the potentiometer, compares it to the control signal, and then provides power to the dc motor to rotate the shaft in the appropriate direction to match the two. Servos usually require a pulse-width modulated control signal.

Pulse-width modulation

Often, when controlling an analog device, the ability to drive a signal with variable power ($P = I \times V$) is needed. For example, you may want to adjust the speed of a dc motor or dim a light-emitting diode (LED). This can be a challenge when the signal is generated by a digital device. Different methods to convert a digital signal to an analog signal exist, one of which is a digital-to-analog converter. Using a converter will add complexity to a project, so generating a variable power signal with existing circuitry is desirable to reduce the number of components.

An easy method to vary the power using a digital signal, when an analog signal isn't available, is by a method called pulse-width modulation (PWM). Instead of controlling the current or voltage of a signal, a pulse-width-modulated signal works by repeatedly pulsing the digital signal high and low at a fast rate. When sufficiently fast, the signal creates an effective average voltage. A shorter PWM period (the length between the rising edges) will create a cleaner average voltage, because the signal is effectively less "jittery" (i.e., less discharge from the capacitance in the line is needed to smooth the signal), but the minimum period will be limited by the speed of the device generating the signal. The period of the PWM signal is usually constant for a given application, and the high pulse width (the duration of the signal being driven high within one period) is usually variable, so that the average voltage of the signal can be changed. The ratio of high pulse width to period of the signal is called the *duty cycle*. By varying the duty cycle, you can vary the average voltage, as shown in Fig. 2.

The power through a device is proportional to the voltage supplied. Therefore, to decrease the power usage

of a device (to dim an LED or to slow a motor), the duty cycle of the PWM signal should be decreased. A PWM signal can be used to limit the power to a device to save energy. This technique is used in many portable devices which have limited battery power. For example, wave a device with an LED back and forth in your hand and you will often see a strobe light effect instead of a straight streak. This blinking is from the pulse-width-modulated signal turning the LED on and off. The blinking cannot normally be perceived by the human eye when the LED is stationary, since it is blinking at a rate faster than the eye can perceive. But when moving, the discrete flashes become visible because the LED is only lit up at certain positions as it moves.

Some devices, such as servos, do not rely on the power of the signal limited by PWM but instead use the width of the high pulses to transmit information. This is also used by infrared remote controls to transmit data to control a television or radio. Pulse-width-modulated signals may be generated from many digital devices, even ones as simple as an inexpensive timer integrated circuit (such as the 8-pin 555 timer). A versatile yet inexpensive solution for many robotics hobbyists is to use microcontroller for PWM generation. Using a microcontroller has the added advantage of containing all of the control circuitry (needed for analyzing

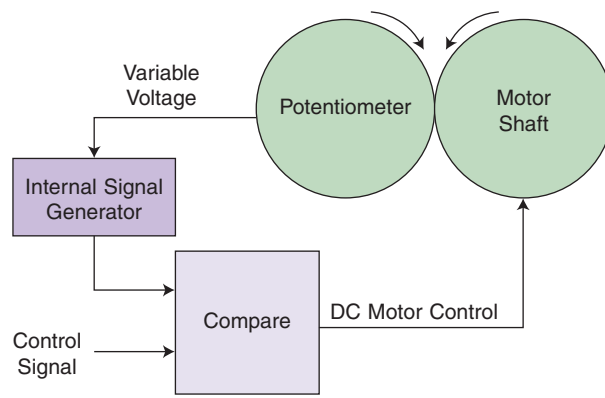


Fig. 1 The inside of a servo

ing and responding to input) for a simple robot on a single chip.

Signal generation with a PIC

Pulse-width-modulated signal generation is easy to implement on microcontrollers. All microcontrollers will be able to generate a PWM signal, but the more expensive and elaborate ones provide hardware to make PWM generation easier, freeing up more processor time to run other tasks. An inexpensive and easily obtainable microcontroller, such as the PICmicro 18F452, provides enough hardware to implement PWM generation using a few different methods, depending on the needs of the application.

