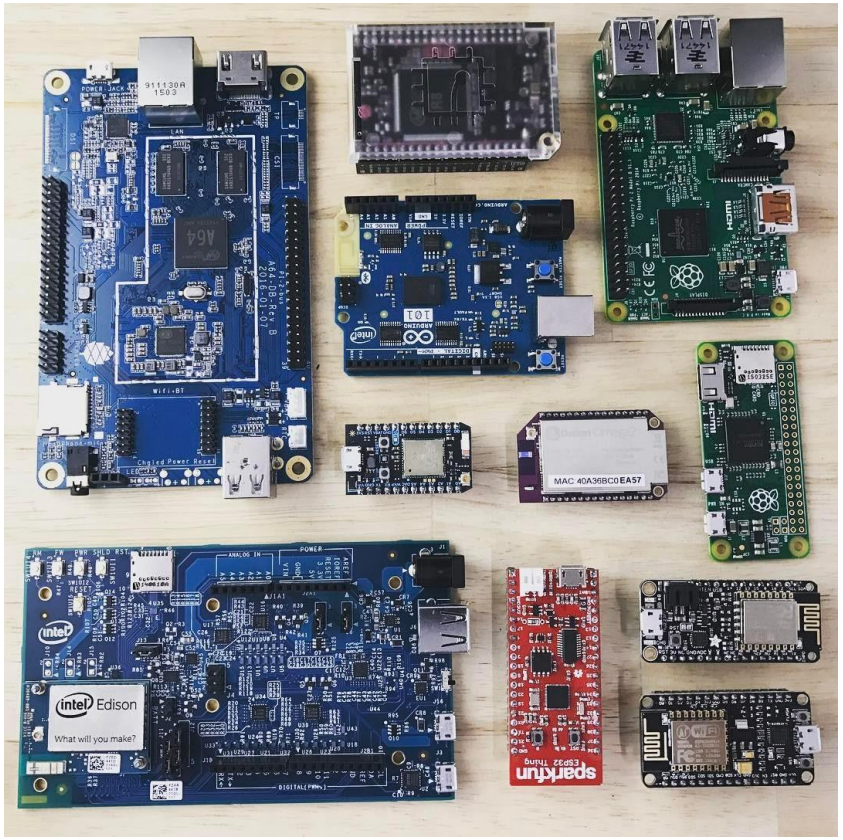
Arduino

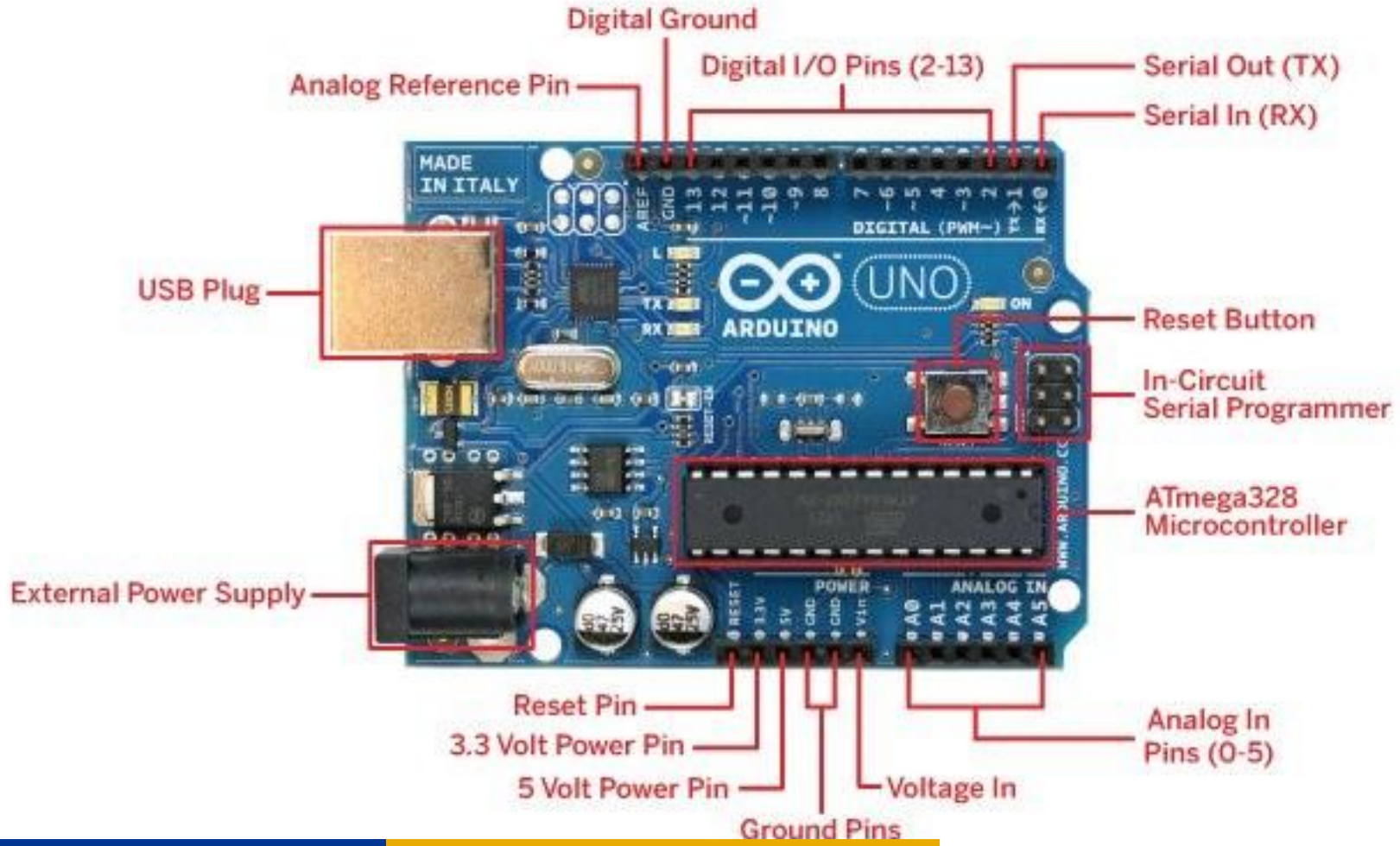
Aggie Innovation Space

What is a Microcontroller?



- A small, self-contained computer
- Hardware interface with real world
- Can be programmed

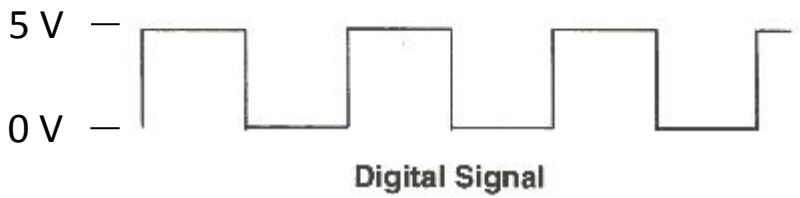
Arduino Uno


Digital Ground
Analog Reference Pin
Digital I/O Pins (2-13)
Serial Out (TX)
Serial In (RX)
USB Plug
Reset Button
In-Circuit
Serial Programmer
ATmega328
Microcontroller
External Power Supply
Reset Pin
3.3 Volt Power Pin
5 Volt Power Pin
Ground Pins
Voltage In
Analog In
Pins (0-5)
MADE
IN ITALY
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN

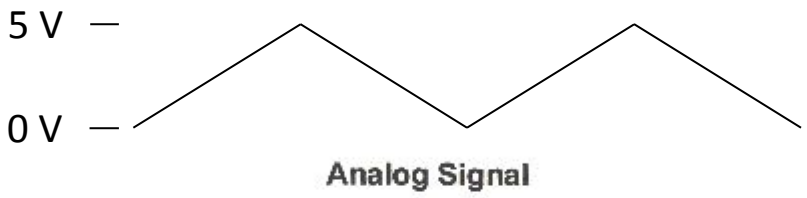
Analog vs Digital

Analog: Represented by a continuously variable quantity. Infinite possible values.

