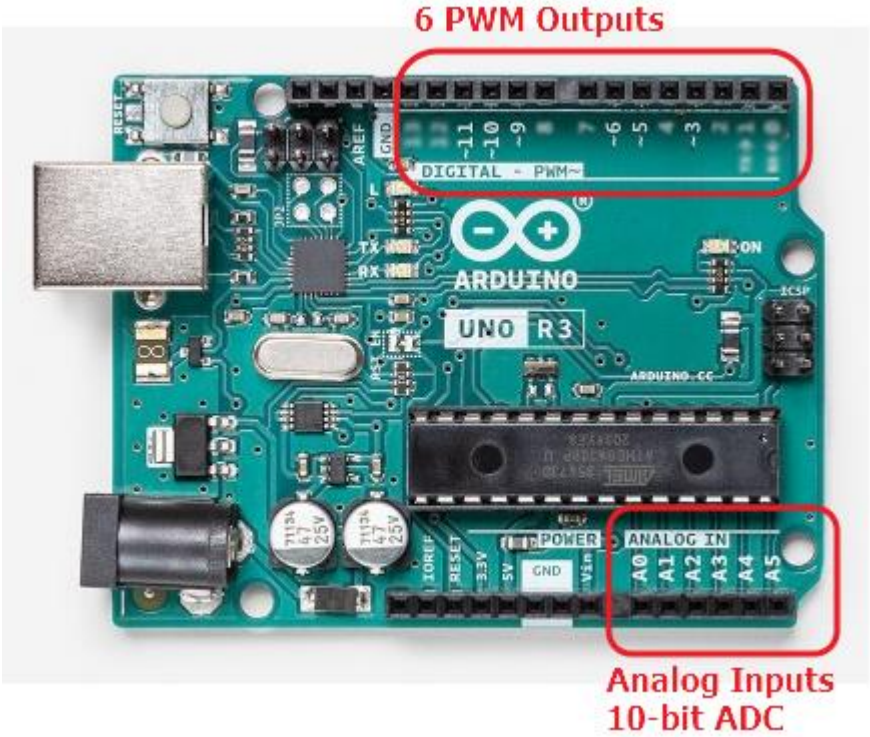
EP1000

Digital I/O

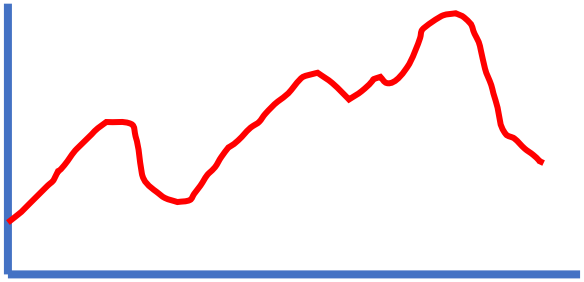
Analog I/O



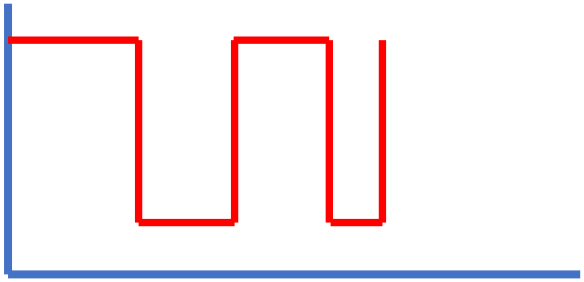
- The Uno uses the ATMega328 processor, which has 6 analog input pins.
- Each analog input has a 10-bit analog-to-digital converter that can produce an equivalent binary value for an analog voltage between 0 and Vref.

- Analog output is done using Pulse Width Modulation which can be used to control LEDs and Motors.
- Pins that can perform PWM are denoted with a ~
(Pins 3, 5, 6, 9 , 10, 11)

Analog vs Digital



- Analog signals are continuous.
- Analog signals require conversion (ADC) before processing
- Analog signals are real world.



- Digital values are discrete e.g. 0, Vcc
- Digital values are easy to process.
- Digital signals need to be converted to appear to be real world signals.

Reading Analog Signals

- Analog signals need to be converted to digital values before they can be processed.
- Analog-to-Digital Conversion required
  - Sampling (at least 2X input signal frequency)
  - Vref
  - Timing

References:
YouTube:
- All About Electronics: Introduction to ADC and DAC
- Great Scott: Electronic Basics #27: ADC (Analog to Digital)
- Embedds: ATMega328 ADC – Raw Mode

