Creating a fun game with Arduino!

Basics of Arduino Circuits with *Eric Xiao*

Introduction

—

- What is everyone’s experience with Arduino or Programming or Electronics?
- Do you want to play a game?
- What are we doing this workshop?

RULES

—

- Follow along with the workshop
- Ask questions whenever you want
- Try to be creative
  (ignore rule one sometimes)

The area of a circle with radius r is:

$$A = 2\int_{-r}^{r} \sqrt{r^2 - x^2}dx$$

$$\text{let } x = r\sin\theta \implies dx = r\cos\theta d\theta$$

$$= 2\int_{x=-r}^{x=r} \sqrt{r^2 - (r\sin\theta)^2} \cdot r\cos\theta d\theta$$

$$= 2r^2 \int_{-r}^{r} \sqrt{1 - \sin\theta^2} \cdot \cos\theta d\theta$$

$$= 2r^2 \int_{-r}^{r} \cos^2\theta d\theta$$

$$= 2r^2 \int_{-r}^{r} \frac{\cos 2\theta + 1}{2} d\theta$$

$$= r^2 \int_{-r}^{r} (\cos 2\theta + 1) d\theta$$

$$= r^2 \left[\frac{1}{2}\sin 2\theta + \theta\right]_{-r}^{r}$$

$$= r^2 [\sin\theta\cos\theta + \theta]_{-r}^{r}$$

$$= r^2 \left[\frac{x}{r} \cdot \frac{\sqrt{r^2 - x^2}}{r} + \arcsin\left(\frac{x}{r}\right)\right]_{-r}^{r}$$

$$= r^2 \left(\frac{r}{r} \cdot \frac{\sqrt{r^2 - r^2}}{r} + \arcsin\left(\frac{r}{r}\right)\right) - \left(\frac{-r}{r} \cdot \frac{\sqrt{r^2 - (-r)^2}}{r} + \arcsin\left(\frac{-r}{r}\right)\right)$$

$$= r^2 \left(\left(\frac{\pi}{2}\right) - \left(-\frac{\pi}{2}\right)\right)$$

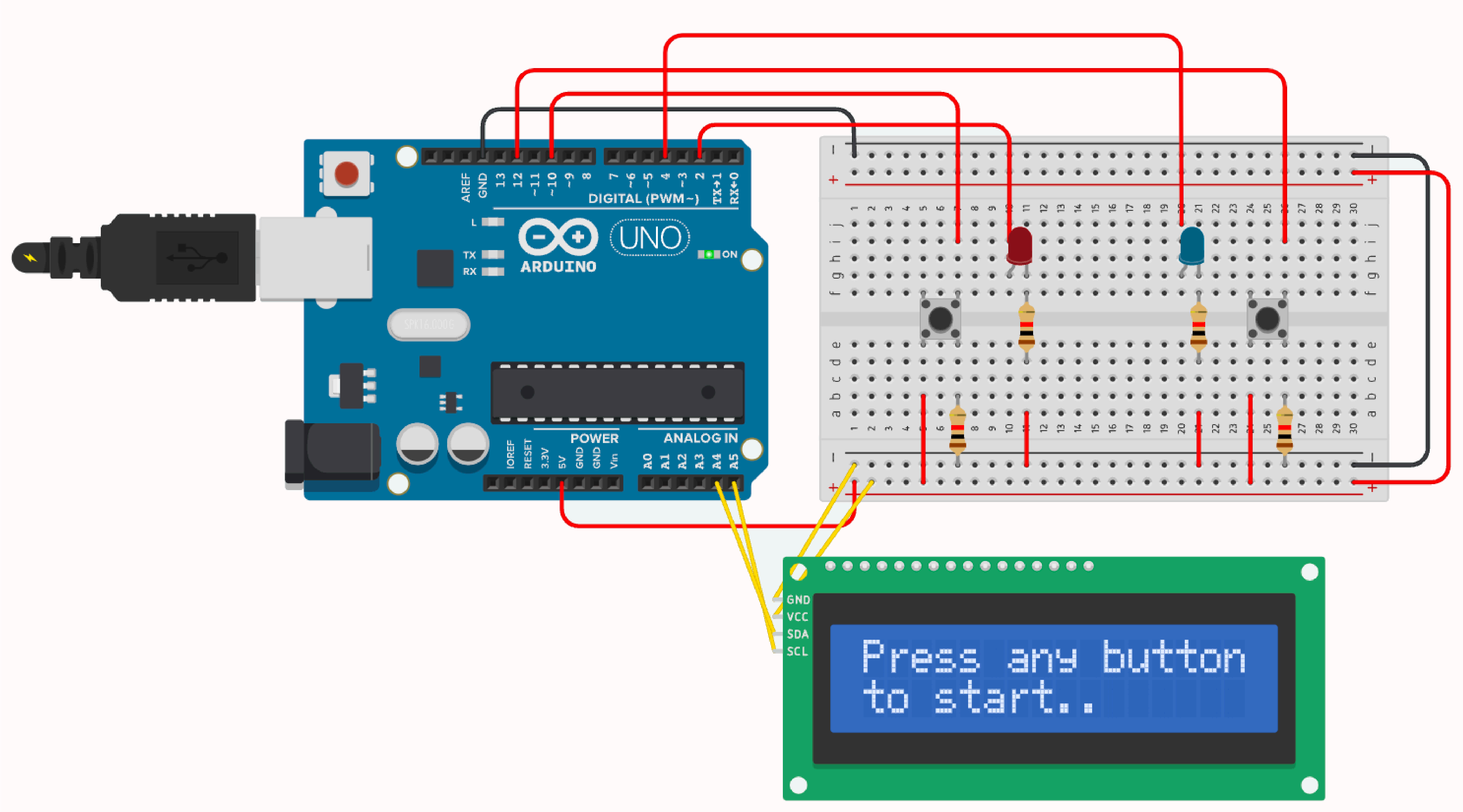
$$= r^2 \pi$$

$$\boxed{A = \pi r^2}$$

Outline of Workshop

—

- Basic Circuits and Electrical Wiring
- Integration of Arduino
- Arduino Programming
- LCD I2C Protocol
- Wiring the Circuit
- Programming the Project