Digital: Only two discrete states: 0 and 1, on and off. (Binary)



Digital Signal



Analog Signal

Analog and Digital Output

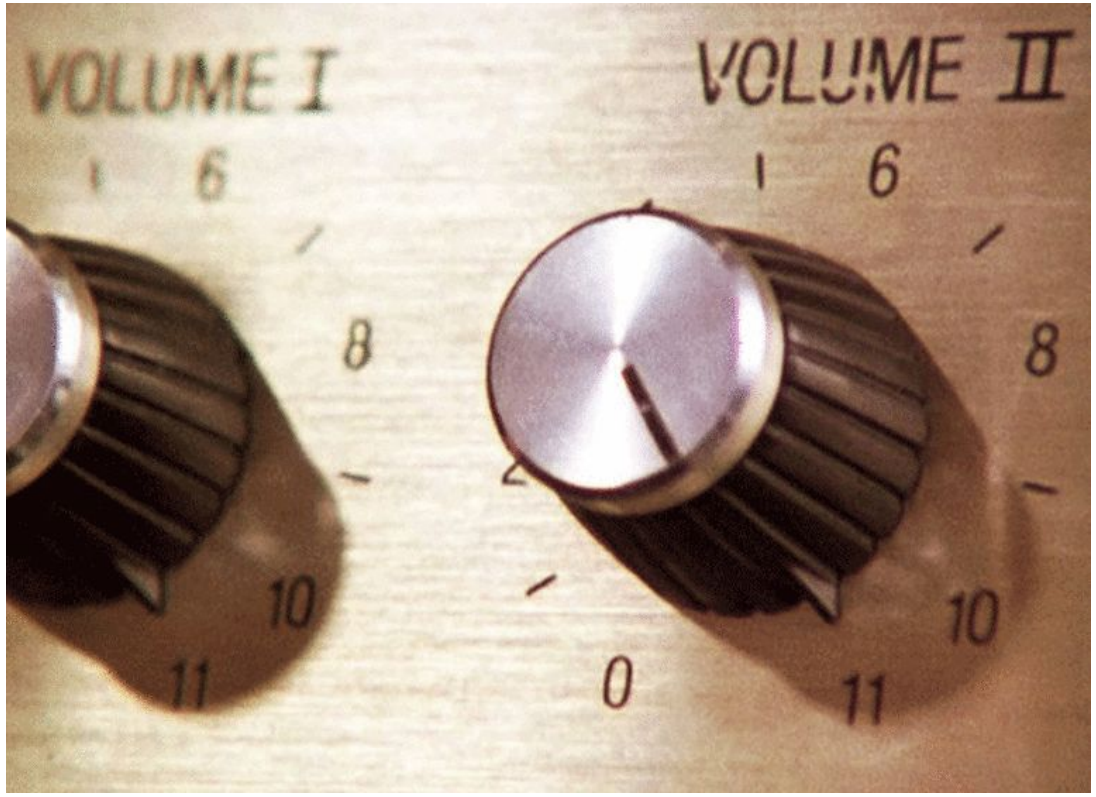
Digital Output

Analog Output

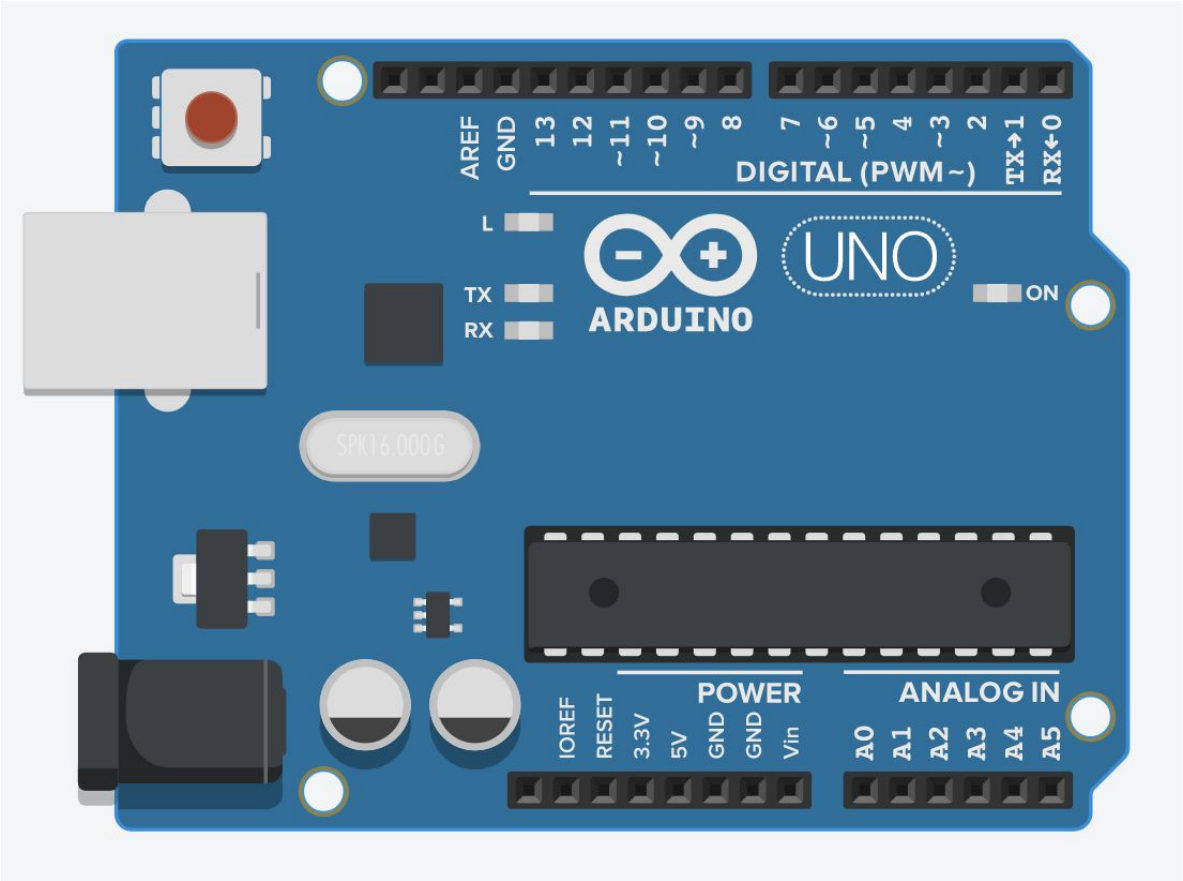
Analog and Digital Input

Digital Input

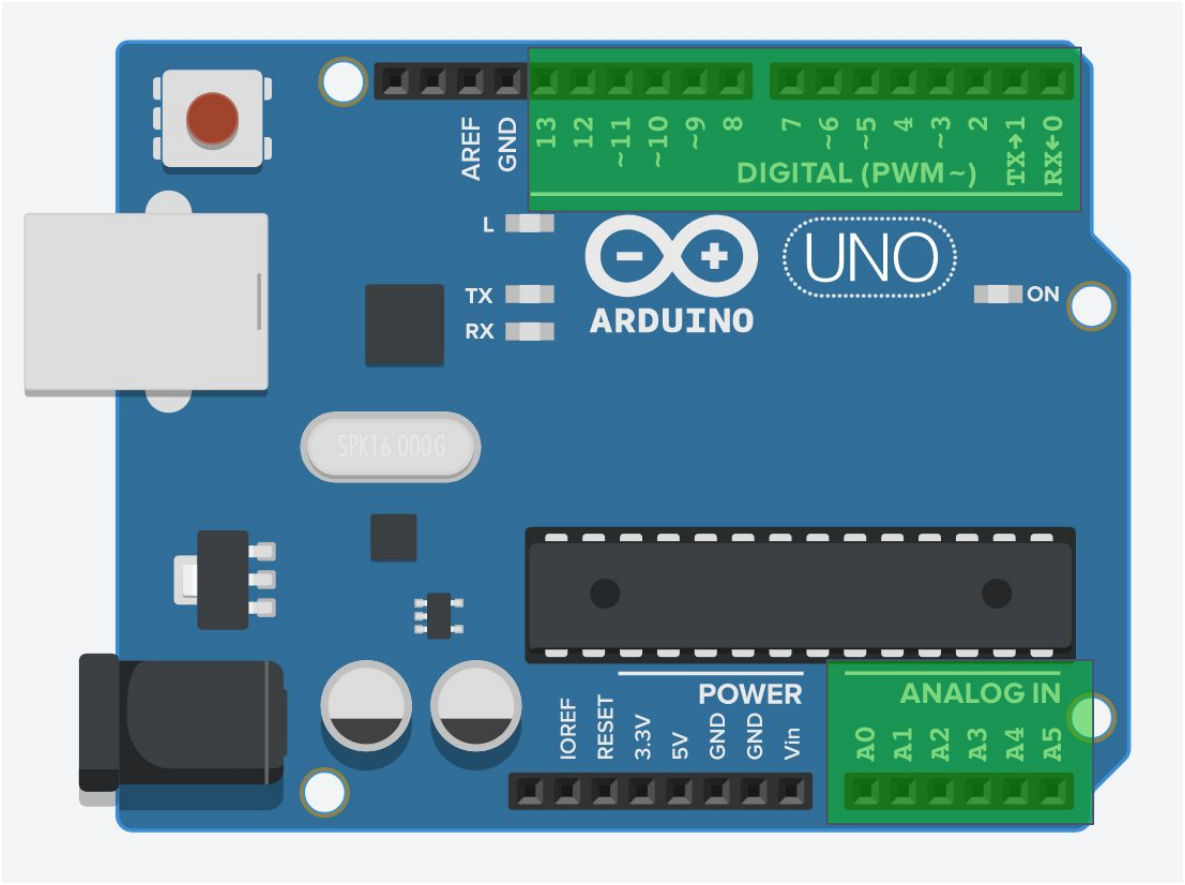
Analog Input



Digital and Analog Pins on Arduino Uno



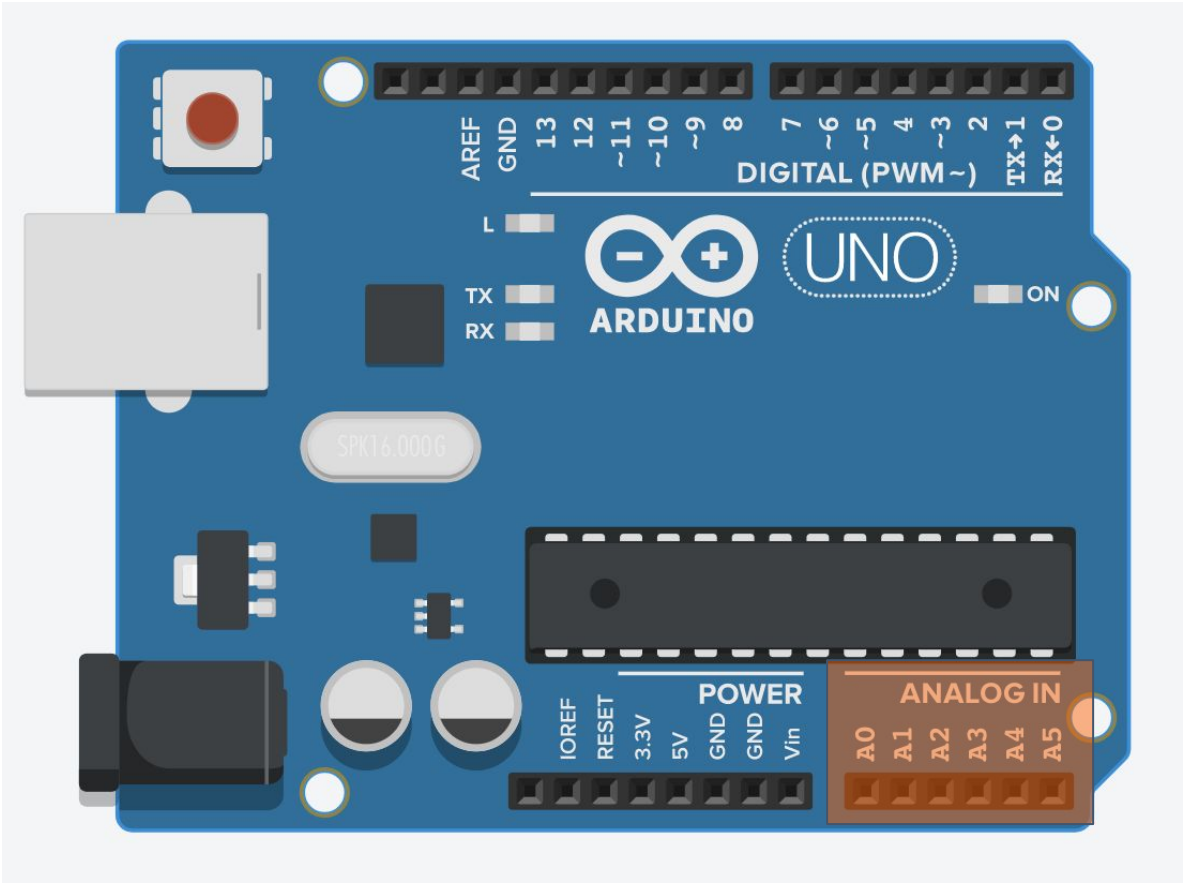
Digital and Analog Pins on Arduino Uno



