


Arduino-PWM-Frequency (/Arduino-PWM-Frequency)

 Editar 0 (/page/messages/Arduino-PWM-Frequency) 6 (/page/history/Arduino-PWM-Frequency)

... (/page/menu/Arduino-PWM-Frequency)

Changing PWM Frequency on the Arduino

Credits: <http://usethearduino.blogspot.com>

Atmega328 pins 12, 11, 15, 16, 17, and 5 can be configured for PWM output. On the Arduino Duemilanove/UNO etc., these are pins 6, 5, 9, 10, 11, and 3 respectively.

The 8-bit PWM value that you set when you call the `analogWrite` function:

```
analogWrite(myPWMpin, 128);
```

 Outputs a square wave

is compared against the value in an 8-bit counter. When the counter is less than the PWM value, the pin outputs a HIGH; when the counter is greater than the PWM value, the pin outputs a LOW. In the example above, a square wave is generated because the pin is HIGH from counts 0 to 127, and LOW from counts 128 to 255, so it is HIGH for the same amount of time it is LOW.

It follows logically that the frequency of the PWM signal is determined by the speed of the counter. Assuming you are using

an Atmega168 with the Arduino Diecimila bootloader burned on it (which is exactly what you are using if you bought an Arduino Diecimila), this counter's clock is equal to the system clock divided by a prescaler value. The prescaler is a 3-bit value stored in the three least significant bits of the Timer/Counter register: CS02, CS01, and CS00. There are three such Timer/Counter registers: TCCR0B, TCCR1B, and TCCR2B.

Since there are three different prescalers, the six PWM pins are broken up into three pairs, each pair having its own prescaler. For instance, Arduino pins 6 and 5 are both controlled by TCCR0B, so you can set Arduino pins 6 and 5 to output a PWM signal at one frequency. Arduino pins 9 and 10 are controlled by TCCR1B, so they can be set at a different frequency from pins 6 and 5. Arduino pins 11 and 3 are controlled by TCCR2B, so they may be set at a third frequency. But you can't set different frequencies for pins that are controlled by the same prescaler (e.g. pins 6 and 5 must be at the same frequency).

If you use the default values set by the Arduino Diecimila's bootloader, these are your PWM frequencies:

Arduino Pins 5 and 6: 1kHz

Arduino Pins 9, 10, 11, and 3: 500Hz

How do you change the PWM frequency?

In the `void setup()` part of your Arduino code, set or clear

the CS02,CS01, and CS00 bits in the relevant TCCRnB register. For example, if you want to set pins 5 and 6 to output a PWM signal at the highest possible frequency (see "List of Possible Frequencies" at the end of this post), insert the following code in the `void setup()` section of your Arduino code:

First clear all three prescaler bits:

```
int prescalerVal = 0x07; create a variable called  
prescalerVal and set it equal to the binary  
number "00000111"
```

```
TCCR0B &= ~prescalerVal; AND the value in TCCR0B  
with binary number "11111000"
```

Now set the appropriate prescaler bits:

```
int prescalerVal = 1; set prescalerVal equal to  
binary number "00000001"
```

```
TCCR0B |= prescalerVal; OR the value in TCCR0B  
with binary number "00000001"
```

The above code cleared bits CS02 and CS01, and set bit CS00.

Another way to do this, is with the following code:

First clear the three prescaler bits

```
TCCR0B &= ~(1<<cs02);> Invert that to get
"11111011". AND that with TCCR0B.
TCCR0B &= ~(1<<cs01);>
TCCR0B &= ~(1<
```

```
//Now set the appropriate prescaler bits
```

✖ Try Wikispaces Classroom now. (<http://www.wikispaces.com/t/y/classroom-switch/banner/0/>)
Brand new from Wikispaces.

If the above code is confusing, try the Bit math explanation here: <http://www.arduino.cc/playground/Code/BitMath>

List of Possible Frequencies:

For pins 6 and 5 (OC0A and OC0B):

If TCCR0B = xxxxx001, frequency is 64kHz

If TCCR0B = xxxxx010, frequency is 8 kHz

If TCCR0B = xxxxx011, frequency is 1kHz (this is the default from the Diecimila bootloader)

If TCCR0B = xxxxx100, frequency is 250Hz

If TCCR0B = xxxxx101, frequency is 62.5 Hz

For pins 9, 10, 11 and 3 (OC1A, OC1B, OC2A, OC2B):

If TCCRnB = xxxxx001, frequency is 32kHz

If TCCRnB = xxxxx010, frequency is 4 kHz

If TCCRnB = xxxxx011, frequency is 500Hz (this is the default from the Diecimila bootloader)

If TCCRnB = xxxxx100, frequency is 125Hz

If TCCRnB = xxxxx101, frequency is 31.25 Hz

[Ayuda](#) · [Acerca de](#) · [Blog](#) · [Precio](#) · [Privacidad](#) · [Términos](#) · **[Apoyo](#)**

Contributions to <http://arduino-info.wikispaces.com/> are licensed under a Creative Commons Attribution

Non-Commercial 3.0 License. 

Portions not contributed by visitors are Copyright 2014 Tangient LLC