The simplest but most processor-intensive method to create a PWM signal is manually comparing a “count” to a variable that describes how long the high pulse width should be. When the count is less than the pulse width variable, the PWM signal is driven high;

otherwise, it is driven low. After the PWM period has elapsed, the count can be reset and the process started over. The PWM period will be the same as the time it takes the PIC to run your code. To increase the length of the PWM period, loops can be used to create delay. This method will work on the simplest of PICs but leaves little processor time to do anything else. A couple variations exist and can be used if the PIC includes the necessary hardware. A timer can be used and polled at intervals in between code to

determine if the PWM signal needs to be updated. Another solution is to have the PIC update the PWM signal on a timer overflow interrupt. This guarantees that the PWM signal will be updated at specific intervals, while most of the processor time can be used for other tasks.

The least processor-intensive method is to use a built-in PWM module if your microcontroller has one. When enabled, the PWM module will automatically generate a PWM signal with a period and duty cycle specified in control registers on the PIC. Depending on the microcontroller you are using and the speed it is running at, the built-in PWM module might not support a large enough period needed for the device you are using, such as for a servo motor (which commonly has a period of 20 ms). In that case, you will want to use one of the previously mentioned methods to generate a longer PWM signal.

Controlling a servo

Most servos, including the Hitec RCD USA, Inc. HS-322HD demonstrated here, have three pins: power, ground, and a control signal. The control signal is a pulse-width-modulated input signal whose high pulse width determines the servo’s angular position, as shown in Fig. 3. Internally, the servo compares the PWM control signal to an internally generated signal, whose pulse widths are controlled by the potentiometer (which determines the shaft angle) and matches the pulse widths by rotating the motor shaft. For the HS-322HD, power can be between 4.8 Vdc (volts of direct current) and 6 Vdc. Since the control signal (which draws a

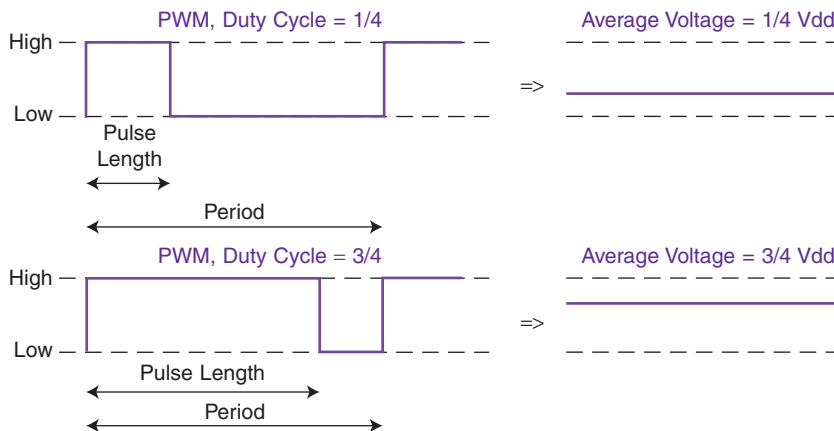


Fig. 2 Pulse width modulation and average voltage

maximum of about 20 mA) does not drive the motor directly, an additional benefit of using a servo is that the PIC (whose output pins can drive up to 25 mA) *can* drive the control signal directly. Most motors draw more than 25 mA of current for operation and, therefore, must be indirectly connected to the PIC through a current amplifying device, like an H-bridge or a transistor.

Typically, servos require a PWM signal with a 20-ms period and a pulse width between 0.9–2.1 ms (0.9 ms corresponds to the minimum angle and 2.1 ms corresponds to the maximum angle); therefore, the middle position is 1.5 ms (the average of the minimum/maximum pulse widths). The HS-322HD has a maximum angle of 180°. Servos only move a finite angular amount per cycle of the signal, so multiple cycles must be sent before the servo arrives at the correct angle. The speed/power at which the servo moves to a new position is proportional to the distance it needs to travel. So as the servo becomes closer to the target angle, it will gradually slow. The servo will resist change away from the designated angle as long as a signal is applied. Note that the servo's control mechanism will only engage when a

