

Introduction to Arduino

Coming up:

- Context –examples of artist's/designers use of sensing technologies
- Introduction to the Arduino Board
- Output – Acting in the world
- Input – Getting data from the World
- Movement – Making something move!



Laetitia Sonami
The Lady's Glove
1994 - 2001

https://www.youtube.com/watch?v=C8GqbS2w_Lg



Rachel Freire - The Embodisuit



Philips Design Probe
– Bubbelle Dress
2007 -2020



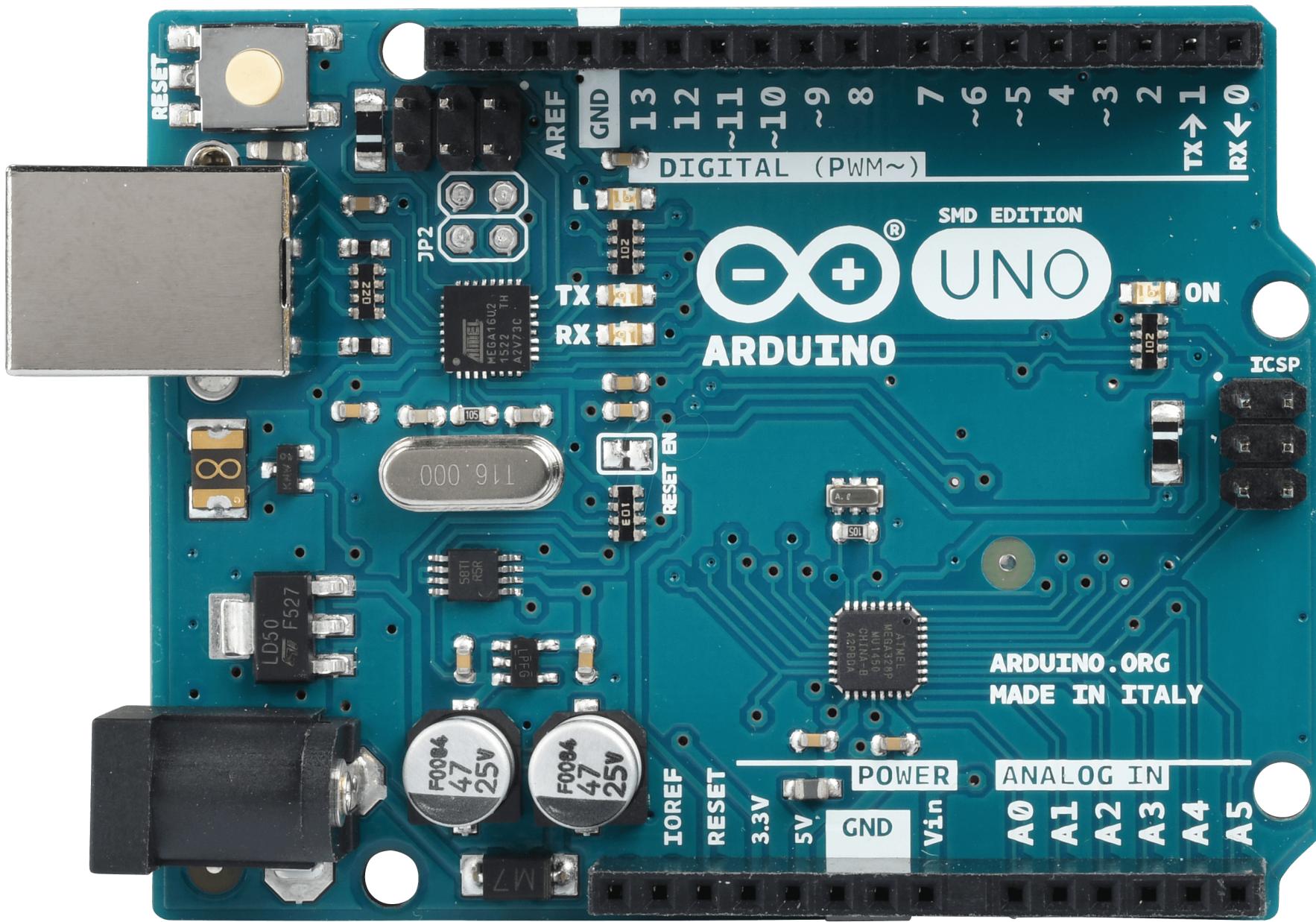
Takehito Etani
Masticator
2005



Swarm Street
Acconci Studio
2013

<https://github.com/JohnMechatronics>

Introduction to the Arduino Board



Get going

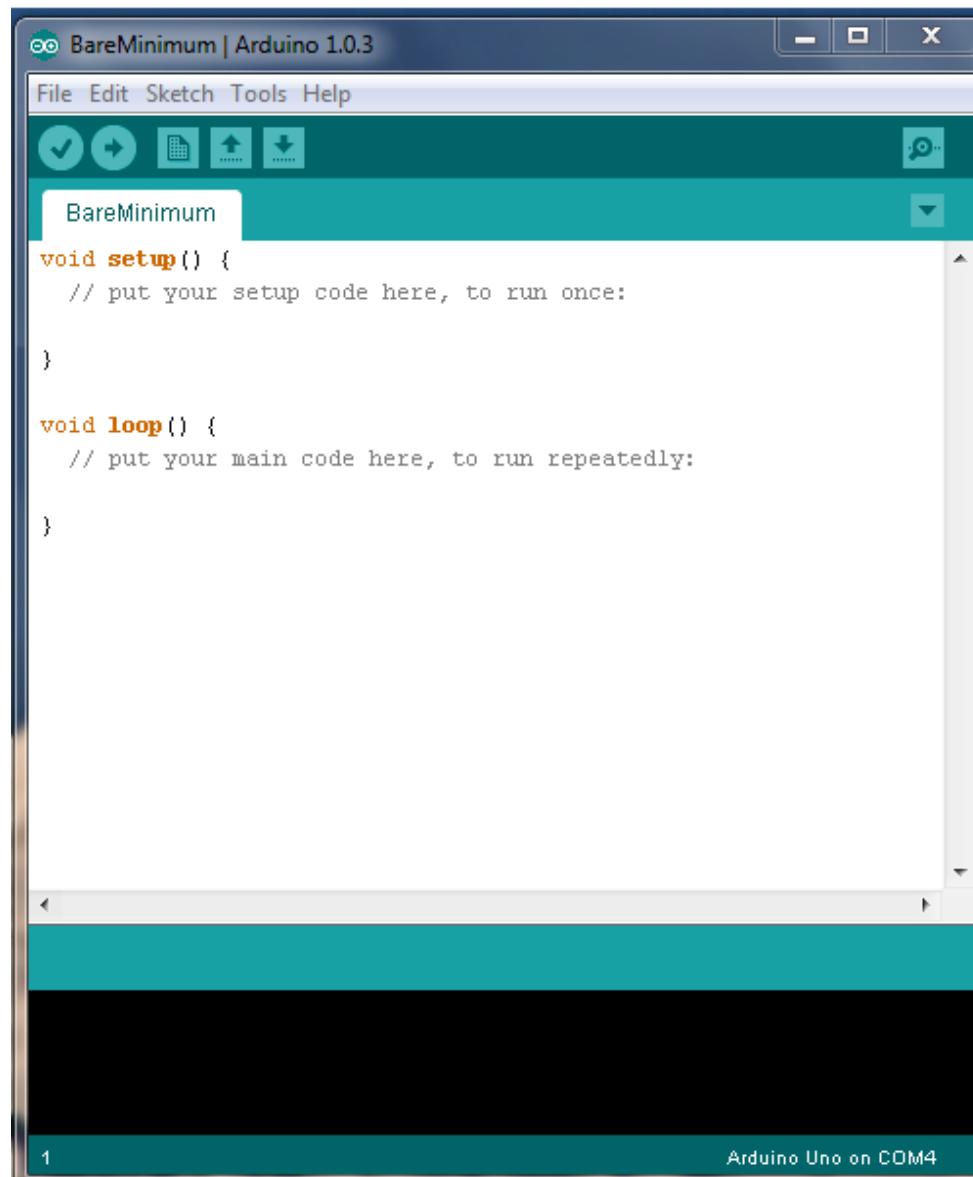
- In the IDE goto:

Tools > Board > Arduino/Genuino Uno

Tools > Port > COM[!] Arduino/Genuino Uno

Arduino IDE

- integrated development environment



Hello World



File > Examples > 0.1 Basics > Blink

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** The title bar displays the name "Blink §".
- Toolbar:** The top toolbar contains standard icons for file operations (New, Open, Save, Print, Undo, Redo) and a magnifying glass icon for search.
- Code Editor:** The main area contains the "Blink" sketch code:

```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                         // wait for a second
  digitalWrite(LED_BUILTIN, LOW);        // turn the LED off by making the voltage LOW
  delay(1000);                         // wait for a second
}
```
- Serial Monitor:** A large black rectangular area representing the serial monitor window.
- Status Bar:** The bottom status bar shows the text "1" on the left and "Arduino/Genuino Uno on COM1" on the right, accompanied by a small blue progress bar.

`digitalWrite()`

If the pin has been configured as an OUTPUT with `pinMode()`, its voltage will be set to:

- 5V for HIGH
- 0V (ground) for LOW.