Digital Output and Digital Input:

ALL pins can be used for digital input and digital output.

Digital and Analog Pins on Arduino Uno

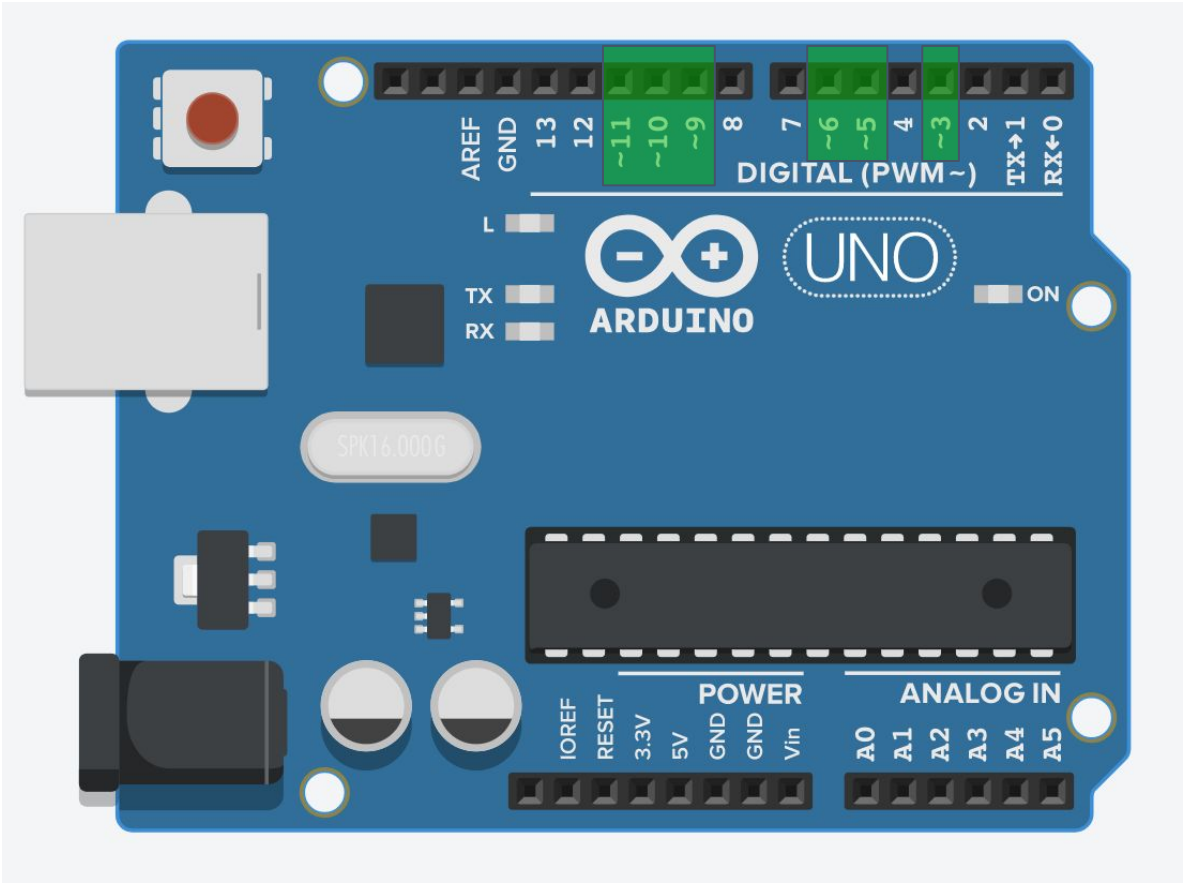


Analog Input:

ONLY A# pins can be used for analogRead.

These pins are connected to the analog-to-digital converter (ADC) inside the microcontroller.

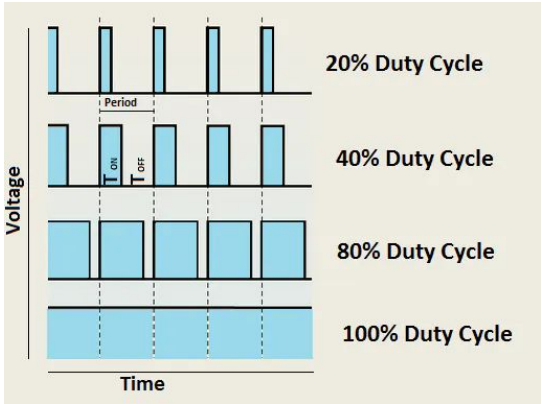
Digital and Analog Pins on Arduino Uno



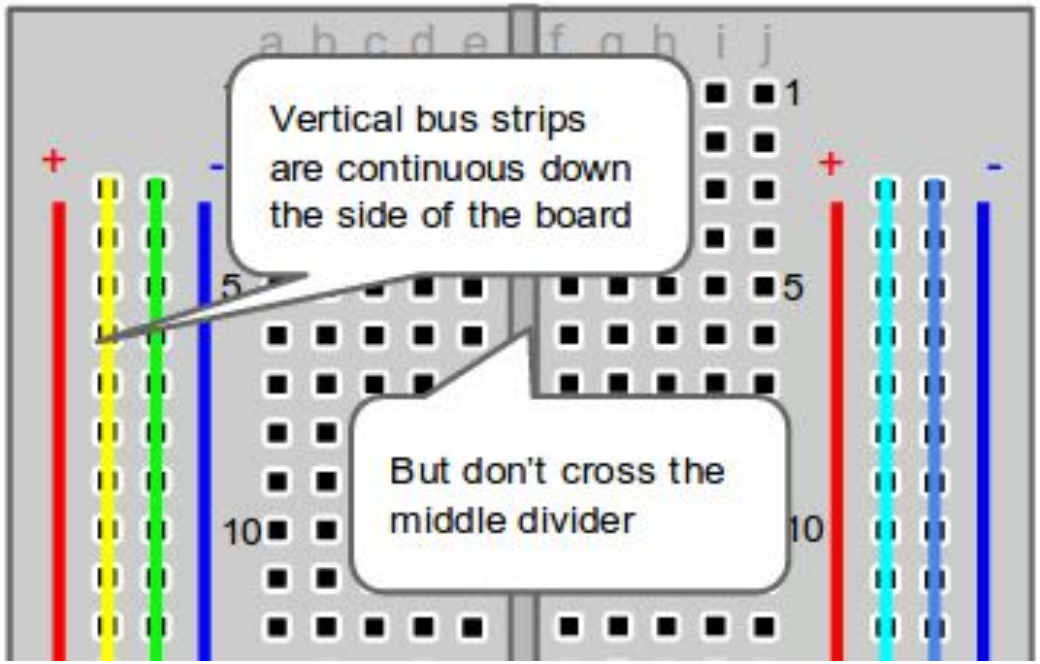
Analog Output:

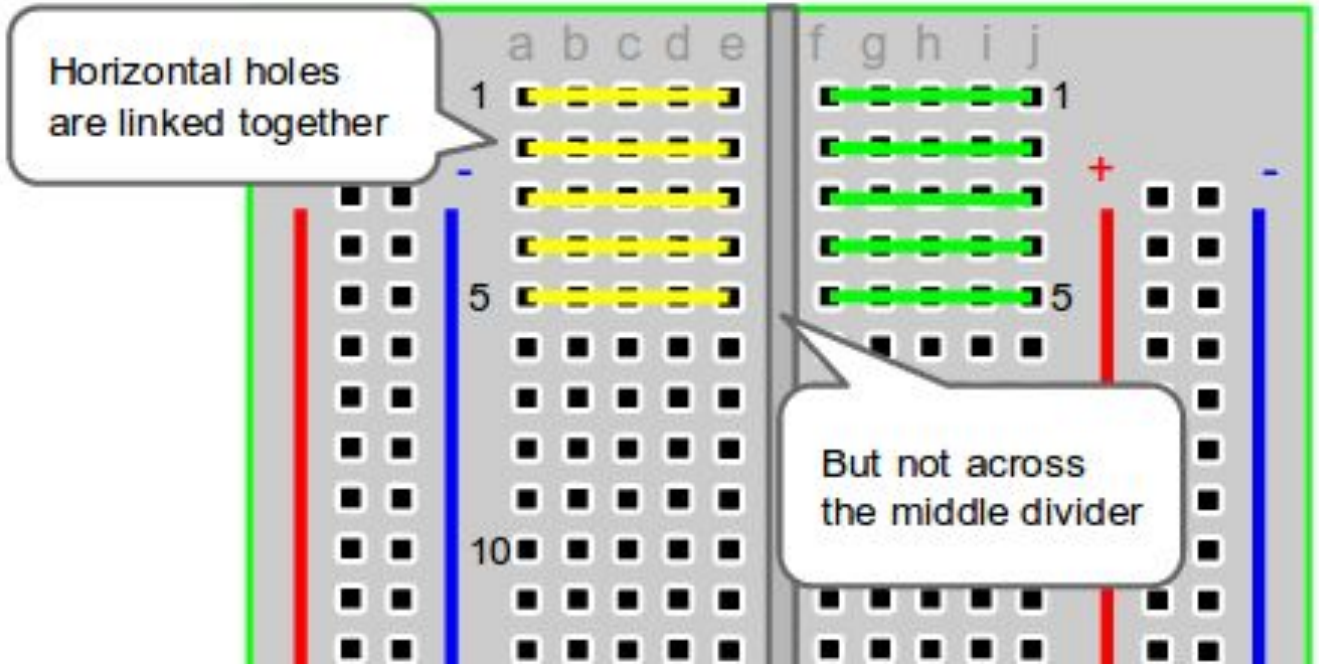
ONLY pins with tilde (~) can be used for analogWrite. (3, 5, 6, 9, 10, 11)

These pins are connected to the microcontroller's internal timer, which is used to generate pulse width modulation (PWM) signals.



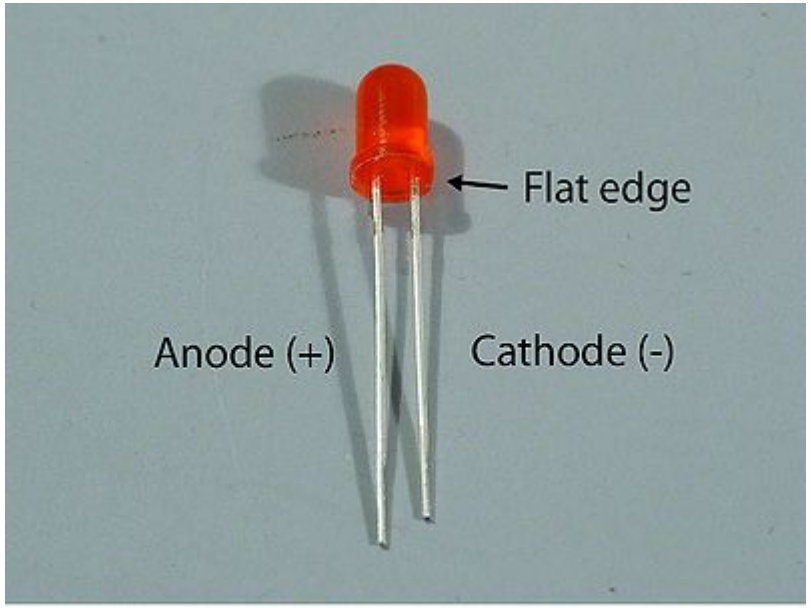
Breadboard Basics

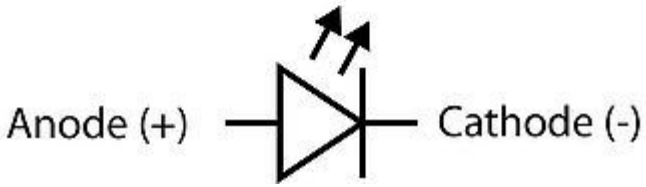




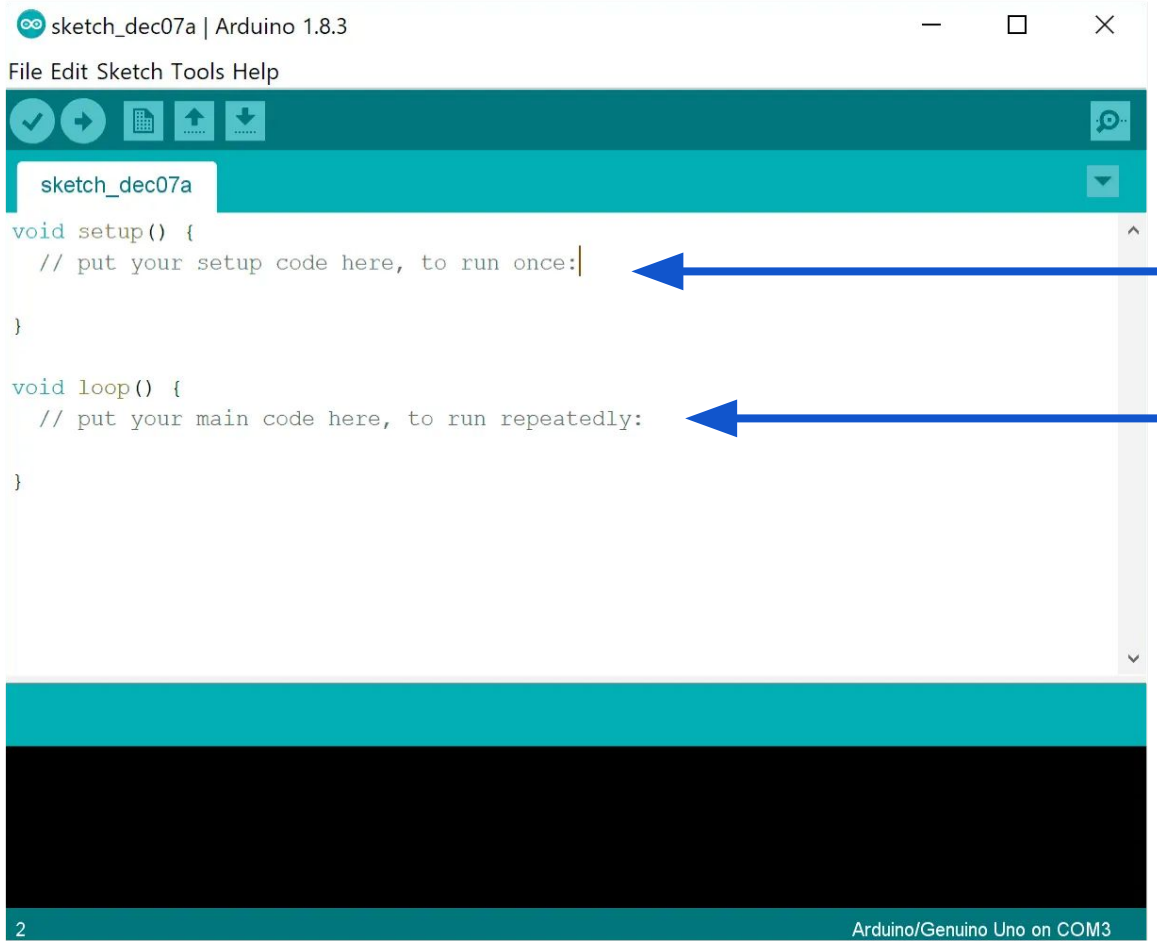
LED Polarity

Standard LEDs
(long leg is positive)





