Sensing & Acting

Physical Computing

Acting

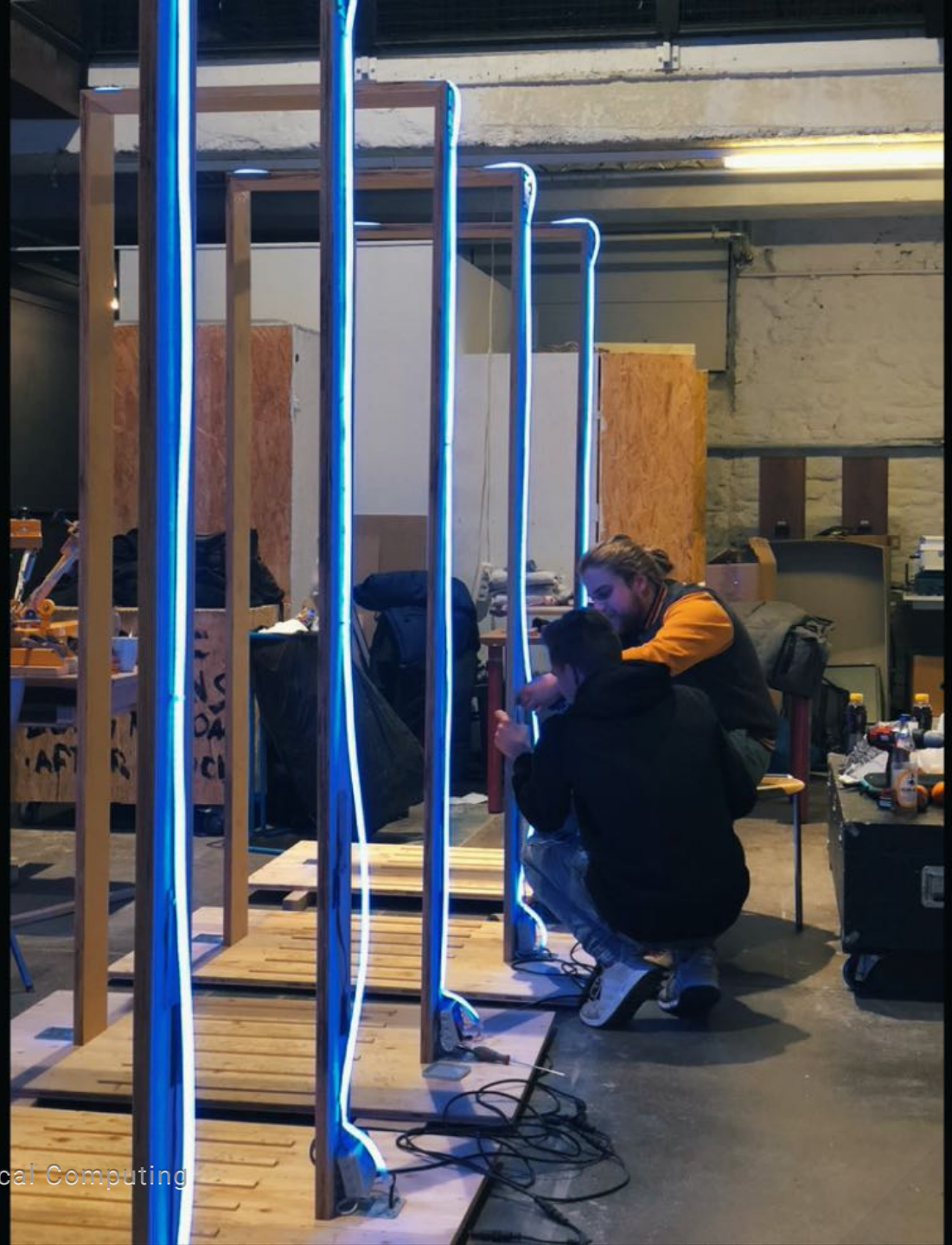
- Digital
  - `digitalWrite(...)`
  - `HIGH` / `LOW`
- Analog
  - `analogWrite(...)`
  - `0-255`
  - PWM