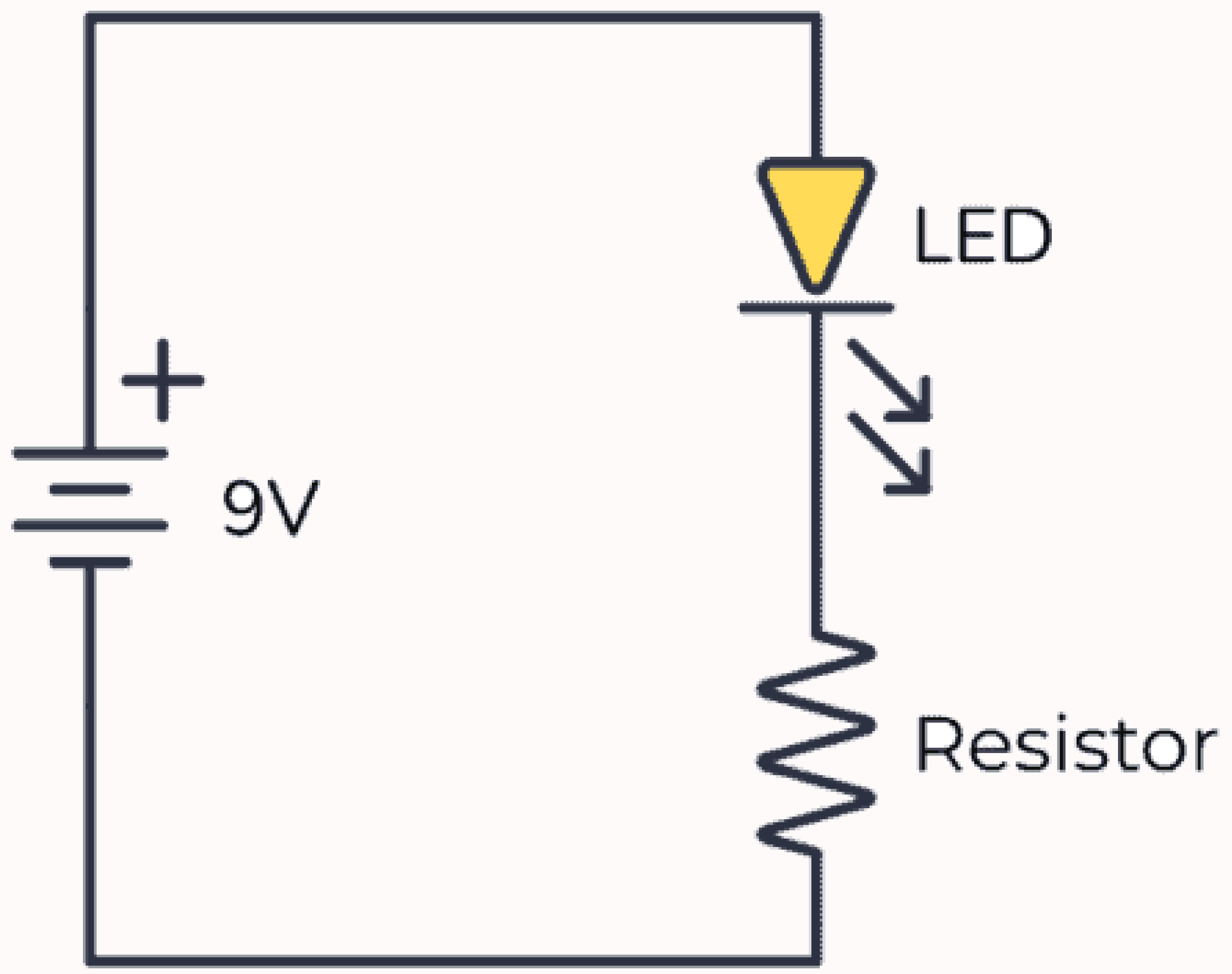


Electricity

—

- **Current** = How much electricity flows per second
- **Voltage** = Potential electricity
- **Resistance** = How much electricity a material stops