Arduino IDE



void setup() runs once at start

void loop() runs forever after setup

Functions to know

- **pinMode**(pin number, INPUT/OUTPUT);
  ex: pinMode(13, OUTPUT);

- **digitalWrite**(pin number, HIGH/LOW);
  ex: digitalWrite(13, HIGH);

- **delay**(time_in_milliseconds);
  ex: delay(2500); // delay of 2.5 seconds

- **// NOTE: -> commands ARE case-sensitive**

More functions to know

- **digitalRead**(pin number); // returns HIGH or LOW
  ex: x = digitalRead(2);

- **analogRead**(pin number); // returns 0 to 1023 (0 to 5V)
  ex: y = analogRead(0);

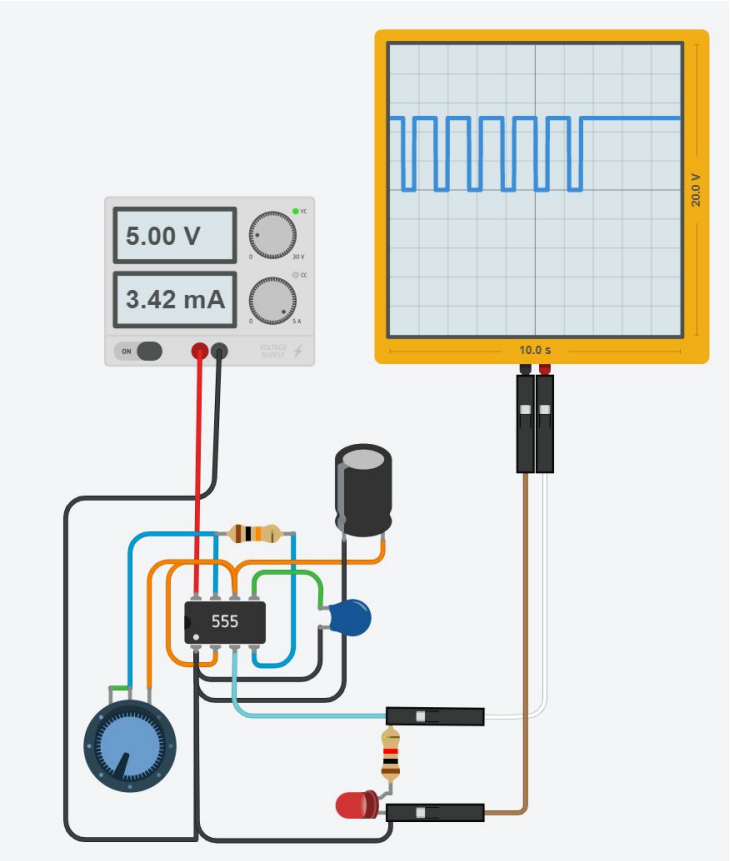
- **analogWrite**(0 to 255);
  ex: analogWrite(128); // equivalent to ~2.5V