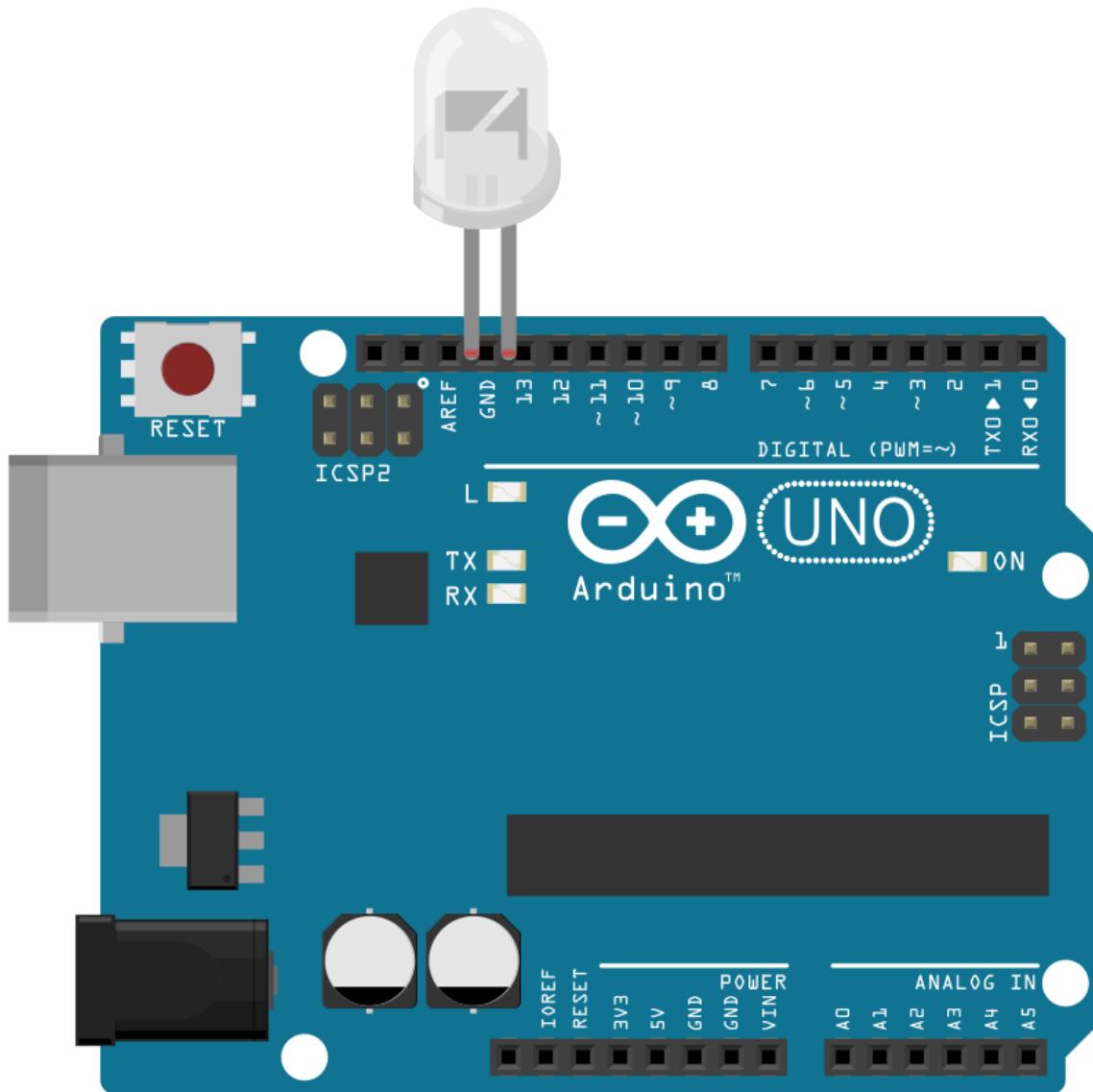


(anode)

+

(cathode)

GND

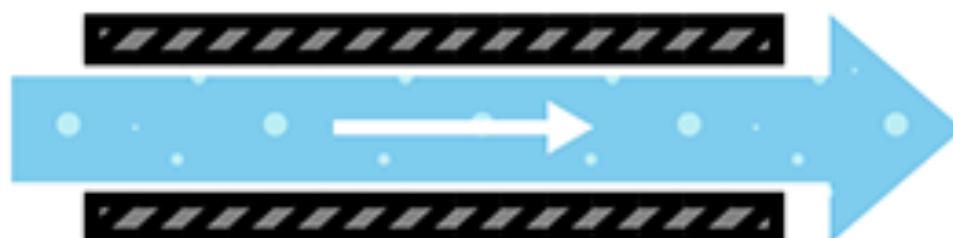


Basic Electronics

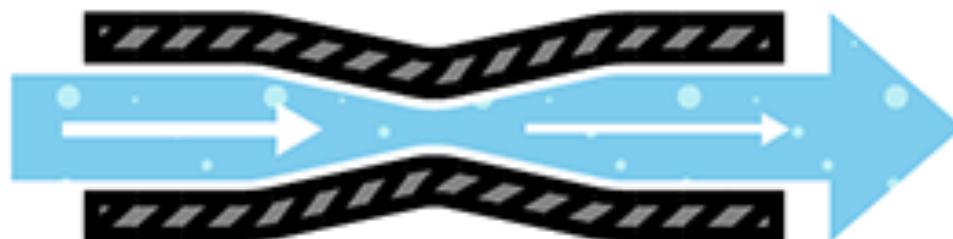
- **Voltage** – The difference in electrical energy between two points. It is measured in volts (**v**).
- **Current** – The quantity or amount of electrical energy passing a particular point. It is measured in amps (**A**)
- **Resistance** - The measure of a materials ability to prevent the flow of electricity. Resistance is measured in ohms (**Ω**)

Resistance

Less resistance



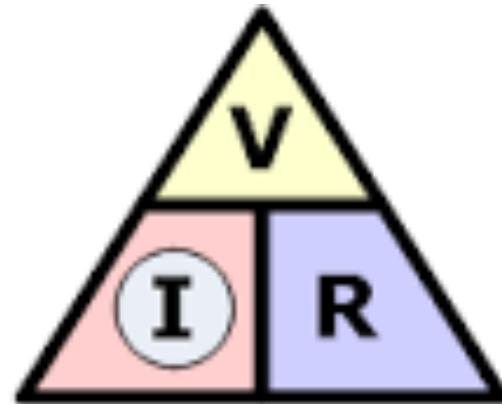
More resistance



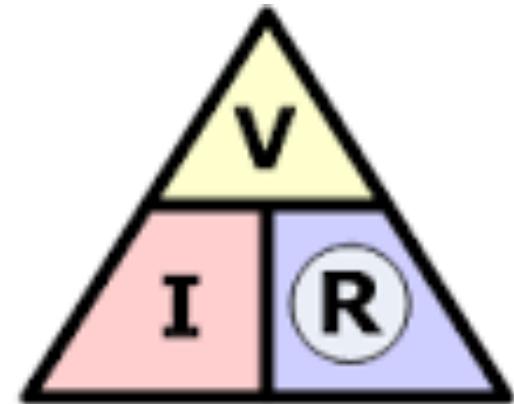
Ohm's law



$$\text{V} = I \times R$$



$$I = \frac{V}{R}$$



$$R = \frac{V}{I}$$

Ohm's law states that **Voltage (V)** is equal to **current (A)** times **resistance ()**

