initialize(period)  
You must call this method first to use any of the other methods. You can optionally specify the timer's period here (in microseconds), by default it is set at 1 second. Note that this breaks analogWrite() for digital pins 9 and 10 on Arduino.

setPeriod(period)  
Sets the period in microseconds. The minimum period or highest frequency this library supports is 1 microsecond or 1 MHz. The maximum period is 8388480 microseconds or about 8.3 seconds. Note that setting the period will change the attached interrupt and both pwm outputs' frequencies and duty cycles simultaneously.

pwm(pin, duty, period)  
Generates a PWM waveform on the specified pin. Output pins for Timer1 are PORTB pins 1 and 2, so you have to choose between these two, anything else is ignored. On Arduino, these are digital pins 9 and 10, so those aliases also work. Output pins for Timer3 are from PORTE and correspond to 2,3 & 5 on the Arduino Mega. The duty cycle is specified as a 10 bit value, so anything between 0 and 1023. Note that you can optionally set the period with this function if you include a value in microseconds as the last parameter when you call it.

attachInterrupt(function, period)  
Calls a function at the specified interval in microseconds. Be careful about trying to execute too complicated of an interrupt at too high of a frequency, or the CPU may never enter the main loop and your program will 'lock up'. Note that you can optionally set the period with this function if you include a value in microseconds as the last parameter when you call it.

setPwmDuty(pin, duty)  
A fast shortcut for setting the pwm duty for a given pin if you have already set it up by calling pwm() earlier. This avoids the overhead of enabling pwm mode for the pin, setting the data direction register, checking for optional period adjustments etc. that are mandatory when you call pwm().

detachInterrupt()  
Disables the attached interrupt.

disablePwm(pin)  
Turns PWM off for the specified pin so you can use that pin for something else.

read()  
Reads the time since last rollover in microseconds.

Method Detail - calling conventions for all public methods

void initialize(long microseconds=1000000);  
void start();  
void stop();  
void restart();  
unsigned long read();  
void setPeriod(long microseconds);  
void pwm(char pin, int duty, long microseconds=-1);  
void setPwmDuty(char pin, int duty);  
void disablePwm(char pin);  
void attachInterrupt(void (\*isr)(), long microseconds=-1);  
void detachInterrupt();