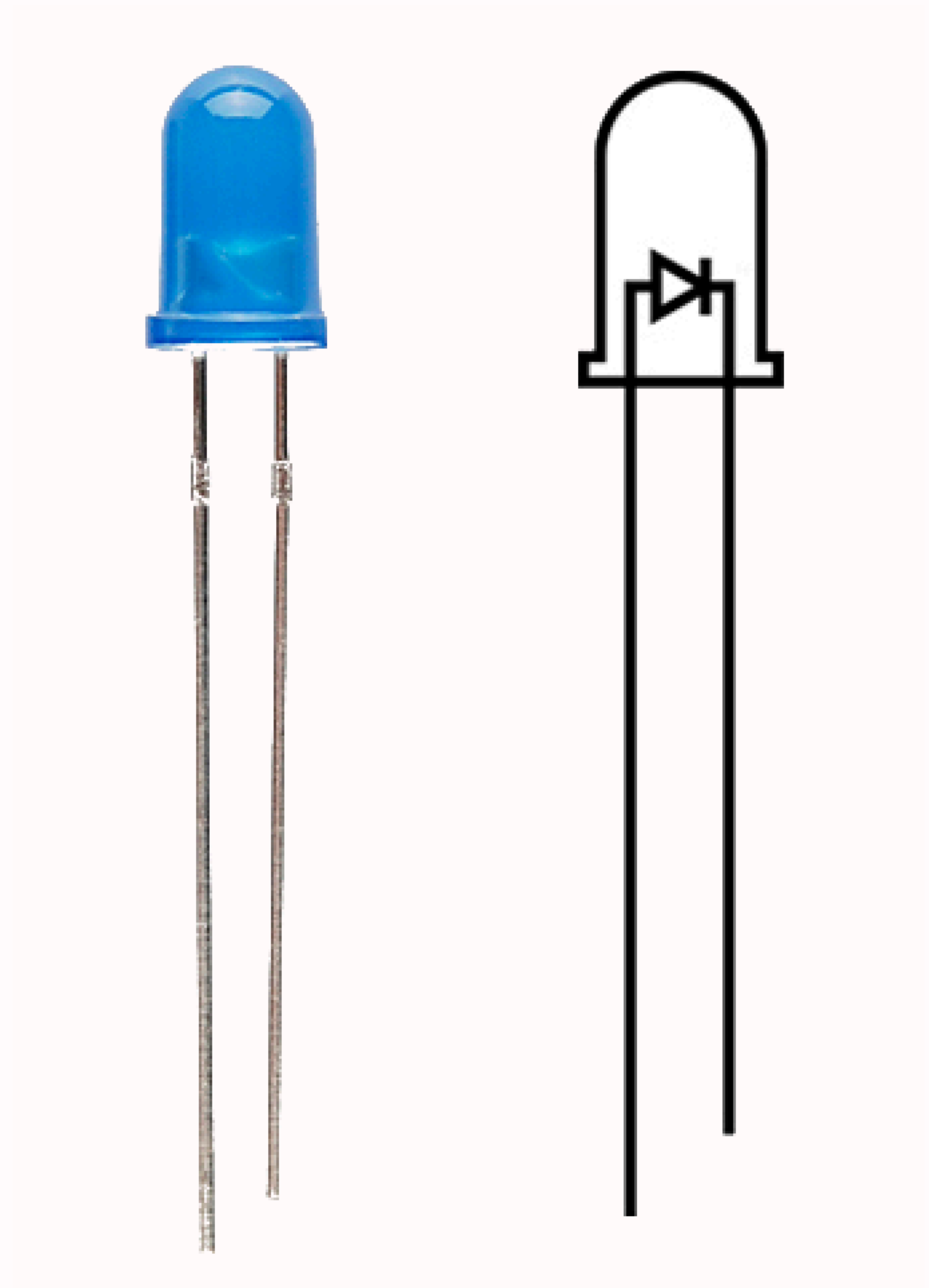
- Ohm's Law: $V = IR \implies I = \frac{V}{R}$
- Too much current = some components overheat and break
  - LEDs, Arduino, etc.



LEDs

—

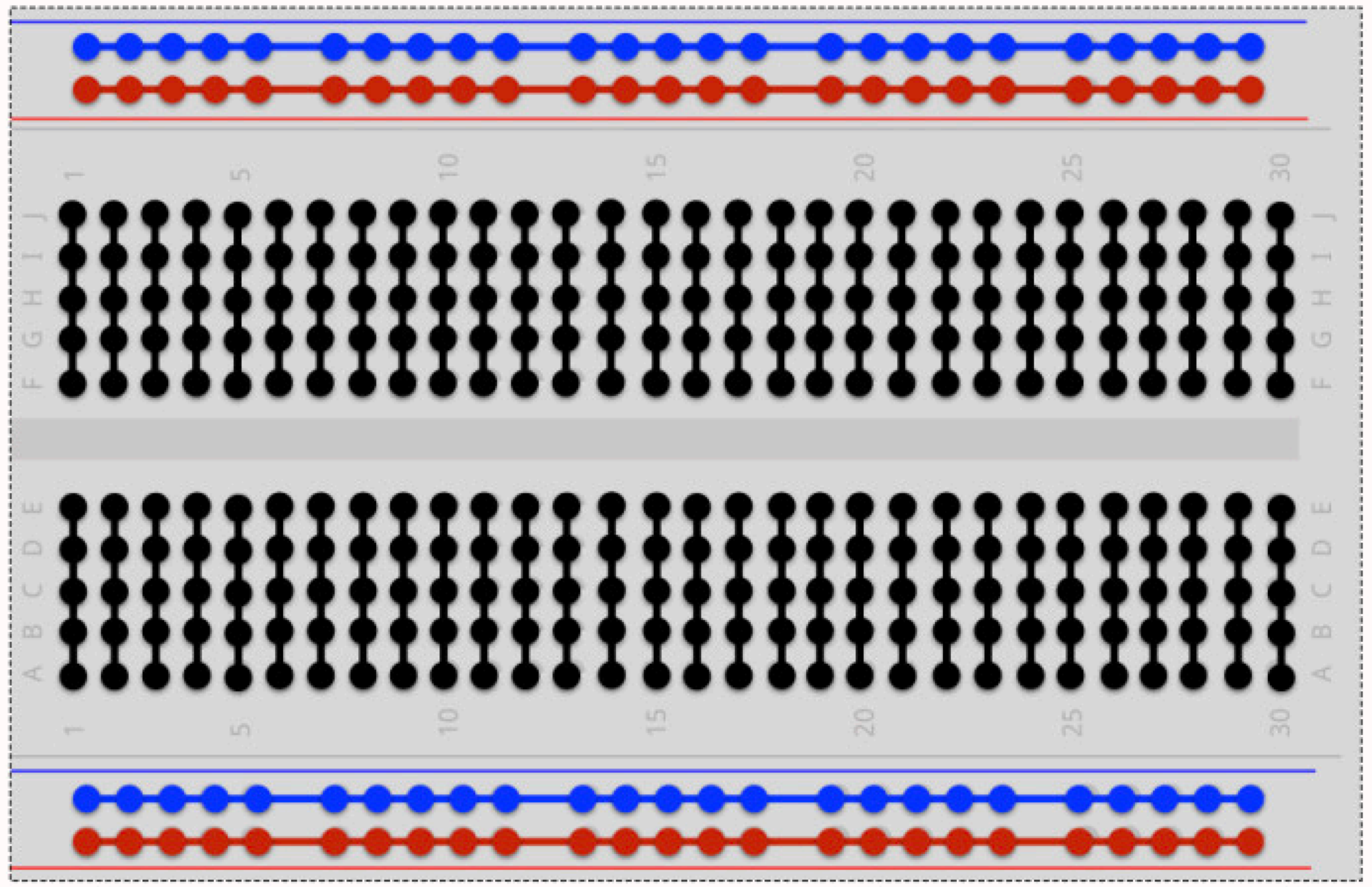
- **Light-emitting diode**
- Diode lets current go through *one direction*
- LED = *“thing that lights up when current goes through in a certain direction”*
- Longer end to high voltage (**5V**), shorter end to low voltage (**Ground**)