- **// NOTE: -> commands ARE case-sensitive**

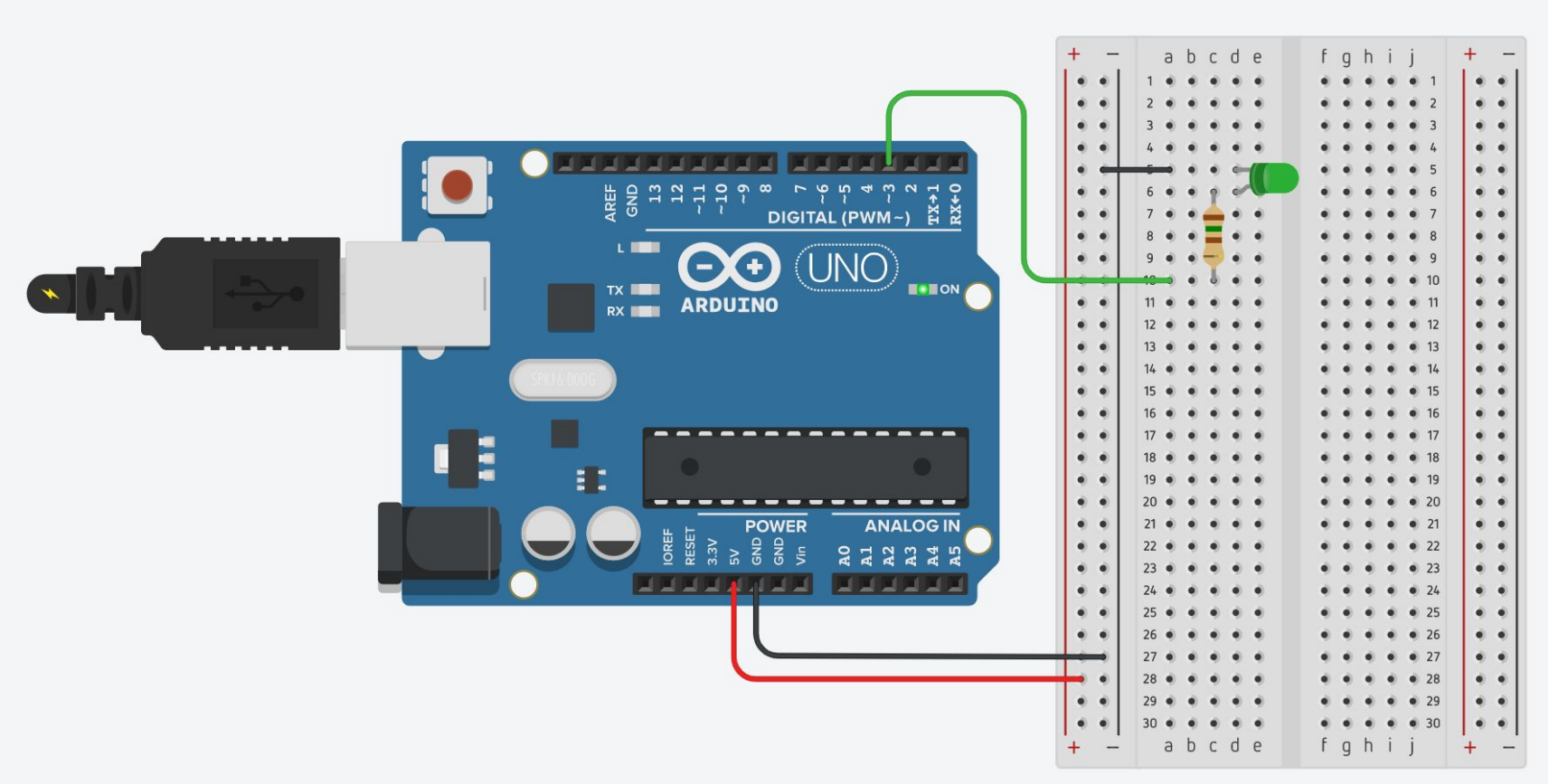
Autodesk Tinkercad

AUTODESK®
TINKERCAD®

tinkercad.com



LED Blink Project



- Green LED
- 150Ω resistor

Ohm's Law

$V = IR$

$I_{LED} = (V_s - V_{f\,(LED)}) / R_{series}$

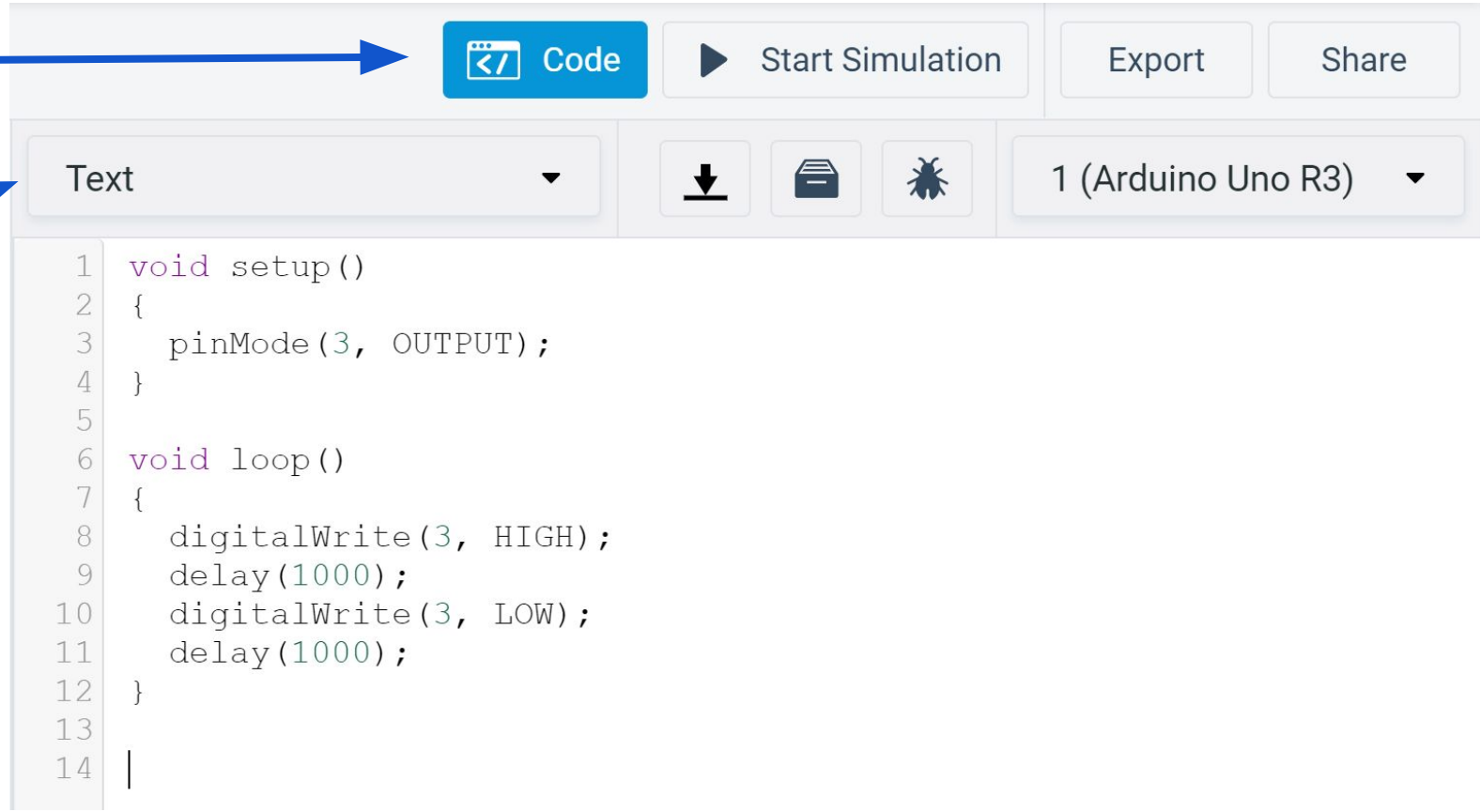
$I_{LED} = (5v - 3.3v) / 150\Omega$

$I_{LED} = 1.7/150 = 11.3mA$

LED Blink Code

Code Window

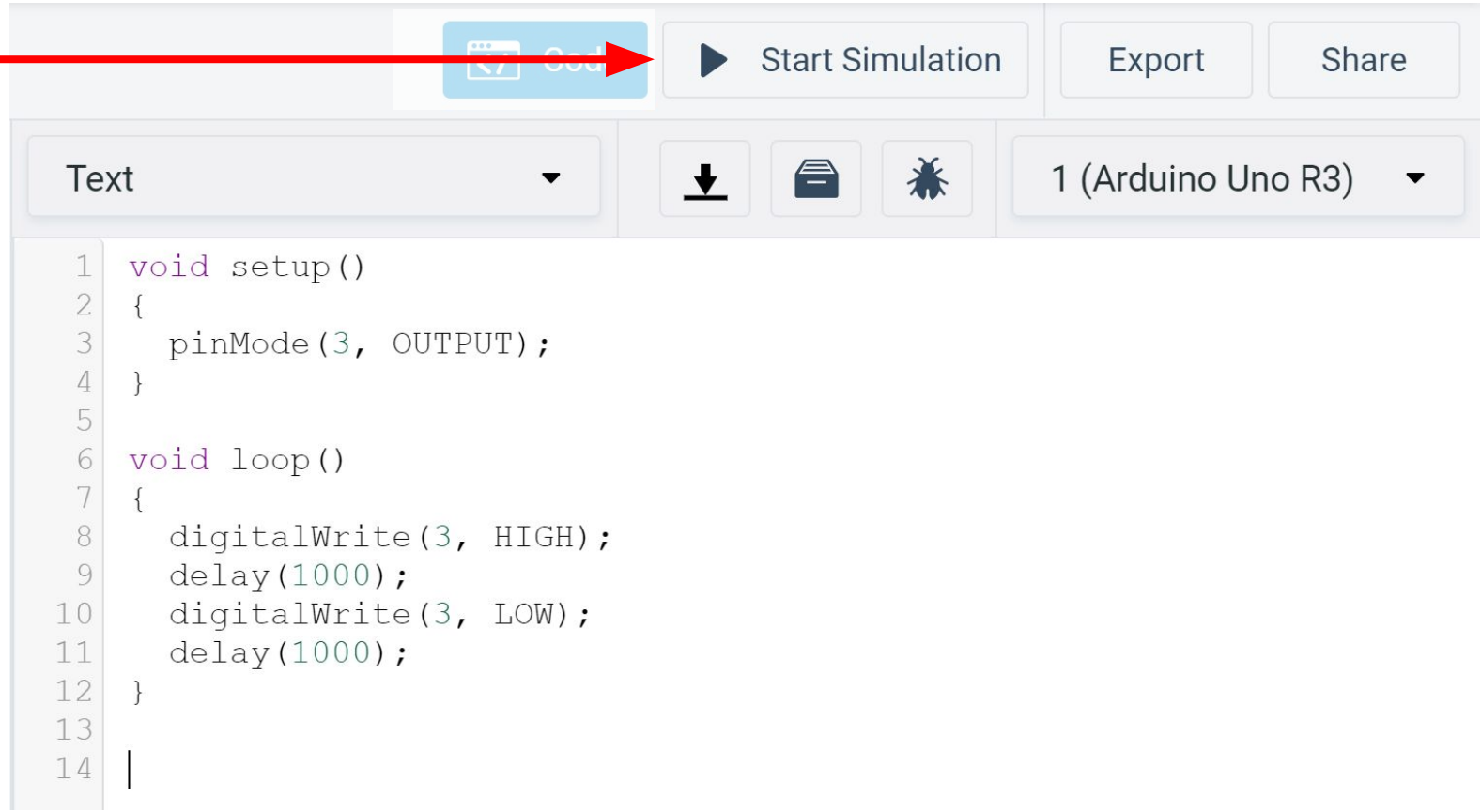
Text Mode



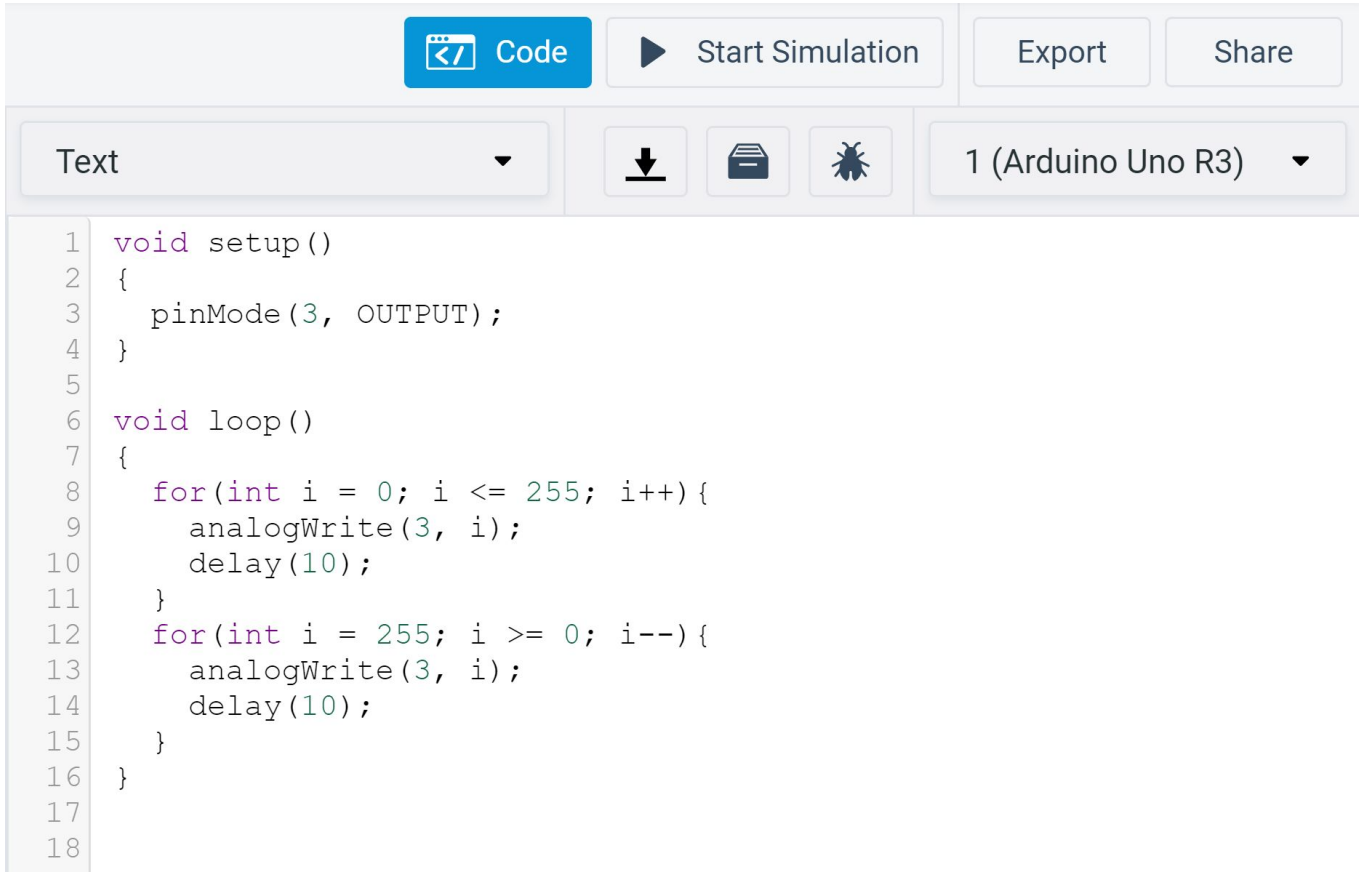

```
1  void setup()
2  {
3    pinMode(3, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(3, HIGH);
9    delay(1000);
10   digitalWrite(3, LOW);
11   delay(1000);
12 }
13
14
```