Work Out the Resistance

$$R = (V_s - V_f) / I$$

V_s = Source Voltage

V_f = LED Forward Voltage

I = LED required voltage



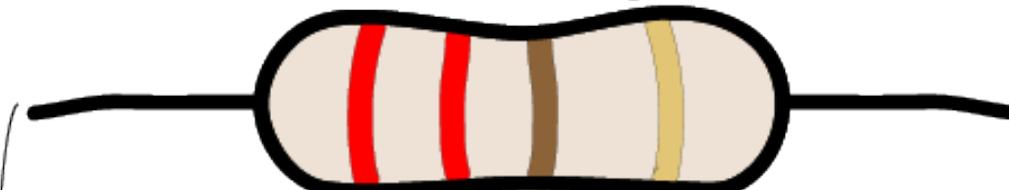
220 Ohm Resistor

220 Ohm Resistor

2 2 $\times 10$

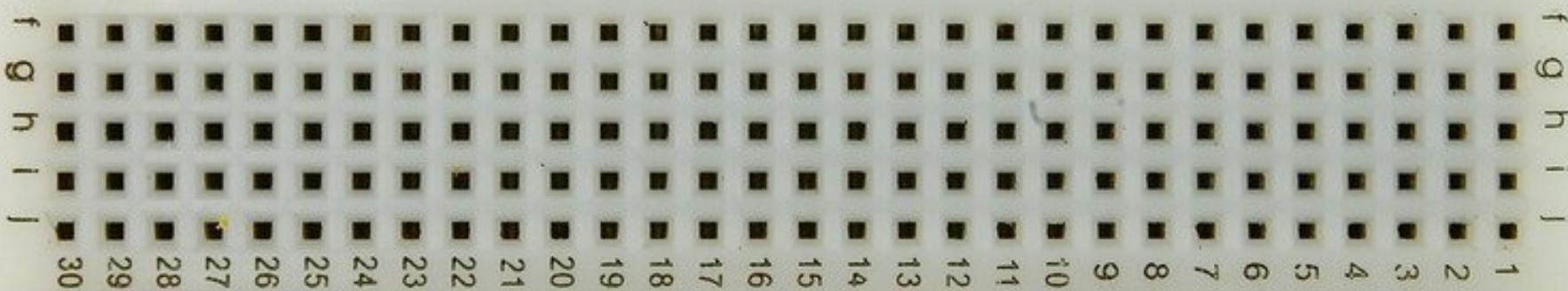
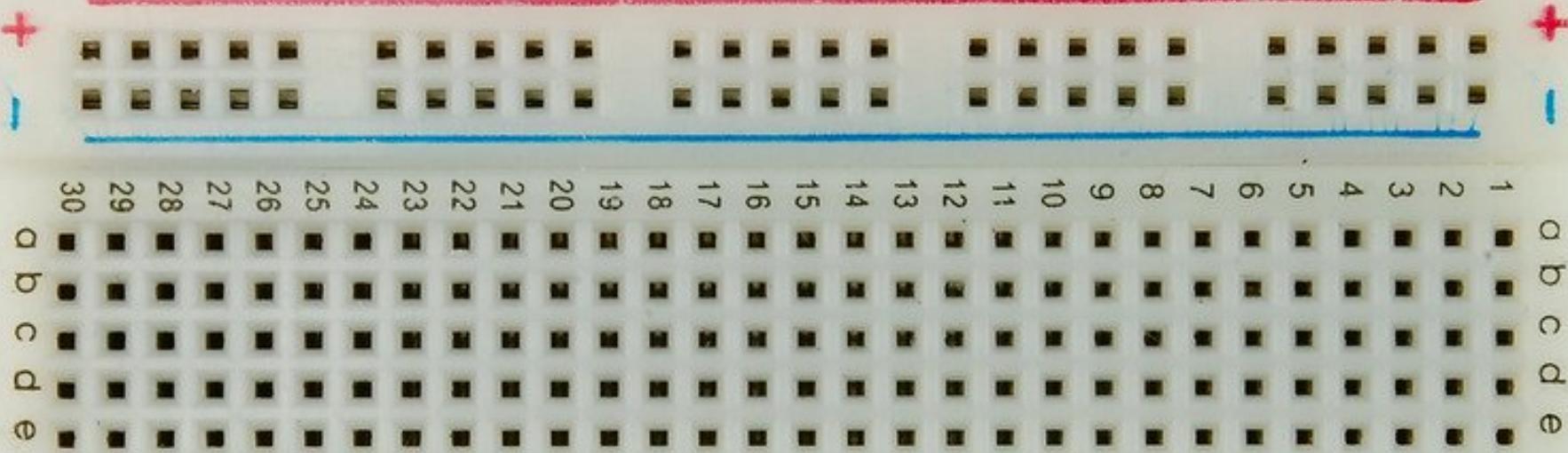
$\pm 5\%$