Simplified ADC with Arduino System

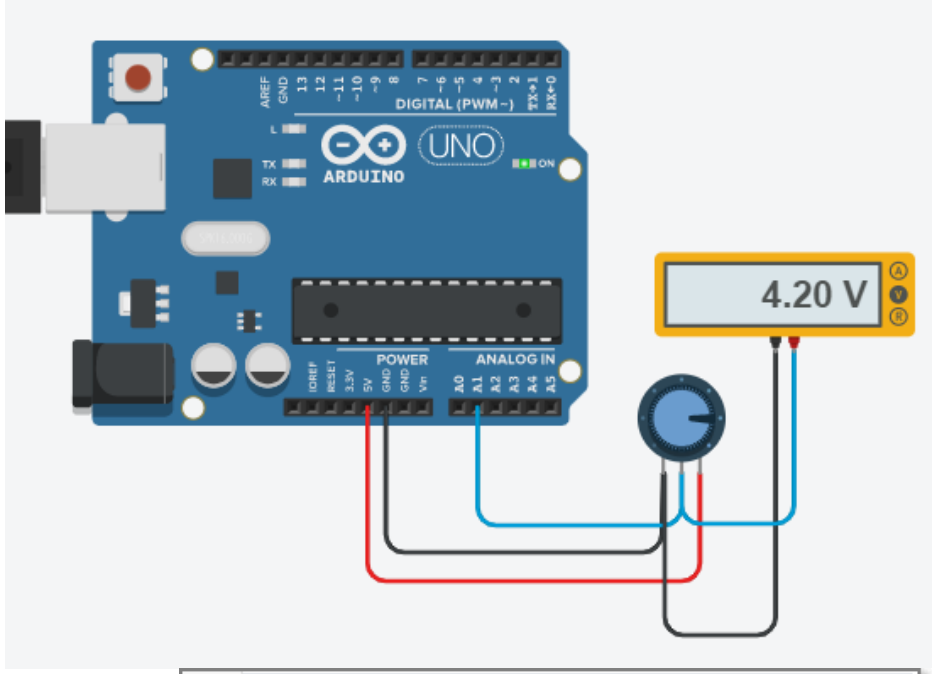
- Assumes that input signals are stable and does not change quickly.
- Uses a default Vref = 5V
- Resolution = $5/2^{10}$ = 4.9mV
- Max Read speed = 100 mS
  = 0.001s
- Result is between 0 ~ 1023

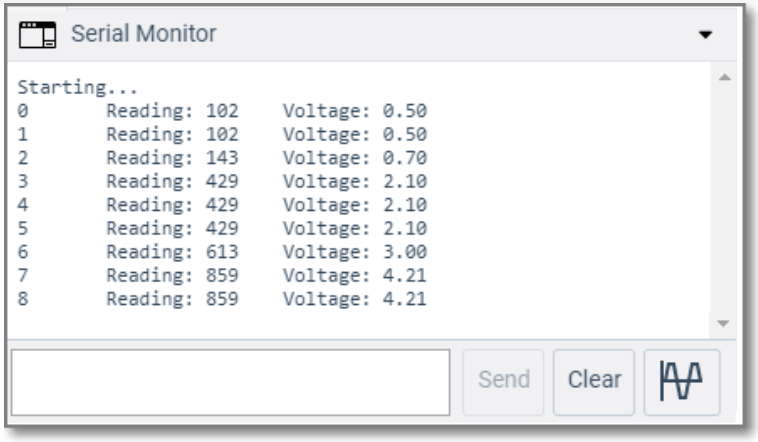
Analog I/O

analogRead(pin)

where
pin = A0..A5

Reading a Potentiometer






```
int idx = 0;

void setup()
{
  Serial.begin(9600);
  Serial.println("Starting...");
}

void loop()
{
  // read the ADC
  int value = analogRead(A1);
  Serial.print(idx);
  Serial.print("\tReading: ");
  Serial.print(value);
  Serial.print("\tVoltage: ");
  float volts = value * 0.0049;
  Serial.println(volts);
  delay(1000);
  ++idx;
}
```

TinkerCAD: [Reading a Potentiometer]