Running Simulation

Run Code



New Code: LED Fading




```
1  void setup()
2  {
3    pinMode(3, OUTPUT);
4  }
5
6  void loop()
7  {
8    for(int i = 0; i <= 255; i++){
9      analogWrite(3, i);
10     delay(10);
11   }
12   for(int i = 255; i >= 0; i--){
13     analogWrite(3, i);
14     delay(10);
15   }
16 }
17
18
```

For loop syntax:

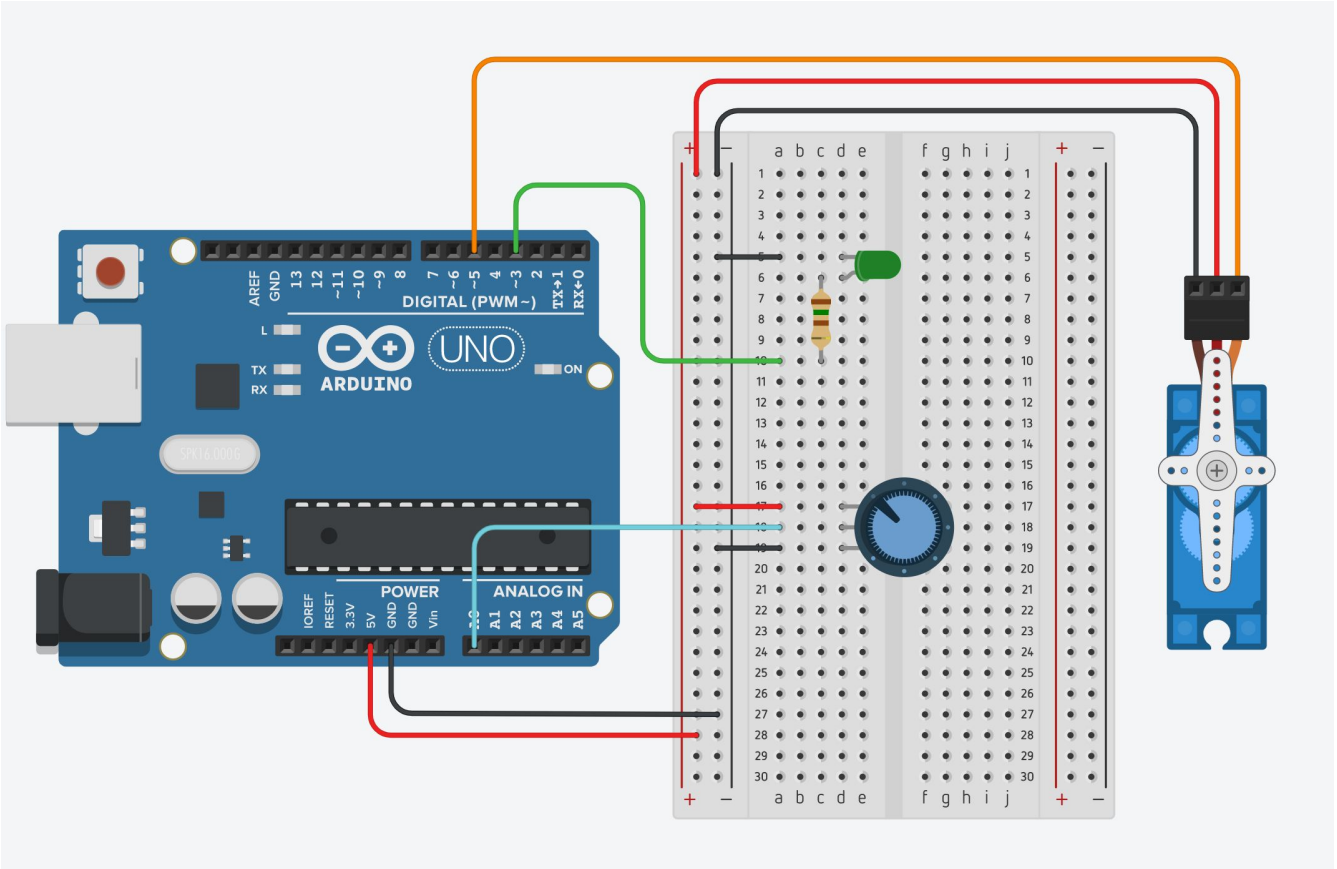
```
for( init variable; condition;
increment statement){

code to run each iteration

}
```

This code ramps up the LED voltage from 0 to 5V (with PWM) and then ramps it back down, creating a fade effect.

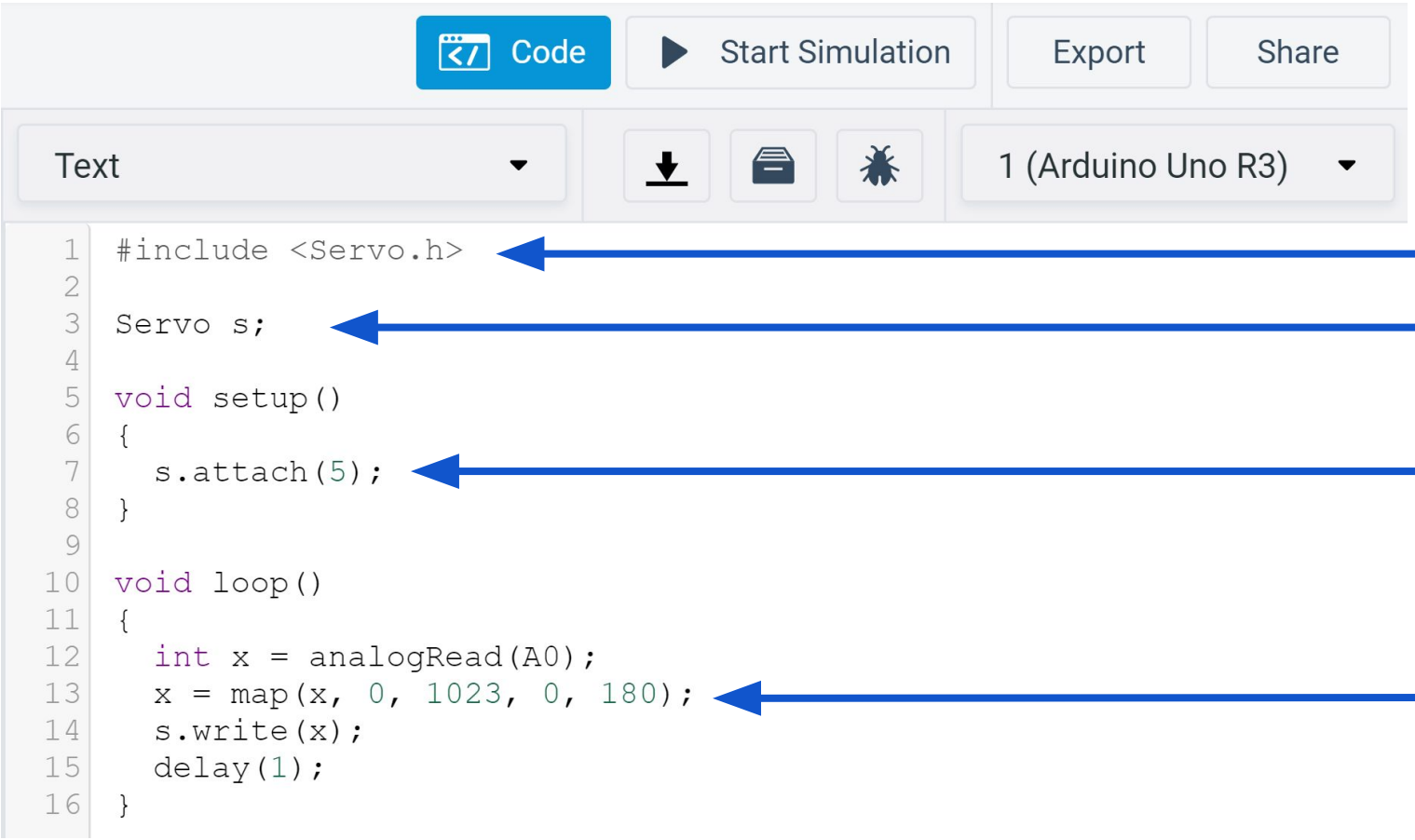
Servo Project



- Servo on Pin 5 (PWM)
- Potentiometer (Variable resistance/dial) on pin A0

We will read in a value from the potentiometer to control the servo's angle.