Circuits and Wiring

—

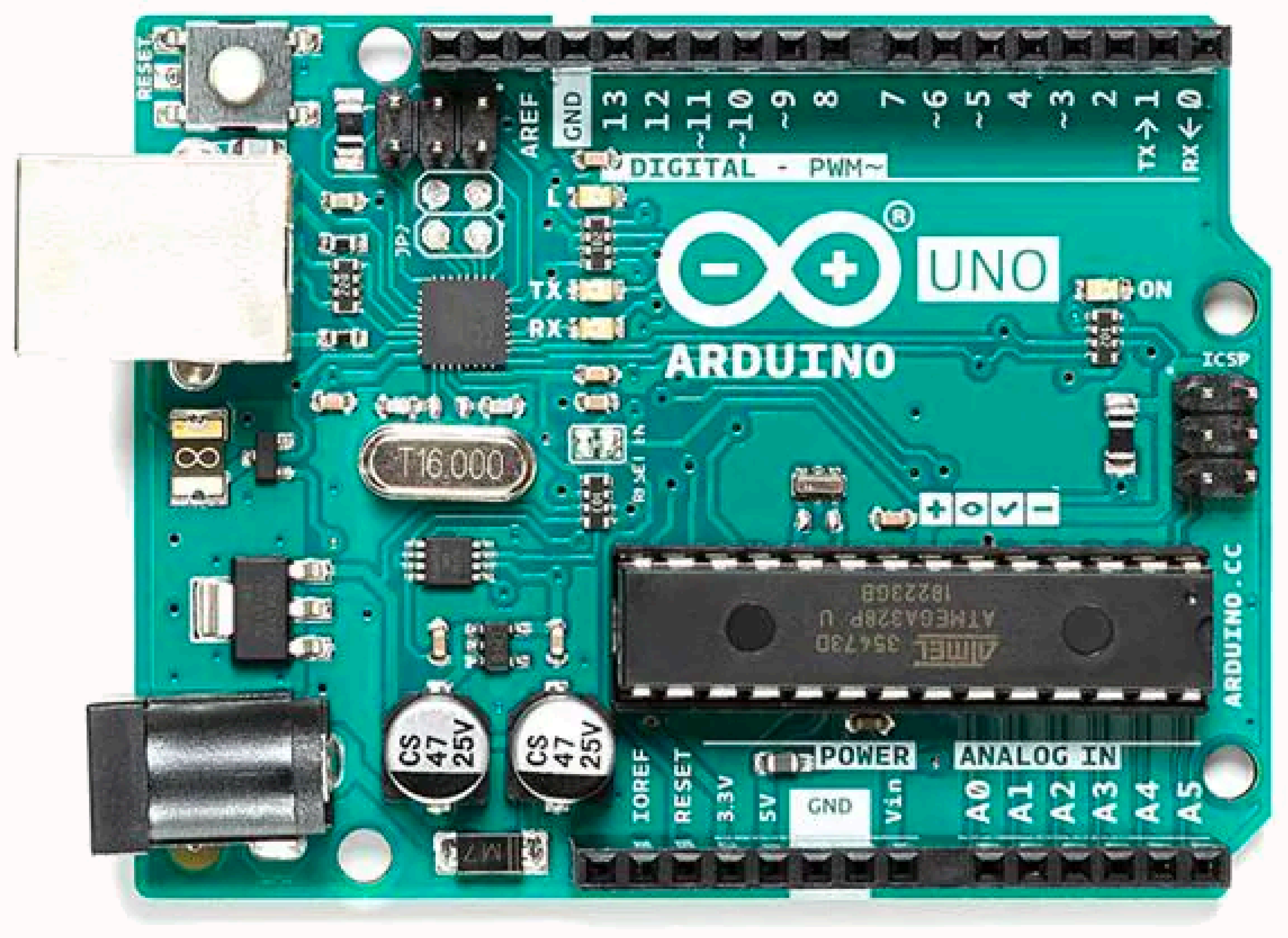
- Circuit on **breadboard**, with wires and other components
  - LEDs, Buttons, LCDs, Arduino, resistor, etc.
- Breadboards make wiring *easy to change*
  - Connects wires just by plugging in
  - Blue is connected, red is connected, black is connected
  - Great for learning & projects



