

Analog Input

Analog Output

The Arduino has an ADC but doesn't have a DAC (digital to analog converter).

However, it can emulate this using pulse-width modulation. (Note only some specific pins can use PWM.)

PWM

In order to respond to a PWM signal, a device must have some characteristic that allows it to average the energy applied to it.

With a 50% duty cycle, the signal is on half the time and off half the time – it spends the same amount of time high as it does low in each cycle.

So the device sees about half the power. With an LED, the LED can flash very fast, but our eyes can't keep up – with a 50% duty cycle, we see it as about half as bright. This is persistence of vision – you can look up some cool things on youtube.

A 10% duty cycle means the signal spends 10% of the time in the on phase.

You can also configure the frequency of the signal in the Arduino, but only between a couple of preset values.

For controlling the duty cycle from code, we write out a value between 0 and 255 – at 0 the signal is always low, at 255 the signal is always high.