

# Introduction to Arduino

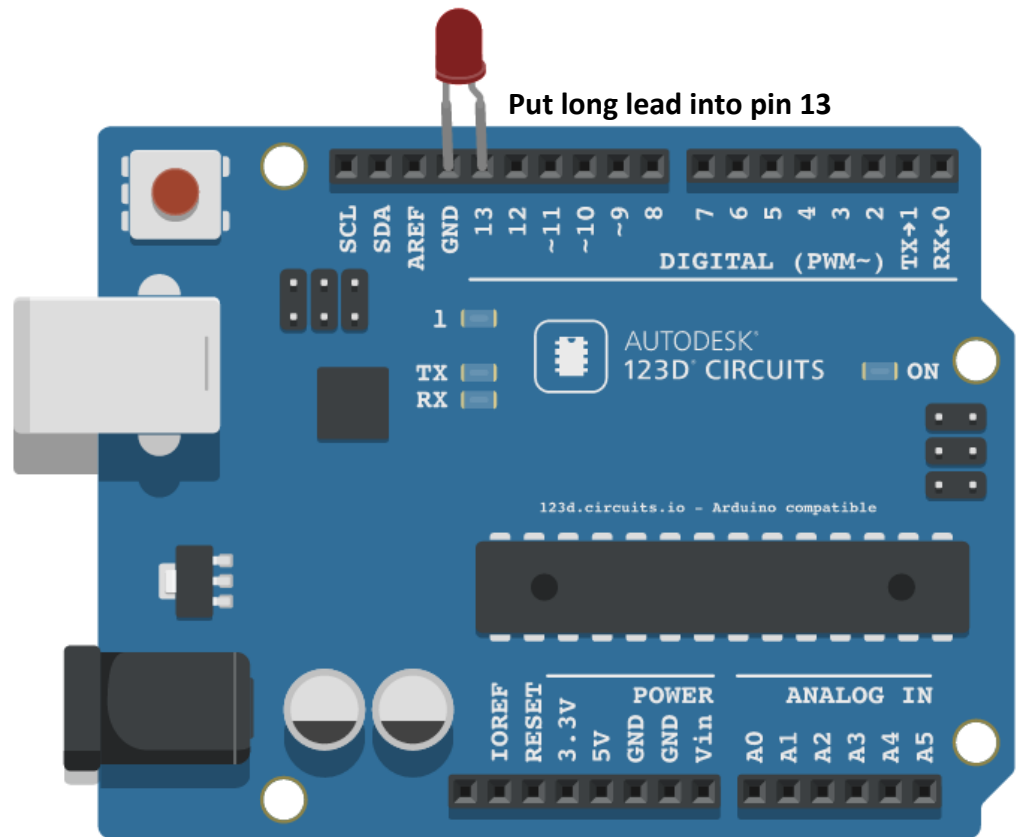
## Diagrams & Code

Brown County Library

### Project 01: Blinking LED

Components needed:

- Arduino Uno board
- LED



### // Project 01: Blinking LED

// Source: Getting Started with Arduino, 3rd edition by Massimo Banzi and Michael Shiloh

```
int LED = 13; // LED connected to digital PIN 13

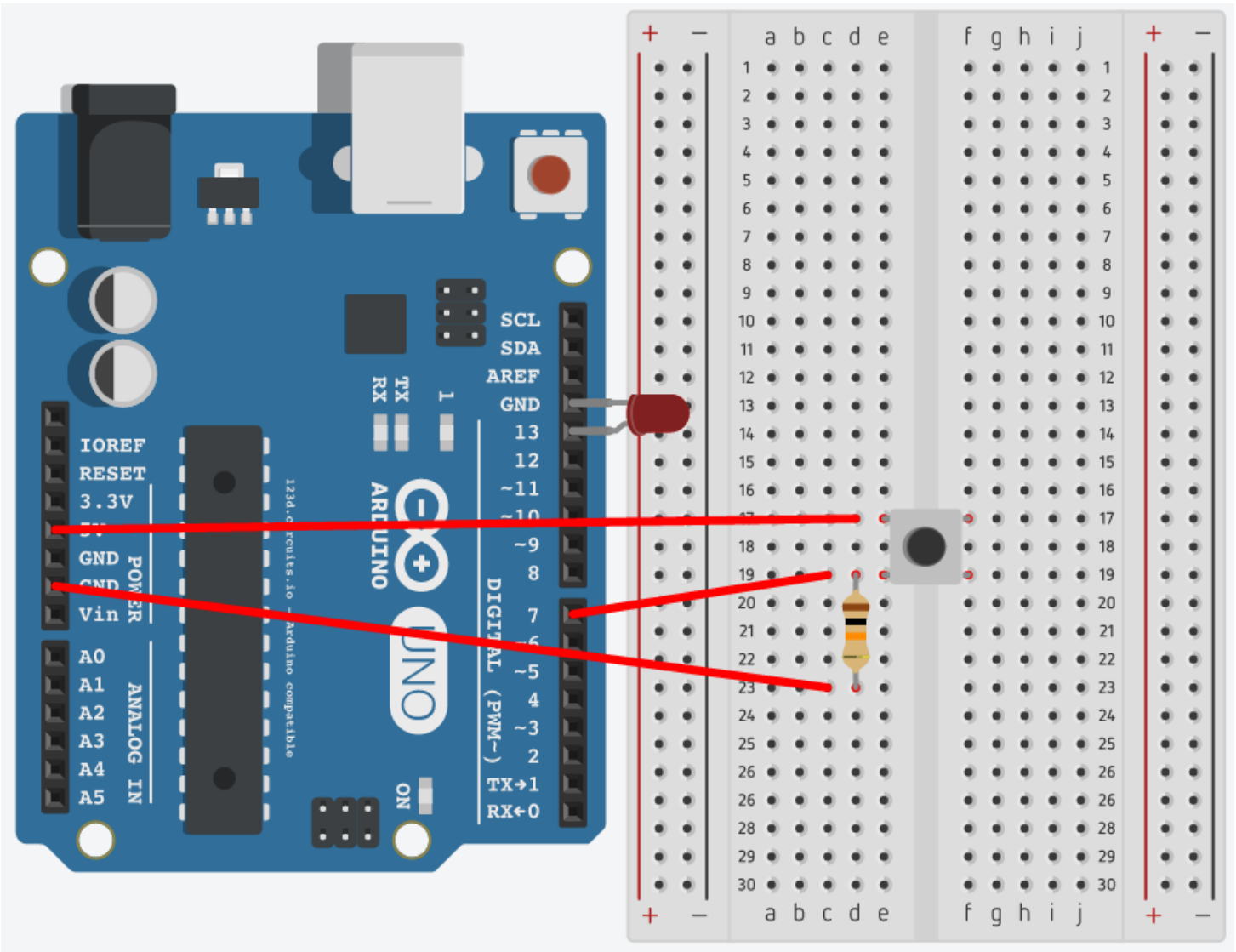
void setup() {
  pinMode(LED, OUTPUT); // sets the digital PIN as output
}

void loop() {
  digitalWrite(LED, HIGH); // turns the LED on
  delay(1000);           // waits for a second (1000 milliseconds)
  digitalWrite(LED, LOW); // turns the LED off
  delay(1000);           // waits for a second (1000 milliseconds)
}
```

## Projects 02, 03 and 03B

Components needed:

- Arduino Uno board
- LED
- 3 jumper wires
- 10k ohm resistor
- breadboard
- push button