Arduino

–

- Has pins for *output* and *input*
- Has pins for **ground** and **V5** / **V3.3** (constant)
- Computer connects to upload code
- Arduino runs code that is uploaded, with any power source



Arduino Code

–

- Uses Arduino language (C++ with special built-in functions)
  - *digitalWrite*(…), *delay*(…), *analogRead*(…), etc.
- Runs **setup**, then runs **loop** function until it is off

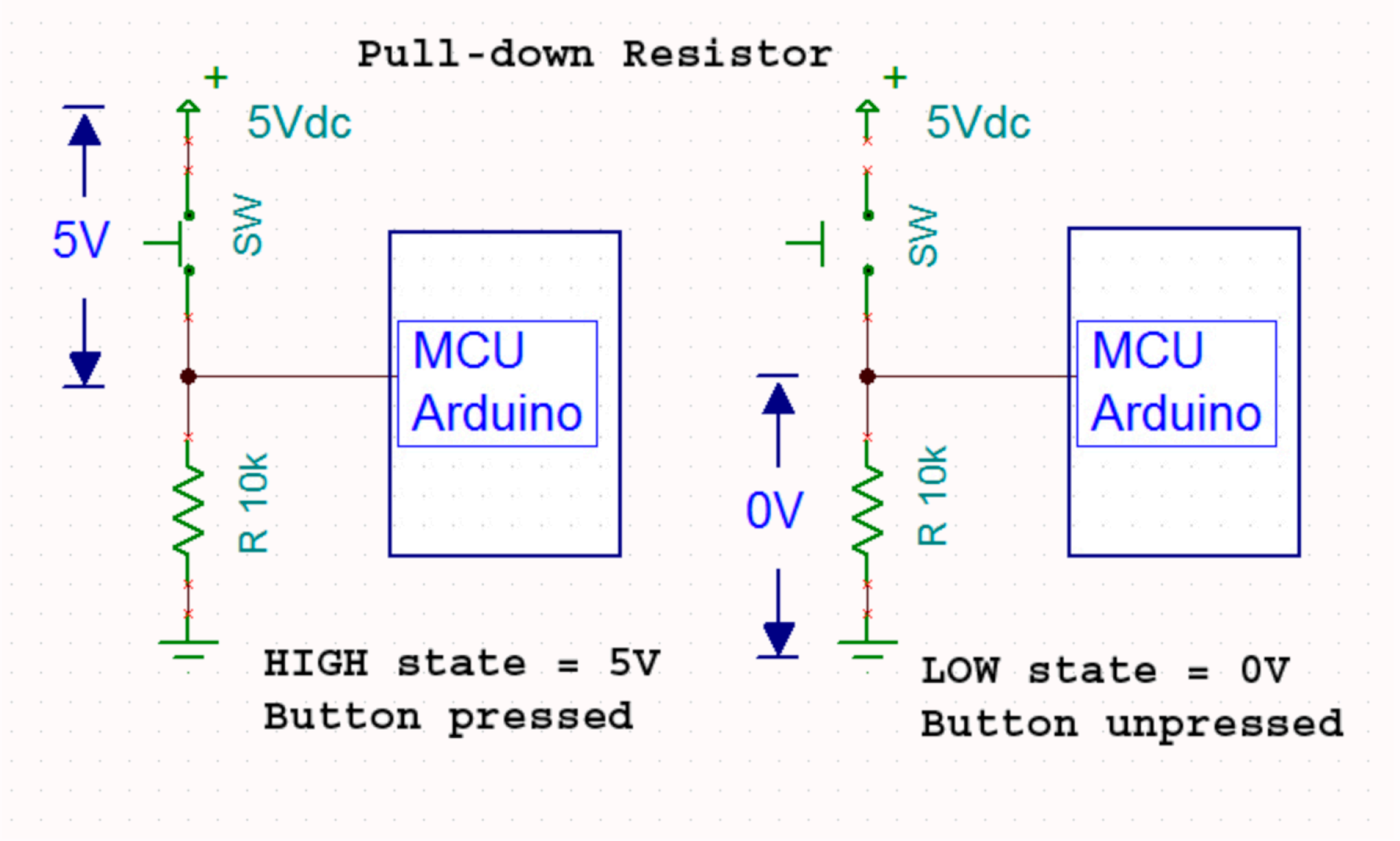
Code that turns LED on and off

```
const int LED_PIN = 6; // digital 6 pin

void setup() {
  pinMode(LED_PIN, OUTPUT); // set pin to output
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // turn on
  delay(1000); // in milliseconds

  digitalWrite(LED_PIN, LOW); // turn off
  delay(1000); // in milliseconds
}
```

Buttons

—

- Controls when wires are connected
- *"Pull-Down Resistors"* are necessary when using buttons for input
  - Get rid of excess charge (sends it to **ground**)



