|  |
| --- |
| Guide to PWM and PPM |

## PWM

The Fading example demonstrates the use of analog output (PWM) to fade an LED. It is available in the File->Sketchbook->Examples->Analog menu of the Arduino software.

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to [analogWrite](http://arduino.cc/en/Reference/AnalogWrite)() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.



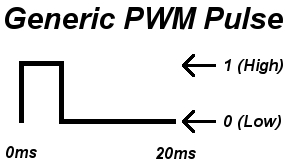
Once you get this example running, grab your arduino and shake it back and forth. What you are doing here is essentially mapping time across the space. To our eyes, the movement blurs each LED blink into a line. As the LED fades in and out, those little lines will grow and shrink in length. Now you are seeing the pulse width.

PWM and PPM are two common words used in the R/C industry. PWM stands for Pulse Width Modulation and PPM stands for Pulse Position Modulation. Some devices that use PWM for control are ESC's (electronic speed controls) and servos. PWM is a technique used to relay data in the form of a varying pulse width.

You may be already familiar with binary, 1's and 0's; where a 1 is represented as 'on' and a 0 as 'off'. An example of this would be a light switch. Turning the switch on would indicate a 1, off a 0. In the case of a PWM/PPM signal, a voltage applied indicates a 1 and vice versa. However, in the case of R/C electronics this 'on/off' data is not enough, this is where the pulse width comes in.

The way we relay data to a servo for instance is the time the pulse is on. In the case of R/C electronics this time is usually around 1-2 milliseconds. A servo or ESC will monitor this pulse and begin counting when the pulse is detected and stop counting when the pulse stops. The time the pulse is on will determine the servo position. For example, sending a servo a 1ms pulse will make the servo swing completely left while a 2ms pulse will swing the arm completely right.

Generally in R/C equipment an entire PWM pulse will last a total of 20ms. The entire pulse is called a frame. A complete frame will include both the time the pulse is high (1-2ms) and the time the pulse is low. The image below represents a typical PWM frame.



Although the frame lasts 20ms the important part of the pulse is the time the pulse is on; 1-2ms. Although the time between pulses is not as important it does play an important role. Usually keeping the time between pulses around 20ms is best. If the delay is longer, a servo for example will lose holding power. A pulse can be generated much faster but 20ms is best for most situations.

RC transmitters use PWM or PPM for transferring digital signal. PWM stands for Pulse Width Modulation and PPM stands for Pulse Position Modulation. PWM is a technique used to relay data in the form of a varying pulse width. In PPM (Pulse Position Modulation) the analogue sample values determine the position of a narrow pulse relative to the clocking time.

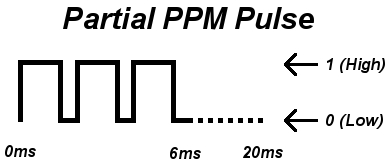
RC Devices that use PWM Pulses:

* Servos
* Electronic Speed Controllers
* R/C receivers
* Data loggers
* Autopilot/Stabilization systems
* Servo Controller

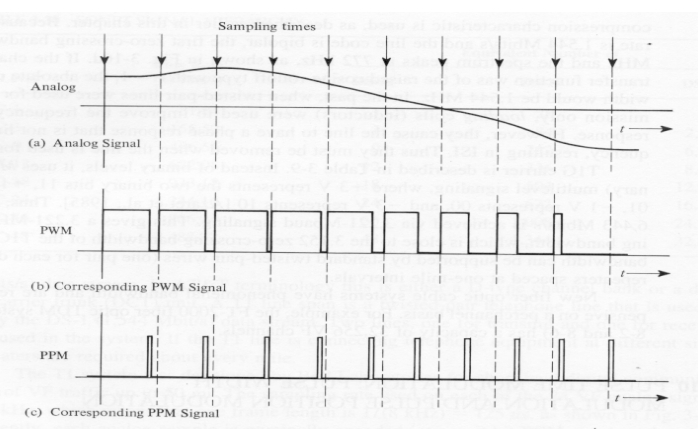
RC Devices that use PPM Pulses:

* R/C transmitters
* R/C receivers
* Autopilot/Stabilization systems
* PCTx

So what is the point in waiting up to 20ms? This is an R/C specific and will help understand PPM. Again, PPM stands for Pulse Position Modulation. PPM basically is several PWM signals lined up back to back. A PPM frame looks like this:



Posted on [November 5, 2013](http://blog.oscarliang.net/pwm-ppm-difference-conversion/) by [Oscar](http://blog.oscarliang.net/author/oscar/)

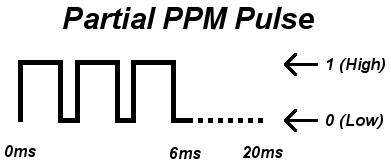
\[](http://blog.oscarliang.net/wp-content/uploads/2013/11/pwm-ppm-signal-example.jpg)

# Pulse Width Modulation

In PWM each RC channel has it’s own cable. If we want 9 channels we must wrie all 9 cables along with the power and ground. The value of each channel is represented as a 1 millisecond (ms) to 2ms “ON” signal and this signal repeats (or updates) every 20 milliseconds. It goes high for the 1-2ms, then it falls to Low. The length of time it is high is the value for that channel. We see this in the GUI directly as 1000-2000, so we are seeing the raw ON time in microseconds.

# Pulse Position Modulation (PPM)

PPM basically is several PWM signals lined up back to back. In PPM, the same signaling is used but each channel is sent successively, then a delay, then it loops back to channel 1.

[](http://blog.oscarliang.net/wp-content/uploads/2013/11/ppm-signal.jpg)

In normal PWM there are 50 updates sent per second (50Hz) which means each update takes 20 milliseconds. So if each channel takes up to 2ms, then we can do 10 channels within that 20ms before which we need to loop back to ch1. So we don’t even have a downside of multiplexing all the channels down to 1 wire and yet we have less wiring, and in our case 2 extra AUX channels! If you can do PPM, it’s better!

However PPM is not the most popular because many radios don’t support PPM. If you don’t have a PPM transmitter and receiver you can use a little device that converts between regular PWM and PPM. The MultiWii supports PPM input. Just define the appropriate option SERIAL\_SUM\_PPM in config.h and wire the PPM signal into the Throttle channel. The rest of the transmitter inputs like roll, pitch, and yaw are now free for other things. It is also possible to obtain PPM from PWM by using a mono-stable multivibrator circuit.