Servo Project Code




```
1  #include <Servo.h>
2
3  Servo s;
4
5  void setup()
6  {
7    s.attach(5);
8  }
9
10 void loop()
11 {
12   int x = analogRead(A0);
13   x = map(x, 0, 1023, 0, 180);
14   s.write(x);
15   delay(1);
16 }
```

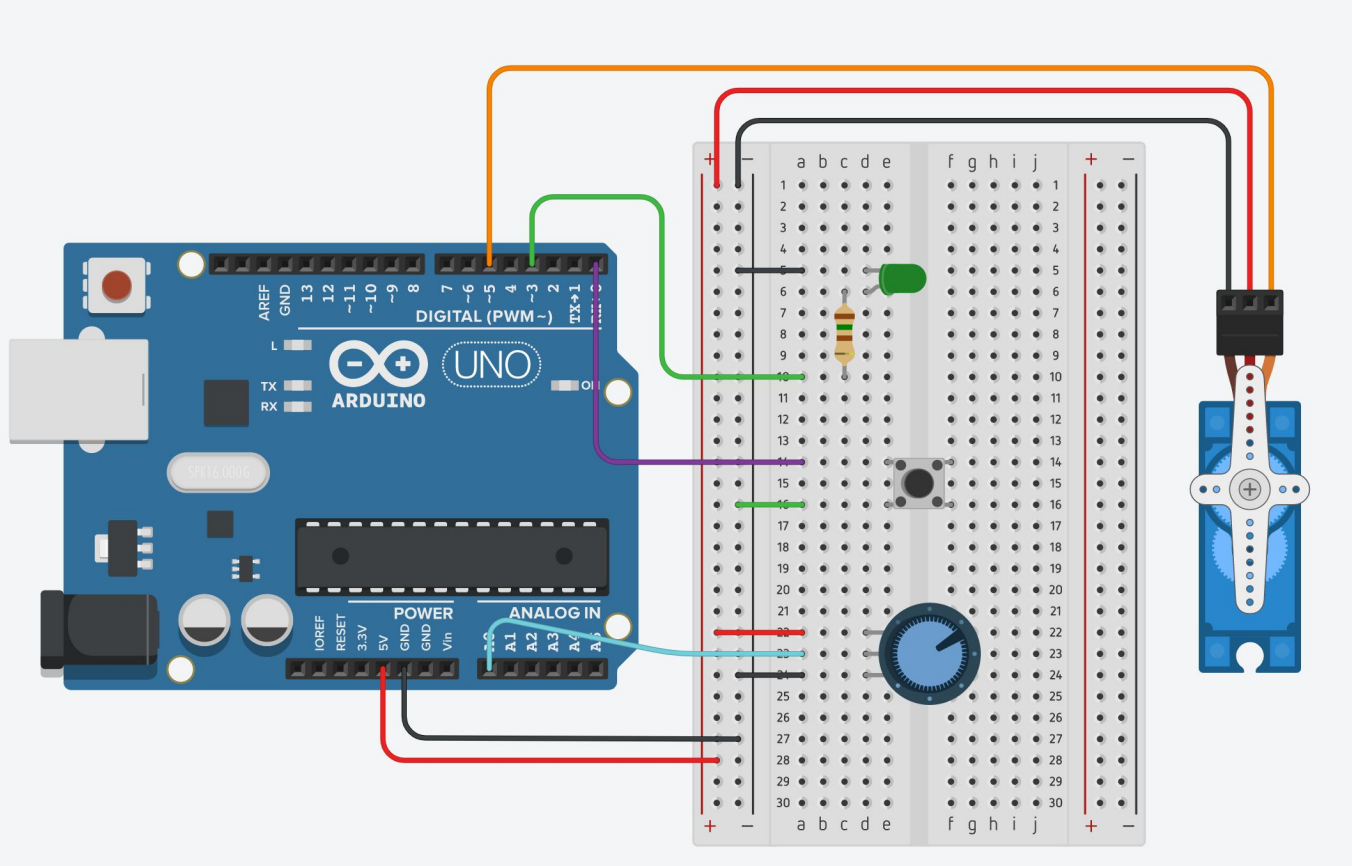
Include Servo library

Initialize servo

Connect servo to pin 5

Read in voltage from potentiometer
Map analogRead value to servo angle
Write angle to servo
Wait a bit to let servo reach new angle

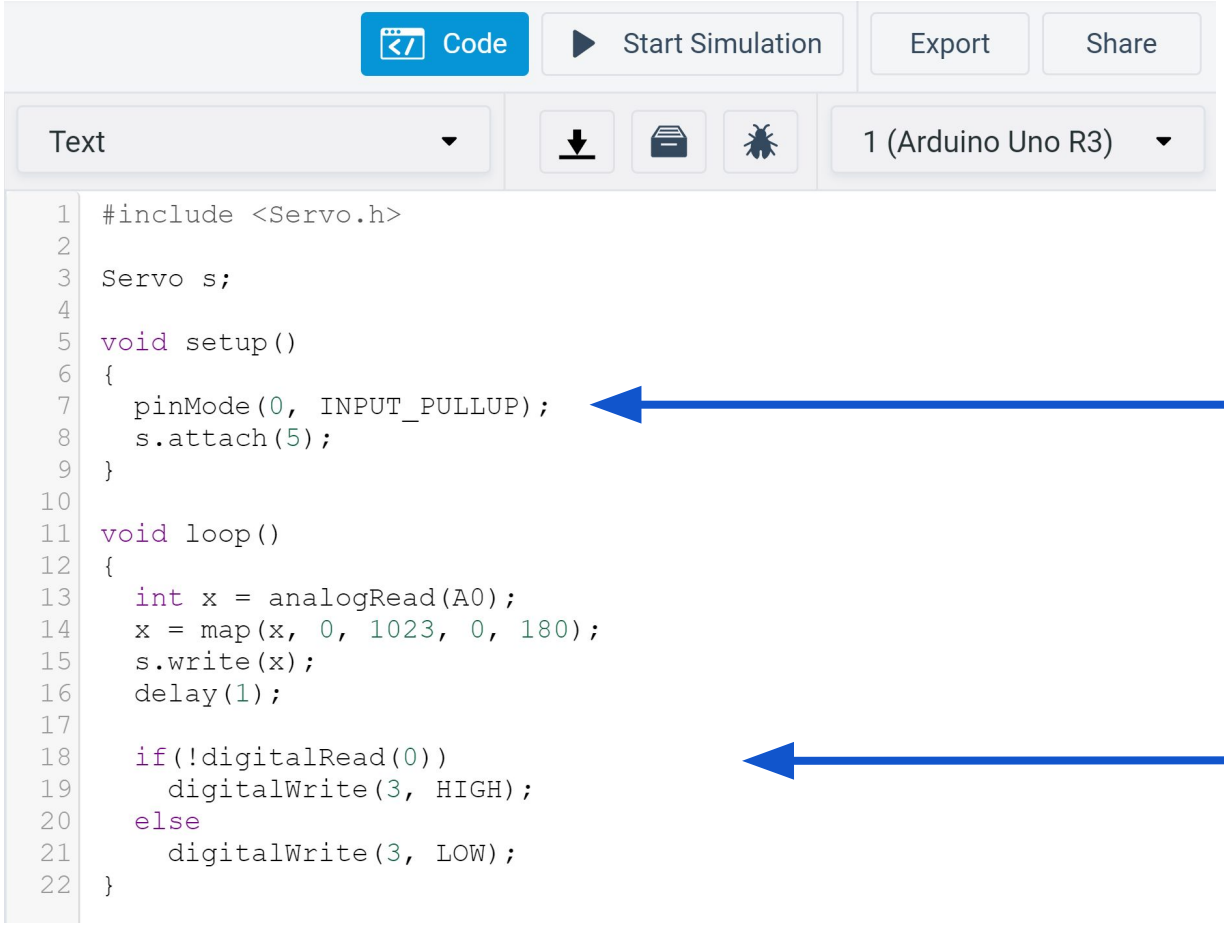
Servo, LED and Button Project



- Button added to pin 0

We can control things now by digitalRead'ing the button

Servo, LED, Button Project Code




```
#include <Servo.h>

Servo s;

void setup()
{
  pinMode(0, INPUT_PULLUP);
  s.attach(5);
}

void loop()
{
  int x = analogRead(A0);
  x = map(x, 0, 1023, 0, 180);
  s.write(x);
  delay(1);

  if(!digitalRead(0))
    digitalWrite(3, HIGH);
  else
    digitalWrite(3, LOW);
}
```

Set up button input

Check if button is pressed (active low)
If button is pressed, turn LED on
Otherwise, keep LED off