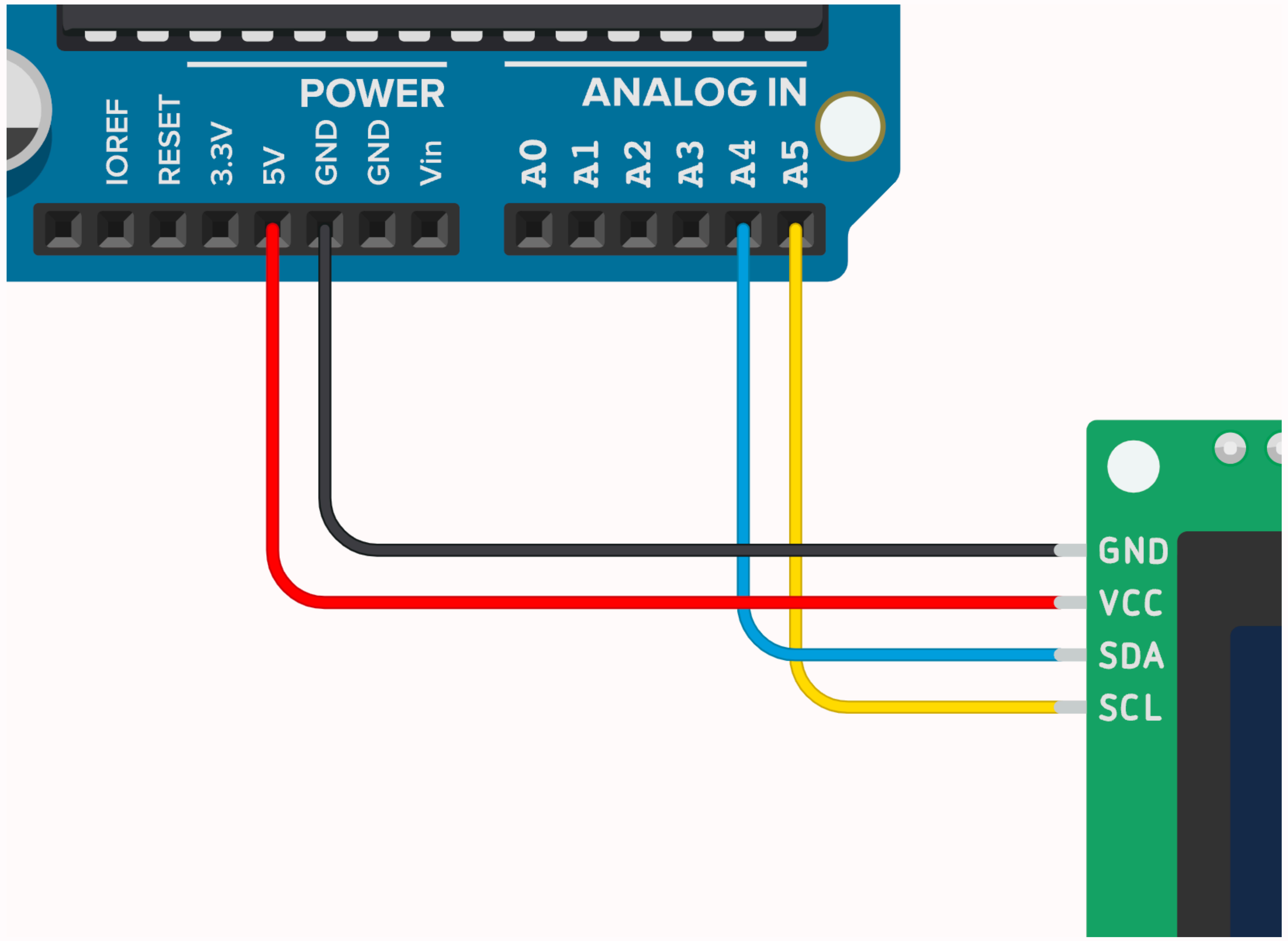
Code with Button Logic

```
const int BUTTON_PIN = 10;
const int LED_PIN = 6;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT); // set pin to input
}

void loop() {
   //checks if there is high or low input to the pin
  if (digitalRead(BUTTON_PIN) == HIGH) {
     // flash light once
    digitalWrite(LED_PIN, HIGH);
    delay(100);
    digitalWrite(LED_PIN, LOW);
    delay(100);
  }
}
```

LCD I2C Protocol

—

- Uses two signals
  - SDA used to transmit data
  - SCL used to synchronize data
- Two other pins are…
  - VCC, just constant voltage
  - GND, just zero voltage
- With LCD, the data transmitted is the text to display