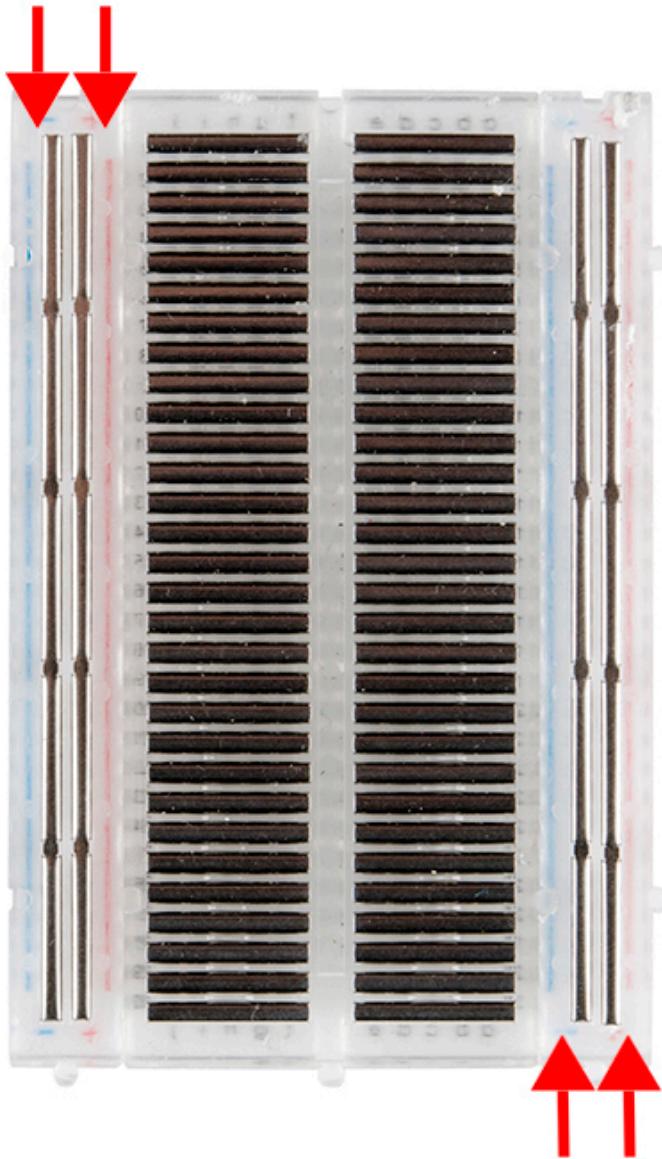
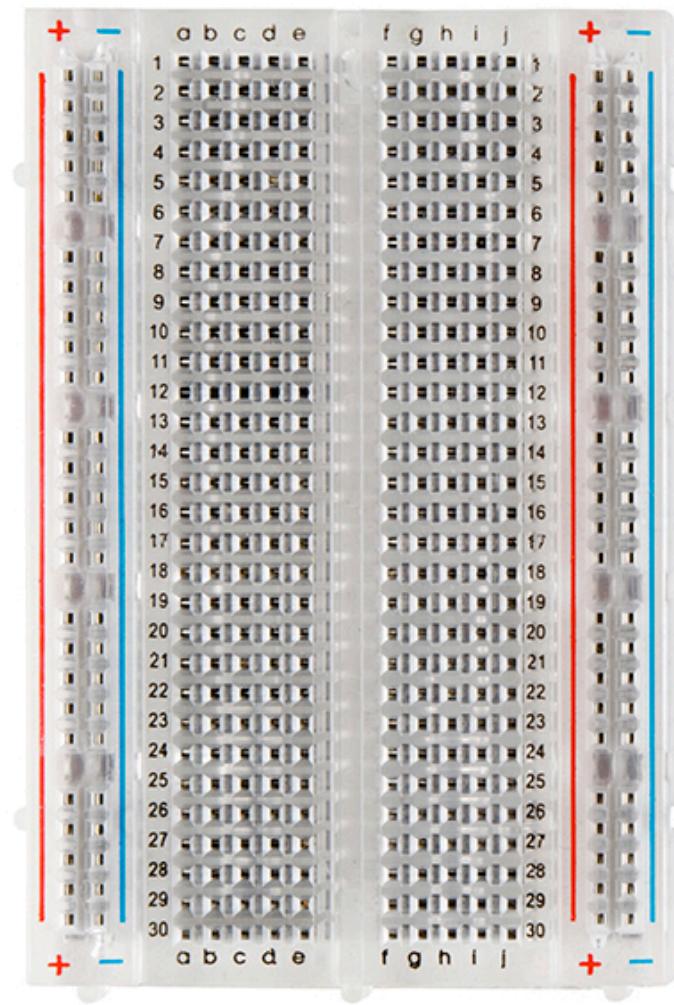
sample resistor
with color bands