## // Project 02 : Turn on LED while the button is pressed

// Source: Getting Started with Arduino, 3rd edition by Massimo Banzi and Michael Shiloh

```
int LED = 13; // the pin for the LED
int BUTTON = 7; // the input pin where the pushbutton is connected
int val = 0; // val will be used to store the state of the input pin
```

```
void setup() {
  pinMode(LED, OUTPUT); // tell Arduino LED is an output
  pinMode(BUTTON, INPUT); // and BUTTON is an input
}
```

```
void loop() {
  val = digitalRead(BUTTON); // read input value and store it

  // Check whether the input is HIGH (button pressed)
  if (val == HIGH) {
    digitalWrite(LED, HIGH); // turn LED on
  } else {
    digitalWrite(LED, LOW); // turn LED off
  }
}
```

## // Project 03 : Turn on LED while the button is pressed and keep it on after it is released

// Source: Getting Started with Arduino, 3rd edition by Massimo Banzi and Michael Shiloh

```
int LED = 13; // the pin for the LED
int BUTTON = 7; // the input pin where the pushbutton is connected
int val = 0; // val will be used to store the state of the input pin
int old_val = 0; // this variable stores the previous value of "val"
int state = 0; // 0 = LED off while 1 = LED on

void setup() {
  pinMode(LED, OUTPUT); // tell Arduino LED is an output
  pinMode(BUTTON, INPUT); // and BUTTON is an input
}

void loop() {
  val = digitalRead(BUTTON); // read input value and store it

  // Check if there was a transition
  if ((val == HIGH) && (old_val == LOW)) {
    state = 1 - state;
  }

  old_val = val; // val is now old, let's store it

  if (state == 1) {
    digitalWrite(LED, HIGH); // turn LED on
  } else {
    digitalWrite(LED, LOW); // turn LED off
  }
}
```

## // Project 03 with debouncing!

Insert just the red text into the code above, as shown here:

.....

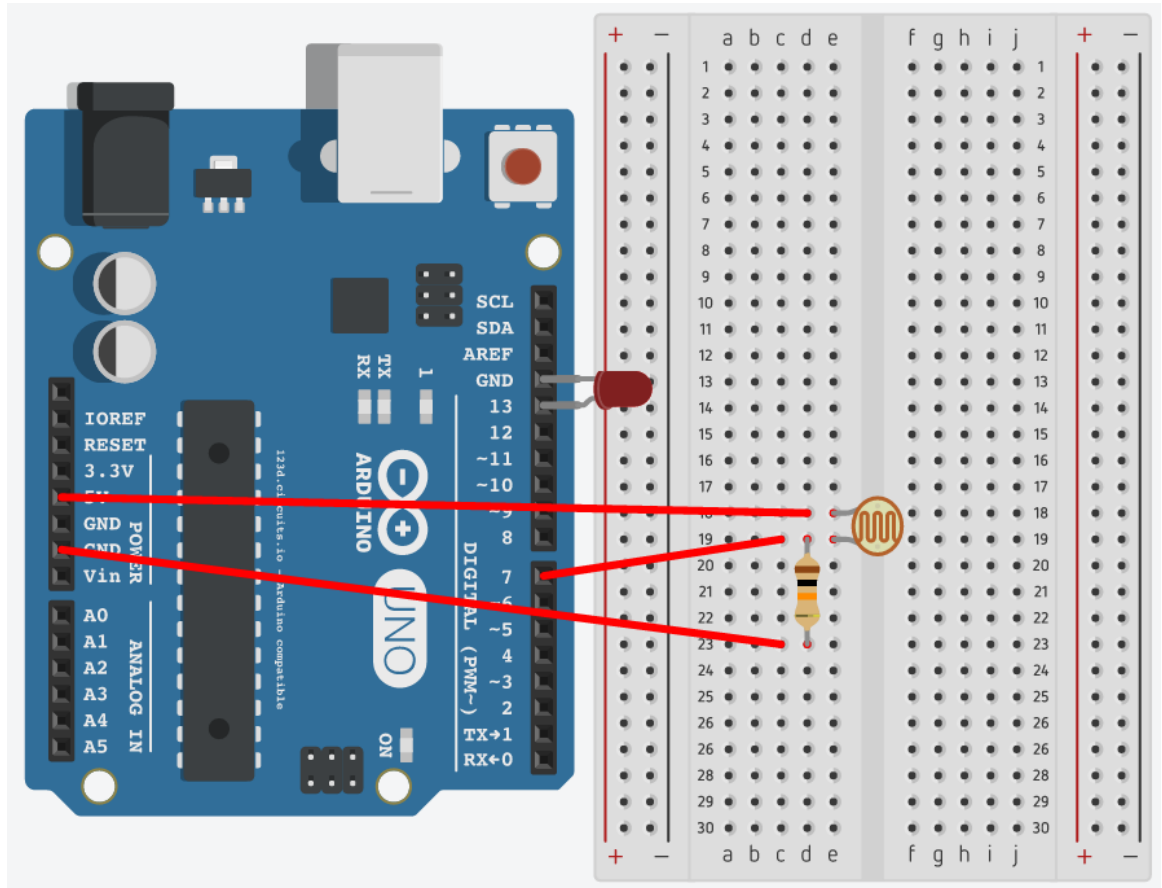
```
// Check if there was a transition
if ((val == HIGH) && (old_val == LOW)) {
  state = 1 - state;
  delay(50);
}
```