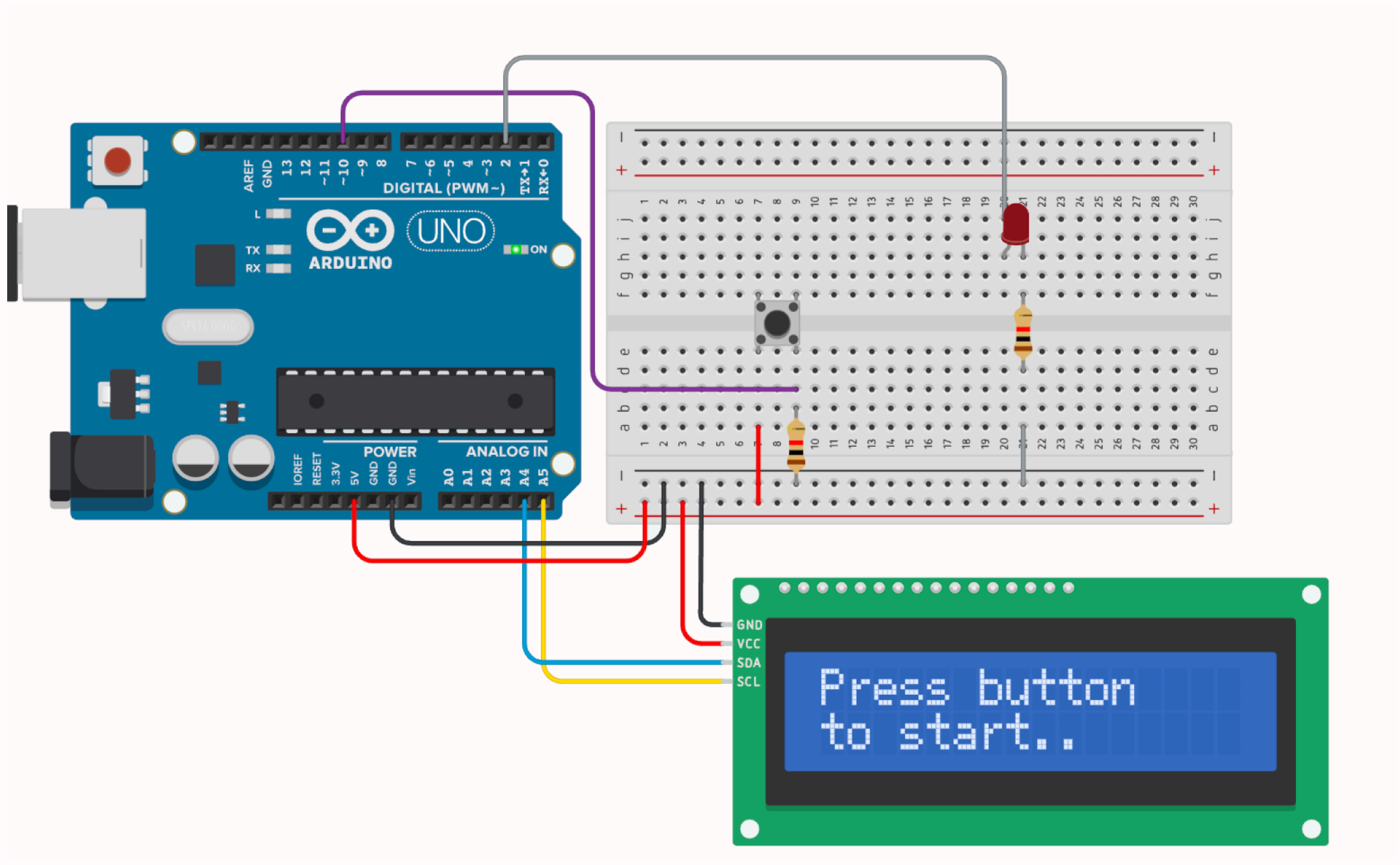
LCD I2C Code

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // port, rows, columns

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.print("Starting the LCD");
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print("                "); // clear line
  lcd.setCursor(0, 1);
  lcd.print(millis()); // the time in milliseconds
  lcd.print("ms");
  delay(200);
}
```

Reaction Time Game

–

- Wait for an LED to turn on
- Player presses button ASAP
- Displays their reaction time
- That’s it!
- But will still be challenging..



```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int BUTTON_PIN = 10;
const int LED_PIN = 2;

long ledStartTime; // stores start time of LED turning on

void setup() {
  pinMode(BUTTON_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.print("Press button");
  lcd.setCursor(0, 1);
  lcd.print("to start..");

  reset();
}
```

```
void waitForClick() {
  // wait for button release
  while (true) {
    if (digitalRead(BUTTON_PIN) == LOW) {
      break;
    }
  }
  // wait for button down
  while (true) {
    if (digitalRead(BUTTON_PIN) == HIGH) {
      break;
    }
  }
  // wait for button release
  while (true) {
    if (digitalRead(BUTTON_PIN) == LOW) {
      break;
    }
  }
}
```

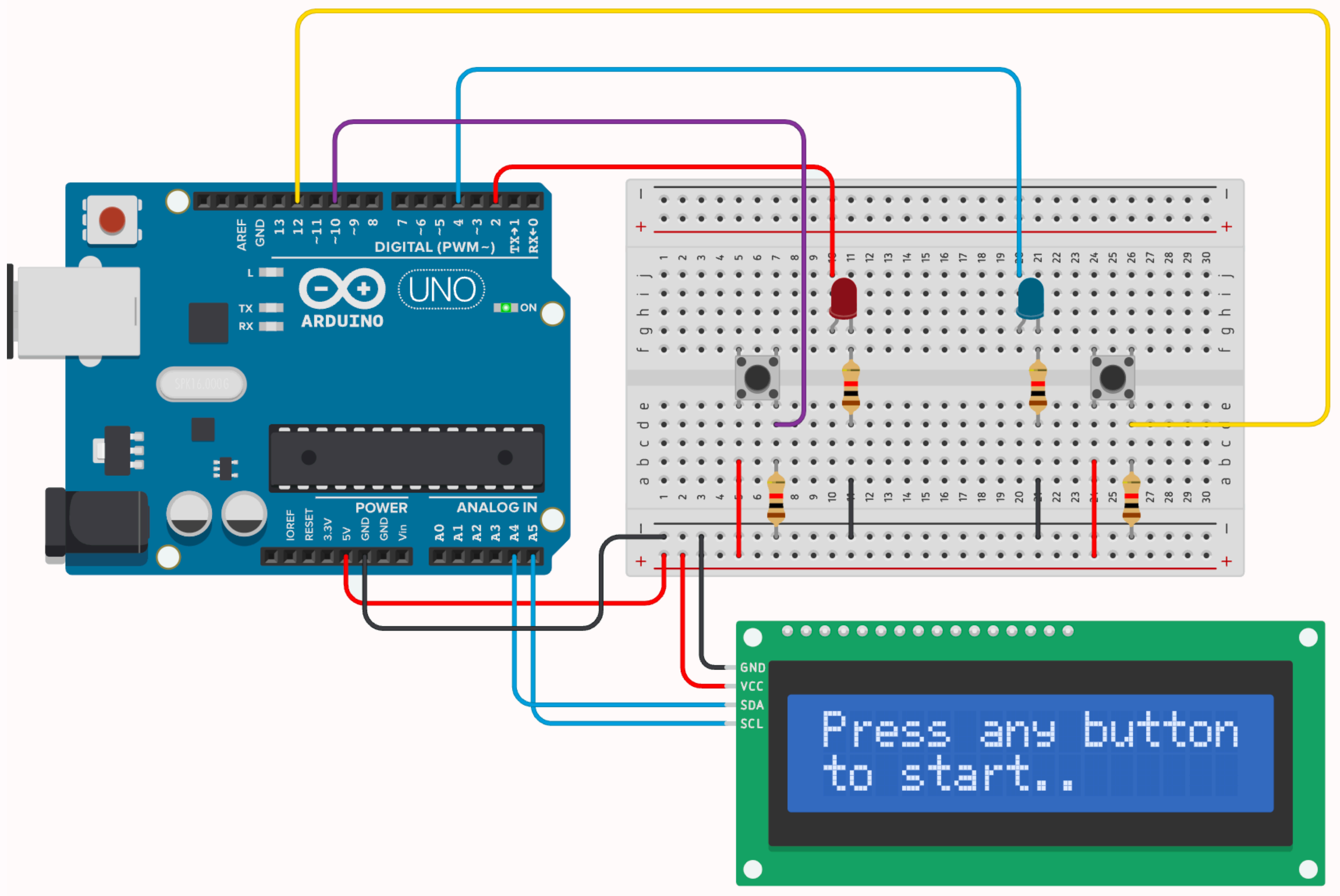
```
void reset() {
  waitForClick(); // click to reset