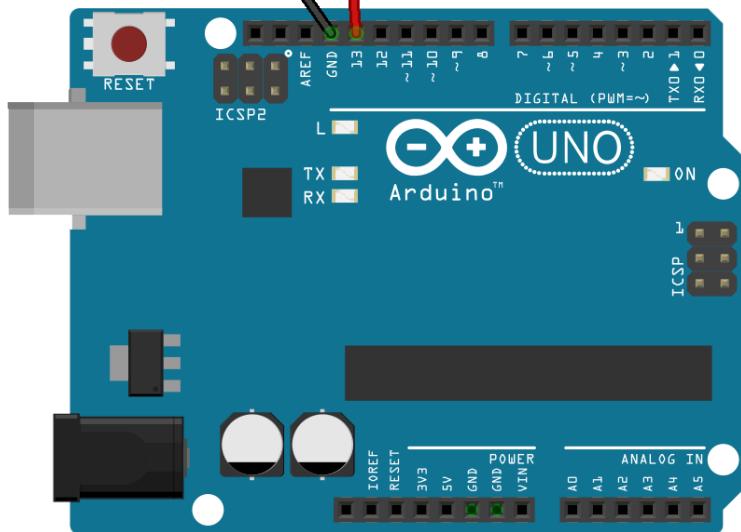
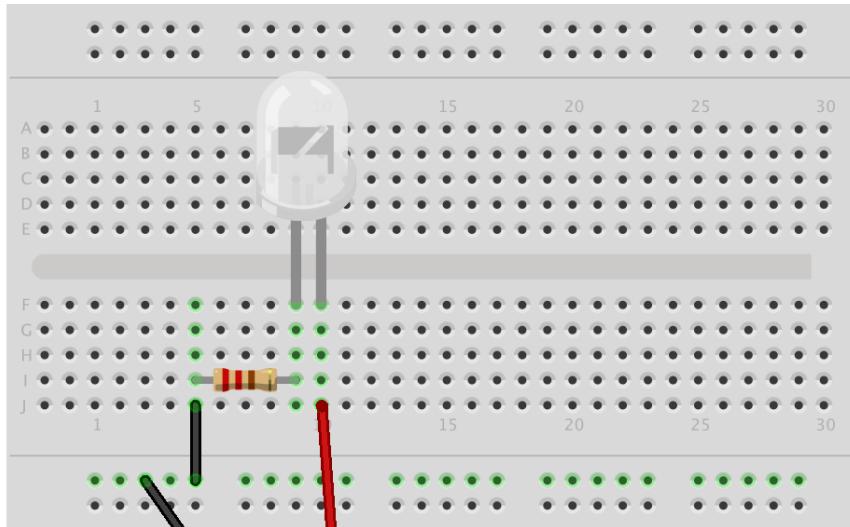


what the color
bands represent on
the resistor

	1st Digit	2nd Digit	Multiplier	Tolerance
Black	0	0	1	5% Gold
Brown	1	1	10	10% Silver
Red	2	2	100	
Orange	3	3	1,000	
Yellow	4	4	10,000	
Green	5	5	100,000	
Blue	6	6	1,000,000	
Purple	7	7		
Gray	8	8		
White	9	9		







```
int ledPin = 13; // LED connected to digital pin 13

// the setup function runs once when you press reset or power the board
void setup(){
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(ledPin, OUTPUT);
}

// the loop function runs over and over again forever
void loop(){
    digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(ledPin, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

Variables

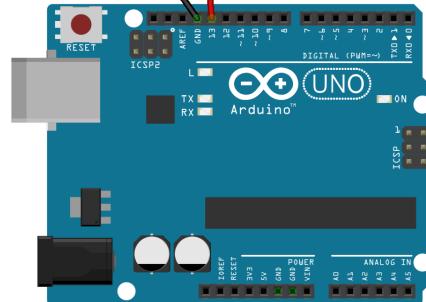
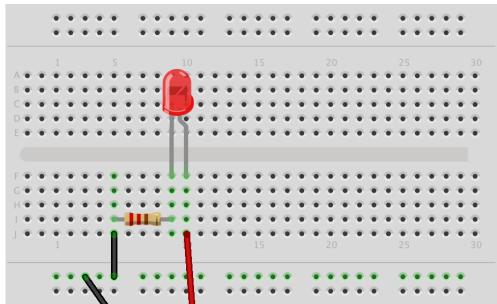
- A variable is a way of naming and storing a value for use by the program
- E.g. data from a sensor or value that will be transformed by a mathematical calculation.

```
int ledPin = 13;
```

```
digitalWrite(ledPin, HIGH); -> digitalWrite(13, HIGH);
```

Task

- Test the circuit
- Can you alter the blink time?
- Try different rates



fritzing

Fade

- Rewire the LED to pin 9
- From the Arduino IED open:
- File > Example > 01.Basics > Fade
- Upload the Code
- What does it do?

The image shows the Arduino IDE interface. At the top, there are standard file operations icons: a checkmark for save, a circular arrow for refresh, a document for new, an upward arrow for open, and a downward arrow for save. On the right side of the header bar, there is a small icon of a person's head with a gear, likely for user profile or settings.

The main workspace is titled "Fade §". The code itself is as follows:

```
/*
  Fade
*/

int led = 9;          // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

`analogWrite()` v `digitalWrite()`

`digitalWrite()`

- Two states High or Low
- High - > 5v
- Low - > 0v

`analogWrite()`

- Set the intensity between 0 and 255

if

```
if (brightness <= 0 || brightness >= 255) {  
    fadeAmount = -fadeAmount;  
}  
}
```