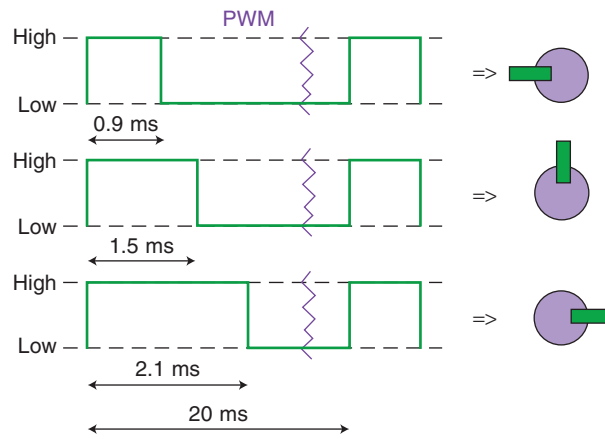


Fig. 3 Controlling an HS-322HD servo with PWM

signal is applied. If there is no signal, the servo's motor shaft is not driven by any circuitry and, hence, can be rotated freely, even when power is supplied to the servo.

Conclusions

Microcontrollers offer a simple and inexpensive solution for controlling servo motors for robotics and other electronics projects. Through the use of PWM, the angular position of the servo motor shaft can be conveniently controlled by a microcontroller for a variety of projects. PWM is an easy solution for the control of analog devices in

other projects as well. Depending on the features included with the microcontroller you are using, a PWM signal can be generated in a variety of ways.

Additional resources

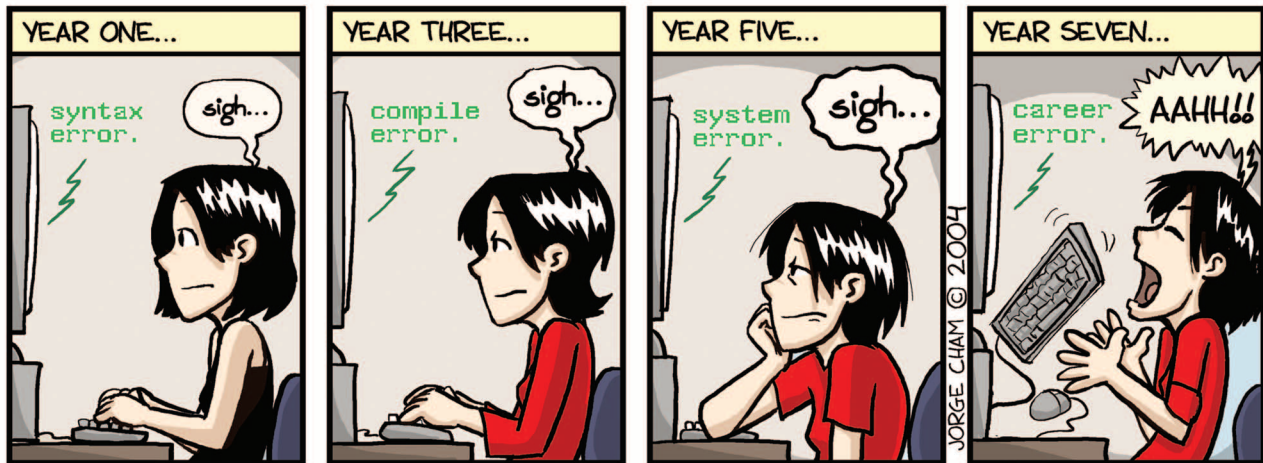
This article was written jointly with the development of a set of tutorials on how to use different devices with the Microchip Technology's PIC 18F452 microcontroller and Xilinx's Spartan 3 FPGA. Five undergraduate students and two professors undertook the development of the tutorials, appropriately named MicroToys guides. More information about PIC-controlled servos, as well as

controlling other devices with a PIC microcontroller, can be found at Harvey Mudd College's E155 MicroToys Web site at <http://www4.hmc.edu:8001/Engineering/microtoys/>.

About the author

Nathaniel Pinckney is a first-year undergraduate engineering student currently attending Harvey Mudd College. Before attending Harvey Mudd, he was home schooled since second grade. He has been involved in the development of various GNU/Linux software programs and is interested in parallel embedded architecture design.

RESIGNATION: THE EVOLUTION OF THE SIGH



www.phdcomics.com