  digitalWrite(LED_PIN, LOW);
  lcd.clear();
  lcd.print("WAIT FOR LIGHT!");

  ledStartTime = millis() + random(1000, 3000);
}

int getClickTime() {
  waitForClick();
  return millis() - ledStartTime;
}
```

```
void loop() {
  if (digitalRead(BUTTON_PIN) == HIGH) { // check if clicked early
    lcd.clear();
    lcd.print("TOO EARLY");
    reset();
  }
  if (millis() >= ledStartTime) { // check if time to start
    lcd.clear();
    lcd.print("CLICK!");
    digitalWrite(LED_PIN, HIGH);

    int resultTime = getClickTime();

    lcd.clear();
    lcd.print("YOUR TIME: ");
    lcd.setCursor(0, 1);
    lcd.print(resultTime);
    lcd.print("ms");
    reset();
  }
}
```

Reaction Time Game 2 Buttons

—

- Wait for an LED to turn on
- Player presses **correct** button as fast as possible
- Displays their reaction time, or states they pressed the wrong button
- A bit more complicated…?



```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int BUTTON_PIN_1 = 10;
const int BUTTON_PIN_2 = 12;
const int LED_PIN_1 = 2;
const int LED_PIN_2 = 4;

int targetButton;
long ledStartTime;

void setup() {
  pinMode(BUTTON_PIN_1, INPUT);
  pinMode(BUTTON_PIN_2, INPUT);
  pinMode(LED_PIN_1, OUTPUT);
  pinMode(LED_PIN_2, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.print("Press any button");
  lcd.setCursor(0,1);
  lcd.print("to start..");
  reset();
}
```

```
int waitForClick() {
  int pressedButton;
  while (true) {
    if (digitalRead(BUTTON_PIN_1) == LOW && digitalRead(BUTTON_PIN_2) == LOW) {
      break;
    }
  }
  while (true) {
    if (digitalRead(BUTTON_PIN_1) == HIGH) {
      pressedButton = 1;
      break;
    }
    if (digitalRead(BUTTON_PIN_2) == HIGH) {
      pressedButton = 2;
      break;
    }
  }
  while (true) {
    if (digitalRead(BUTTON_PIN_1) == LOW && digitalRead(BUTTON_PIN_2) == LOW) {
      break;
    }
  }
  return pressedButton;
}
```

```
int getClickTime() {
  int pressedButton = waitForClick();
  if (pressedButton == targetButton) {
    return millis() - ledStartTime;
  } else {
    return -1;
  }
}

void reset() {
  waitForClick();

  digitalWrite(LED_PIN_1, LOW);
  digitalWrite(LED_PIN_2, LOW);
  lcd.clear();
  lcd.print("WAIT FOR LIGHT!");

  ledStartTime = millis() + random(1000, 3000);
}
```

```
void loop() {
  if (digitalRead(BUTTON_PIN_1) == HIGH || digitalRead(BUTTON_PIN_2) == HIGH) {
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("TOO EARLY");
    reset();
  }
  if (millis() >= ledStartTime) {
    lcd.clear();
    lcd.print("CLICK!");
    targetButton = random(2) + 1; // (0 or 1) + 1 = 1 or 2
    if (targetButton == 1) {
      digitalWrite(LED_PIN_1, HIGH);
    } else {
      digitalWrite(LED_PIN_2, HIGH);
    }

    int resultTime = getClickTime();

    lcd.clear();
    if (result == -1) {
      lcd.print("WRONG BUTTON");
    } else {
      lcd.print("YOUR TIME: ");
      lcd.setCursor(0, 1);
      lcd.print(resultTime);
      lcd.print("ms");
    }

    reset();
  }
}
```

THANK YOU FOR COMING TO THE WORKSHOP

consider

learning more about programming

learning more about electronic circuits

learning more about your interests

learning more