*Remember a – and a – make a +

if

- The `if()` statement allows you to make something happen or not, depending on whether a given condition is true or false. Eg:

```
if (someCondition) {  
    // do stuff if the condition is true  
}
```

if-else

```
if (someCondition) {  
    // do stuff if the condition is true  
} else {  
    // do stuff if the condition is false  
}
```

else-if

```
if (someCondition) {  
    // do stuff if the condition is true  
} else if (anotherCondition) {  
    // do stuff only if the first condition is false  
    // and the second condition is true  
}
```

Comparison Operators

`!=` (not equal to)

`<` (less than)

`<=` (less than or equal to)

`==` (equal to)

`>` (greater than)

`>=` (greater than or equal to)

```
if (brightness <= 0 ||
```

```
brightness >= 255)
```

Boolean Operators

&& - and

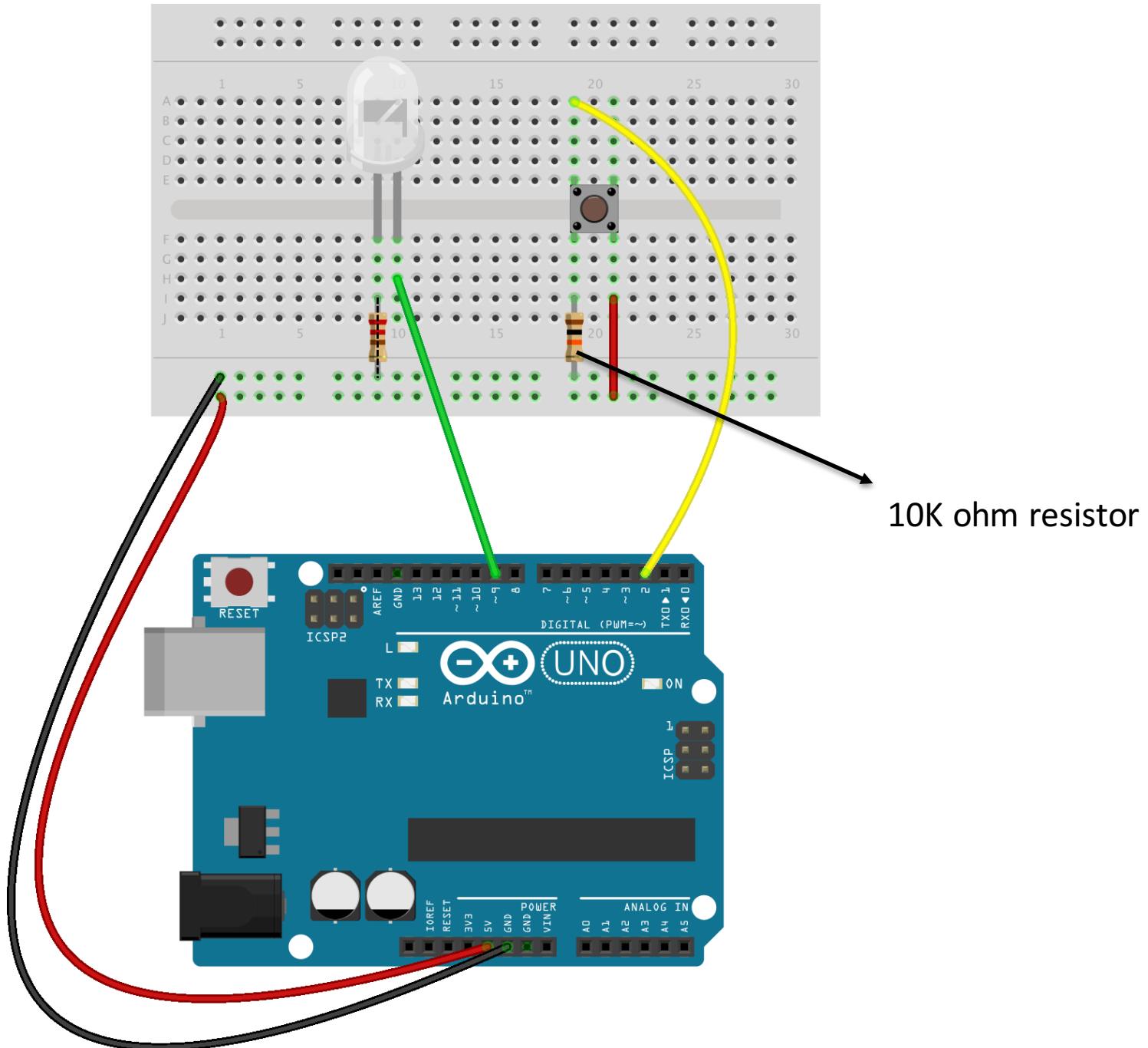
|| - or

```
if (brightness <= 0 || brightness >= 255)
```

Input - Sensing the World

Responding to a button press

- Light on off
- Code = LED-Button.ino



```
const int buttonPin = 2;
const int ledPin = 9;
int buttonState = LOW;

void setup() {
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT);
}

void loop() {
    buttonState = digitalRead(buttonPin);

    if (buttonState == HIGH) {
        digitalWrite(ledPin, HIGH);
    } else {
        digitalWrite(ledPin, LOW);
    }
}
```