Source: Casey Horner

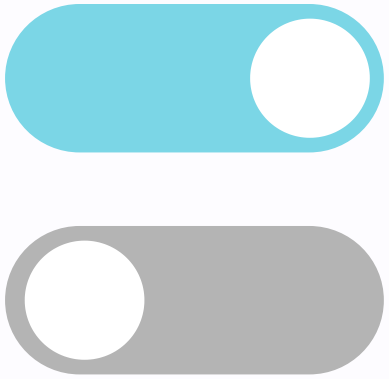
Sensing

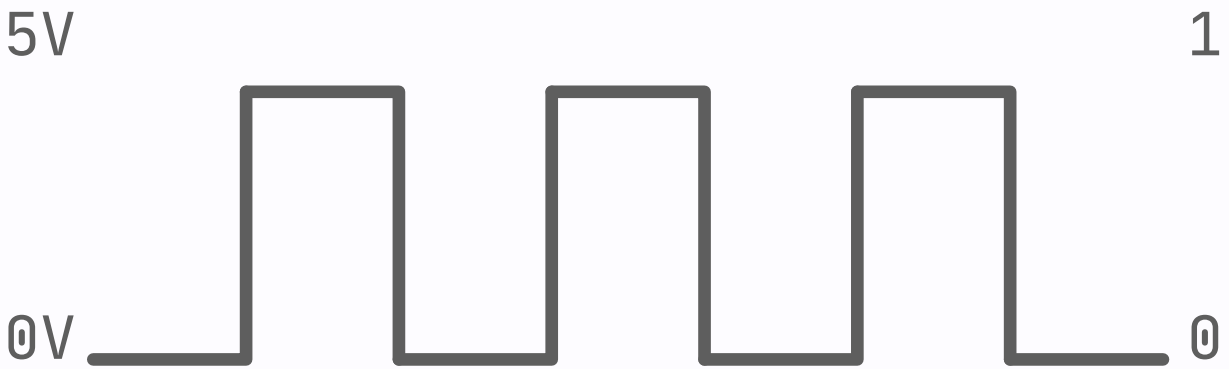
- Sensors
  - Light
  - Distance
  - Temperature
  - Pressure
  - Sonic
  - ...
- Input Types
  - Digital
  - Analog