.....

## Project 04 : Turn on LED while light is hitting a light dependent resistor

Components needed:

- Arduino Uno board
- LED
- 3 jumper wires
- 10k ohm resistor
- breadboard
- light dependent resistor (sometimes called a *photoresistor*)



## // Project 04 : Turn on LED while light is hitting a light dependent resistor

// Source: Getting Started with Arduino, 3rd edition by Massimo Banzi and Michael Shiloh

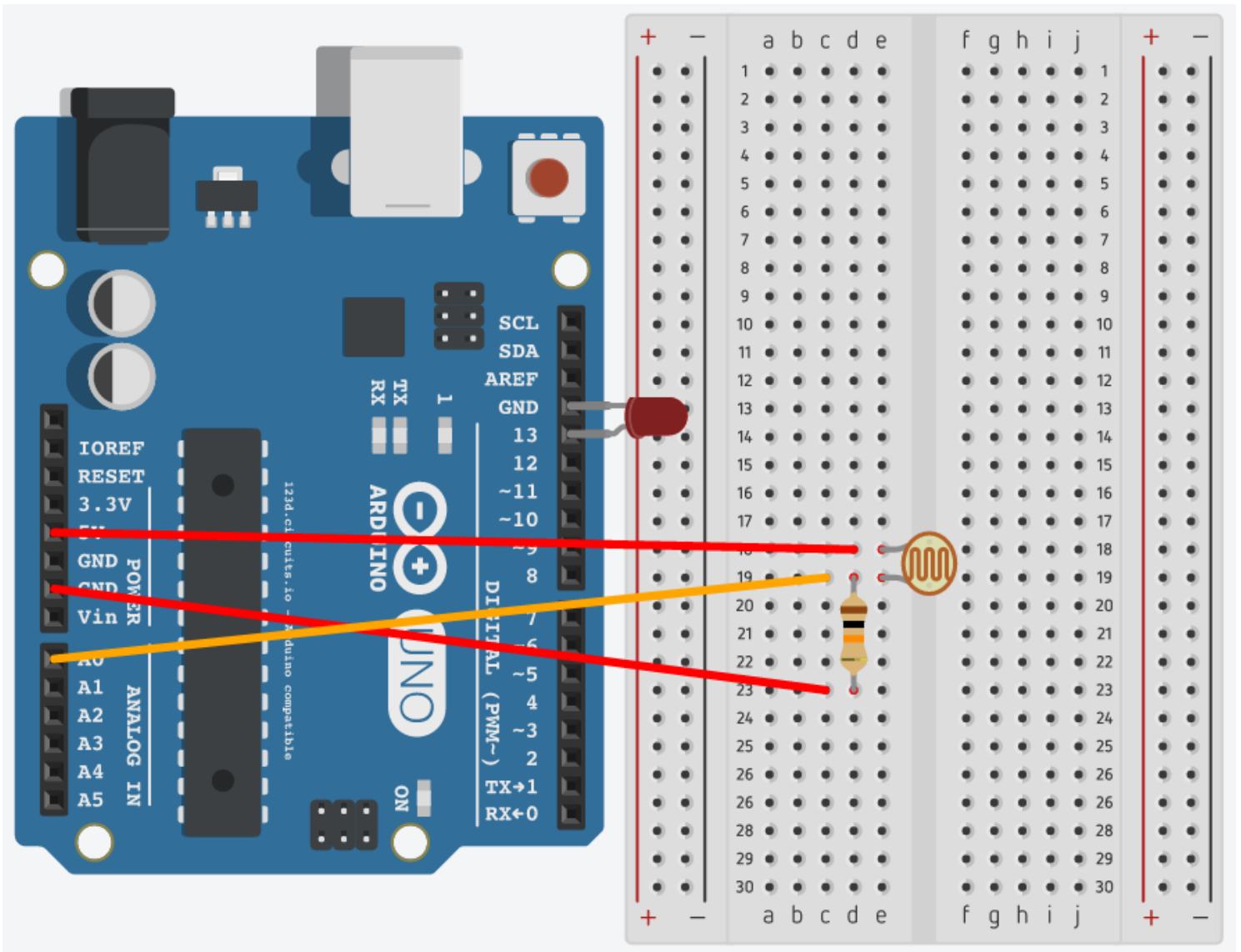
```
int LED = 13; // the pin for the LED
int RESISTOR = 7; // the input pin where the light dependent resistor is connected
int val = 0; // val will be used to store the state of the input pin
void setup() {
  pinMode(LED, OUTPUT); // tell Arduino LED is an output
  pinMode(RESISTOR, INPUT); // and RESISTOR is an input
}
void loop() {
  val = digitalRead(RESISTOR); // read input value and store it

  // Check whether the input is HIGH (light is hitting resistor)
  if (val == HIGH) {
    digitalWrite(LED, HIGH); // turn LED on
  } else {
    digitalWrite(LED, LOW); // turn LED off
  }
}
```

## Project Project 5 : Blink LED at a rate specified by the value of the analog input

Components needed:

- Arduino Uno board
- LED
- 3 jumper wires
- 10k ohm resistor
- breadboard
- light dependent resistor (sometimes called a *photoresistor*)