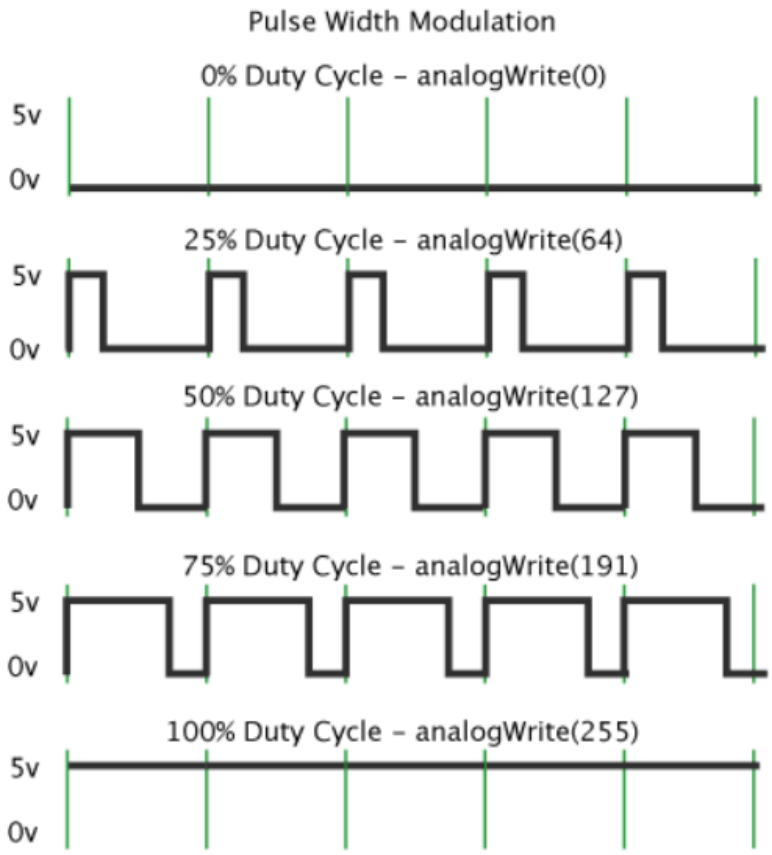
Serial Plotter

- You can use the Serial Plotter to plot a graph of your data values.
- To see the actual effects, you should use a physical board and the Serial Plotter.
- Ref: YouTube Elecktor TV: How to use the Serial Plotter

NB: TinkerCAD has a Serial plotter but no labels

Analog Output (PWM)

- Pulse Width Modulation (PWM) is a method of reducing the average power delivered by chopping it up into discrete parts.
- Usually a square wave is used, and by varying the ON-time vs the OFF-time, we can control the average power delivered.
- Hence, a 50% duty cycle produces 50% less power than a DC signal.
- Applications:
  - DC Motor speed control
  - LED light dimmer
  - Communications



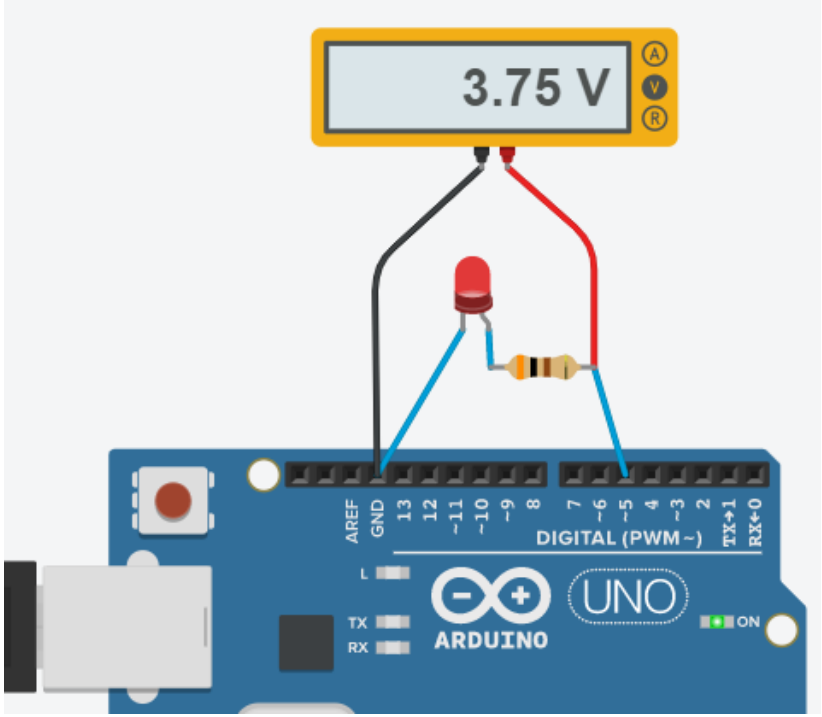
analogWrite()

- Uno has 6 digital output pins (3, 5, 6, 9, 10, 11) for PWM
- These pins are designated with a '~'
- You do not need to call pinMode() to use PWM.
- You can change the PWM from 0 to 255 effectively giving 256 levels of 5/256 V or 0.0195 V per level of output.

Analog I/O

- analogWrite()

LED fader




```
1
2  const int LED = 5;    // digital pin
3  const int STEP = 10;  // increments
4
5  void setup()
6  {
7  }
8
9  void loop()
10 {
11   // increase light intensity
12   for(int i=0; i < 255; i=i+STEP){
13     analogWrite(LED, i);
14     delay(200);
15   }
16   // reduce light intensity
17   for(int i=255; i > 0; i=i-STEP){
18     analogWrite(LED, i);
19     delay(200);
20   }
21 }
22
```

Using PWM to brighten and fade the LED gradually.

EP1000

Digital I/O

End