Source: Florian Bruggisser

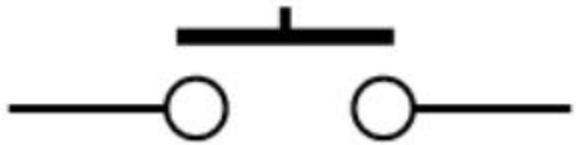


Digital Signal



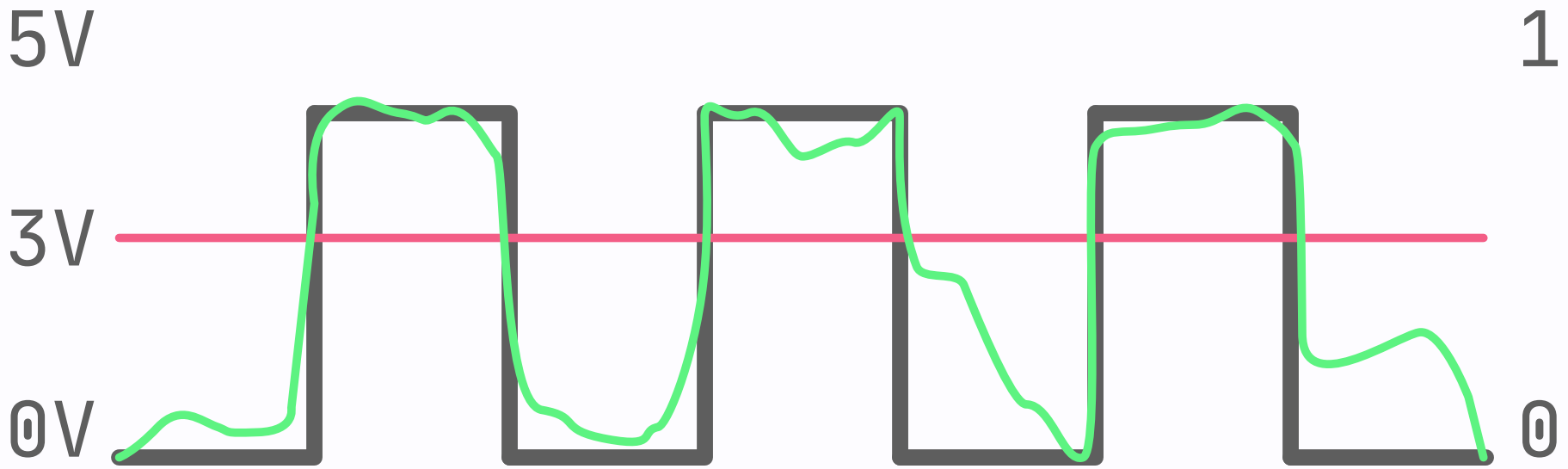


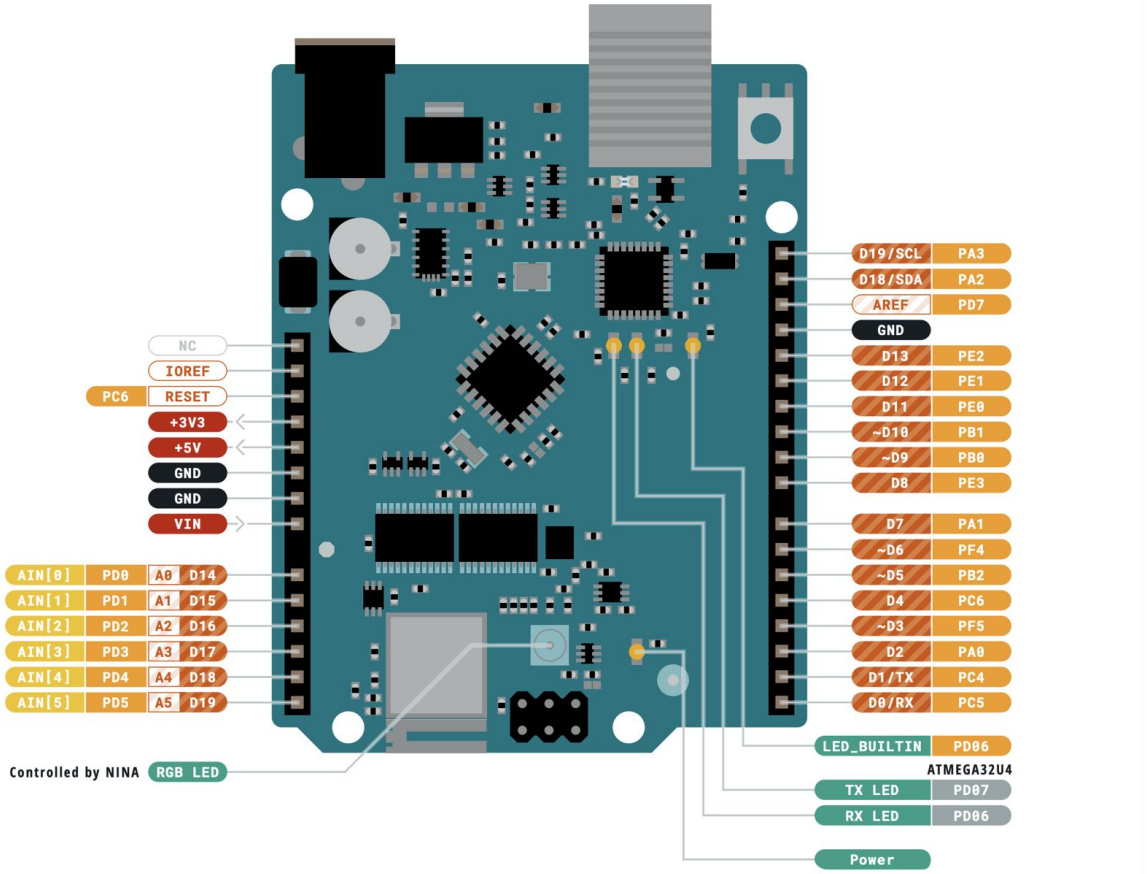
Button Tactile Switch



Digital Sensors

Threshold



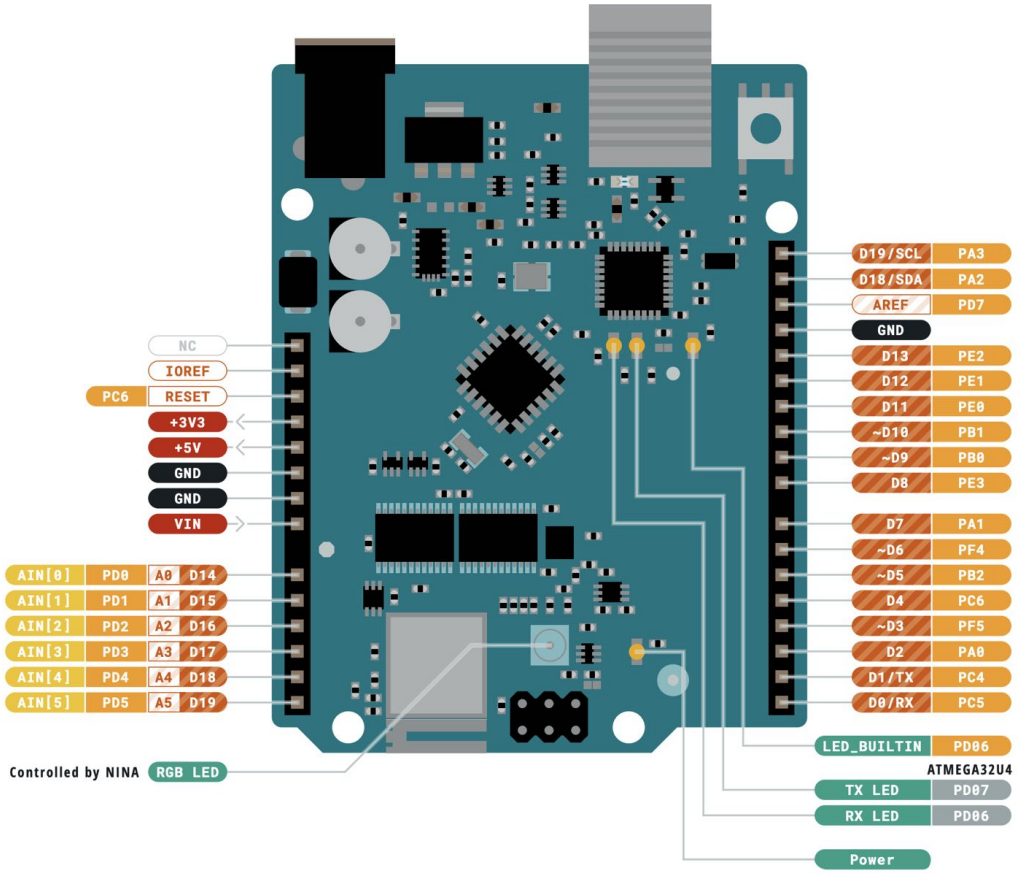