LED-button.ino

digitalRead()

```
buttonState = digitalRead(buttonPin);
```

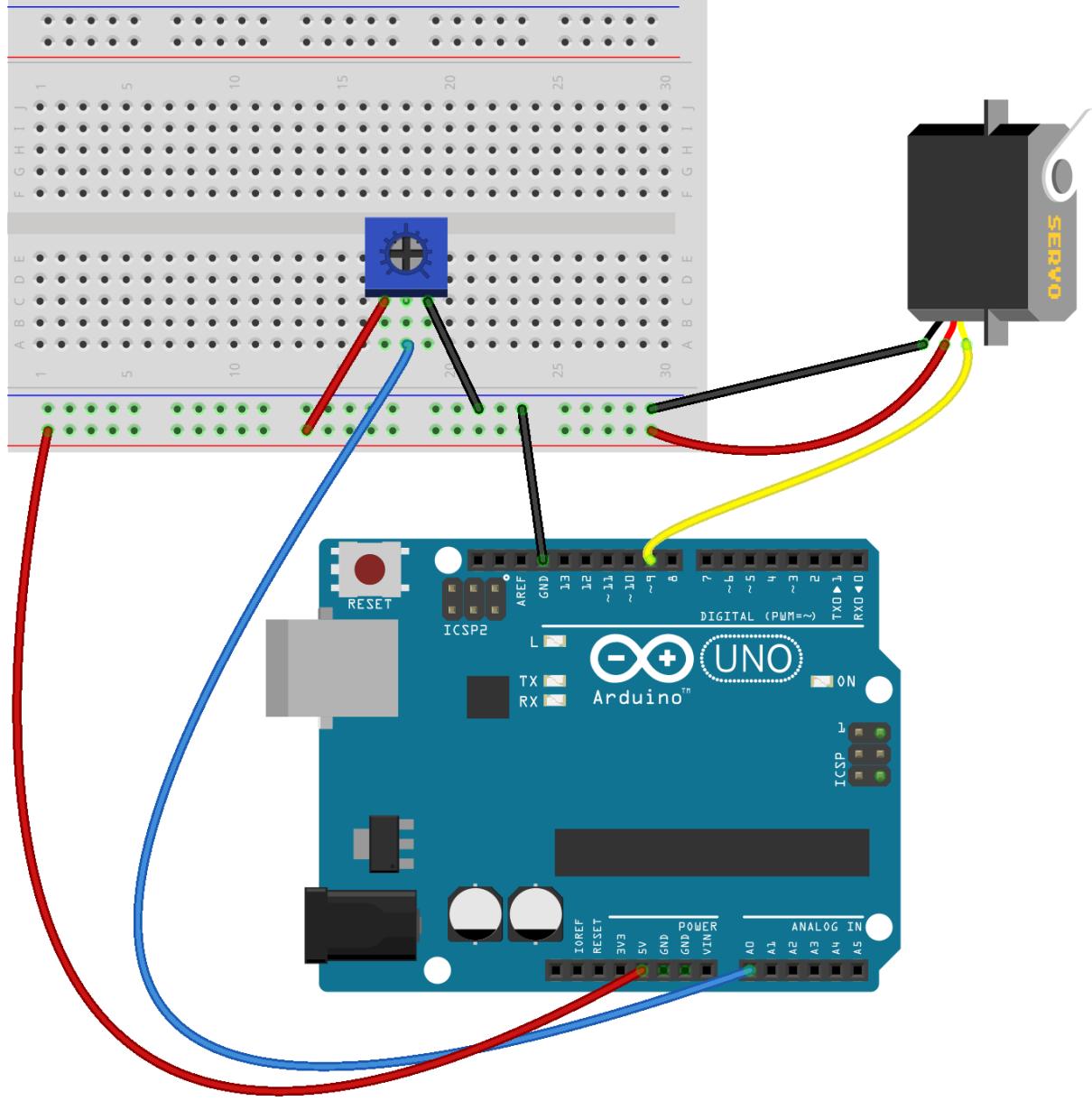
- Reads the value from a specified digital pin
- Value will be stored in variable - buttonState
- The value will be either HIGH (5v) or LOW (0v).

Morse Code

- Swap the LED for the Piezo Buzzer



Movement – Making something move!



```
#include <Servo.h>
```

```
Servo myservo;
```

```
int potpin = 0;
```

```
int val;
```

```
void setup() {
```

```
    myservo.attach(9);
```

```
}
```

```
void loop() {
```

```
    val = analogRead(potpin);
```

```
    val = map(val, 0, 1023, 0, 180);
```

```
    myservo.write(val);
```

```
    delay(15);
```

```
}
```

File > Example > Servo > knob

analogRead()

Val = **analogRead(potPin);**

- Reads the value from the specified analog pin.
- It maps input voltages between 0 and 5 volts into integer values between 0 and 1023.
- resolution = .0049 volts (4.9 mV) per unit

`analogRead()` v `digitalRead()`

`digitalRead()`

- Two states High or Low
- High - > 5v
- Low - > 0v

`analogRead()`

- Reads a value between 0 and 1023.

Map()

```
val = map(val, 0, 1023, 0, 180);
```

- Re-maps a number from one range to another.
- Used here to map the pot range 0 – 123 to the servo range 0 - 180

Task

- add an LED into the circuit
- Program it to fade in response to the potentiometer.

Conclusion

- We have learnt the two key methods of controlling devices connected to the Arduino:
 - digitalWrite()
 - analogueWrite()
- The two key methods of getting input into the Arduino:
 - digitalRead()
 - analogRead()
- And one of the main control methods:
 - if if-else else-if