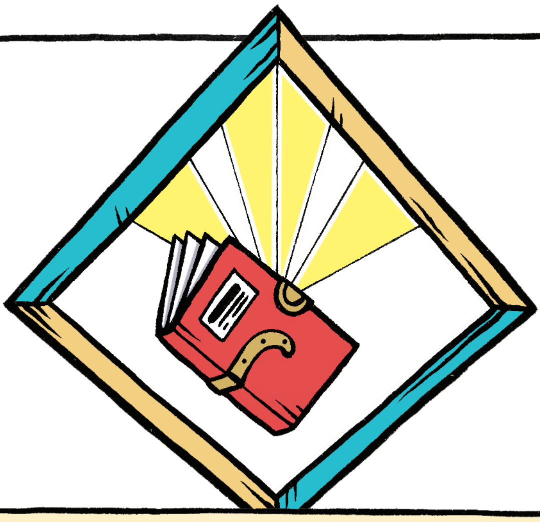
KNOWLEDGE

ASSOCIATED SHEETS

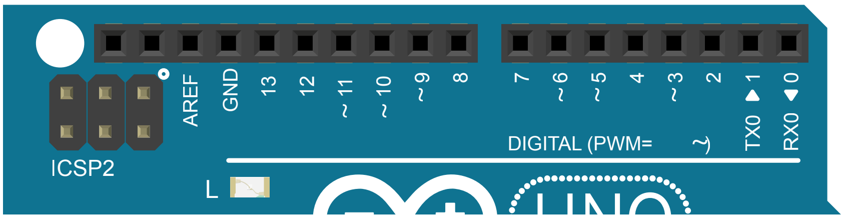
Digital Outputs
Control the lights



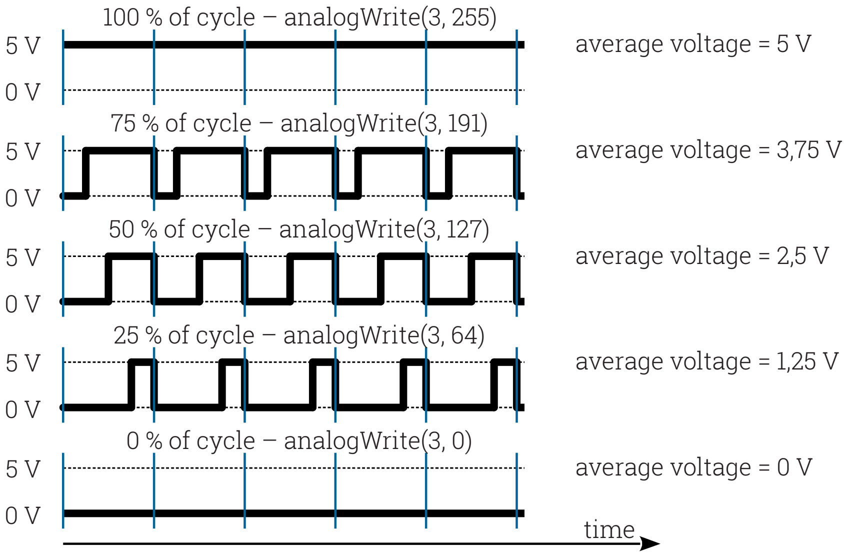
# PWM OUTPUTS

## Understand “pseudo analog” outputs

An analog output is a controllable voltage source. The Arduino board does not have a real analog output capable of producing a voltage value chosen by the user. Certain digital ports can, however, simulate a digital output by using the PWM (Pulse Width Modulation) technique: it concerns ports 3, 5, 6, 9, 10, and 11 (marked with a ~ on the board).



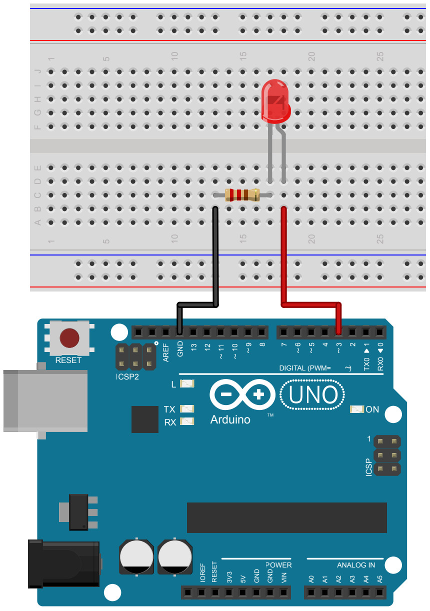
These ports can switch rapidly between their LOW (0 V) state and HIGH (5 V) states. By controlling the relationship between the time spent in the HIGH state compared to the time spent in the LOW state, the Arduino board can vary the average voltage value on the ports between 0 and 5 volts. The diagram below illustrates the principal representing the variations over time of the voltage of port 3 using the PWM output for different required voltages.

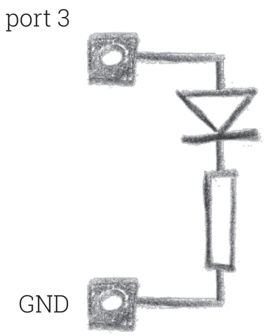


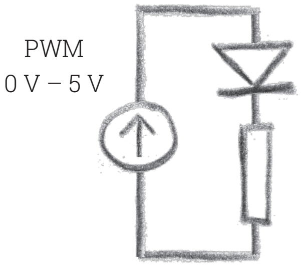
# KNOWLEDGE – **PWM OUTPUTS**

## Usage examples

Be careful: Unlike a real digital output, the PWM does not deliver a constant voltage but a voltage that is constantly oscillating. For certain applications, a PWM output is perfectly adequate; for example, to power a motor or to vary the intensity of a LED. The following circuit allows one to vary the power of a LED based on the value required at port 3 defined by the program as analog output.







If one needs a real continuous analog voltage, one must install a low-pass filter that will eliminate the high frequencies and only keep the average value (the frequency of oscillation of the PWM output is around 500 to 1000 Hz depending on the ports).

## Programming

The board can control the ratio of the PWM cycle with a sensitivity of 8 bits: the command in the "analogWrite" instruction is an integer of between 0 and 255 (= $2^8$ - 1), corresponding to an average voltage of between 0 and 5 volts.

```
int pwmValue ;                   // integer type variable
float VoltageInVolts = 3.1 ;     // decimal type variable
pinMode(3, OUTPUT) ;       // defines digital port3 as output
                           // put this instruction into the setup()

analogWrite(3, 0) ;        //  Creates an average voltage of 0 V on port 3
analogWrite(3, 127) ;      //  Creates an average voltage of 2.5 V on port 3
analogWrite(3, 255) ;      //  Creates an average voltage of 5 V on port 3

pwmValue = VoltageInVolts * 255 / 5 ;

analogWrite(3, pwmValue) ;  //  Creates an average voltage on port 3
                            // of value VoltageEnVolts
```