## // Project 5 : Blink LED at a rate specified by the value of the analog input

// Source: Getting Started with Arduino, 3rd edition by Massimo Banzi and Michael Shiloh

```
int LED = 13; // the pin for the LED

int val = 0; // variable used to store the value coming from the sensor

void setup() {
  pinMode(LED, OUTPUT); // LED is as an OUTPUT

  // Note: analog pins are automatically set as inputs
}

void loop() {
  val = analogRead(0); // read the value from the sensor

  digitalWrite(LED, HIGH); // turn the LED on
  delay(val); // stop the program for some time
  digitalWrite(LED, LOW); // turn the LED off
  delay(val); // stop the program for some time
}
```

## Learn More

Want to learn more about the basics of Arduino? Try these resources:

**Search the Brown County Library catalog for “arduino”** – [click here for a direct link](#).

We highly recommend starting out with this book:

**Getting Started with Arduino** – Massimo Banzi & Michael Shiloh

**Sparkfun – What is an Arduino?:** Sparkfun’s introductory tutorial

[learn.sparkfun.com/tutorials/what-is-an-arduino](http://learn.sparkfun.com/tutorials/what-is-an-arduino)

**Check out the Brown County Library’s page for more resources:** [browncountylibrary/arduino](http://browncountylibrary/arduino)