Pin Mode

`void pinMode(pin, mode)`

- Sets the **pin** mode
- `INPUT`, `OUTPUT`

Source: Ardunio



Digital Read

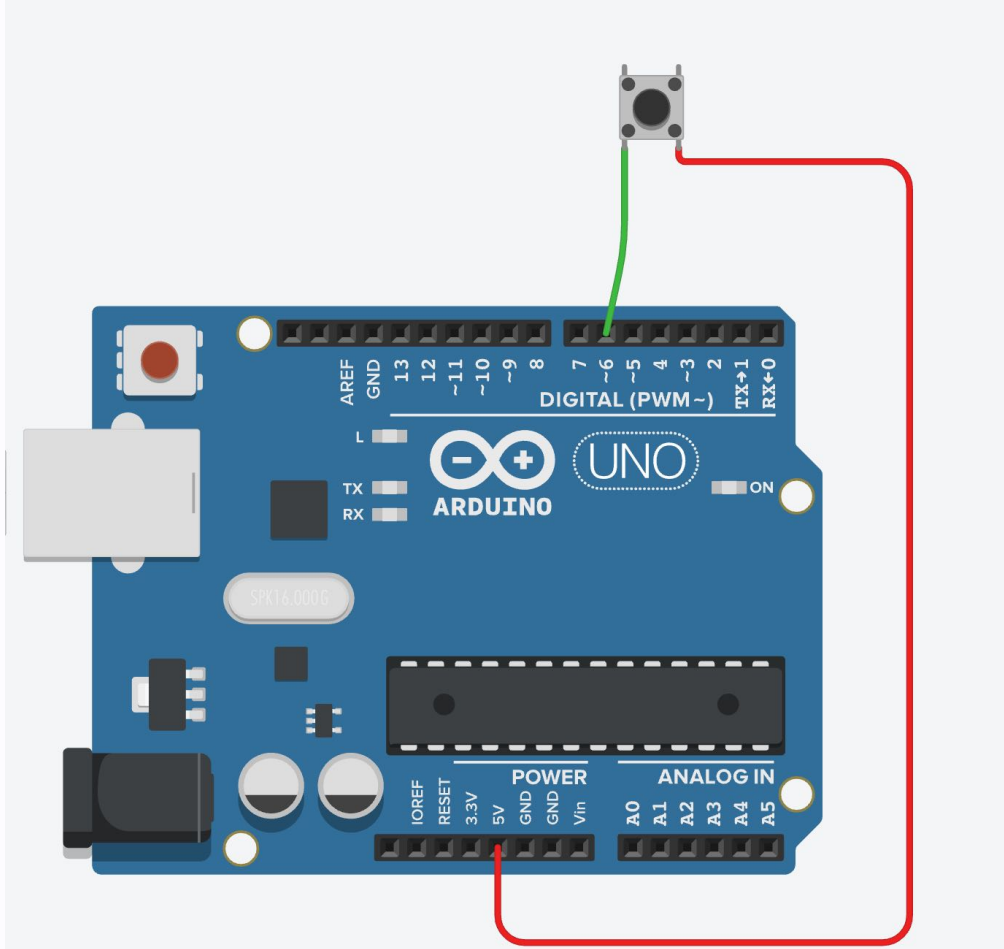
```
int digitalRead(pin, value)
```

- `HIGH` or `LOW` from **pin**
- `HIGH` = `1`
- `LOW` = `0`

Source: Ardunio

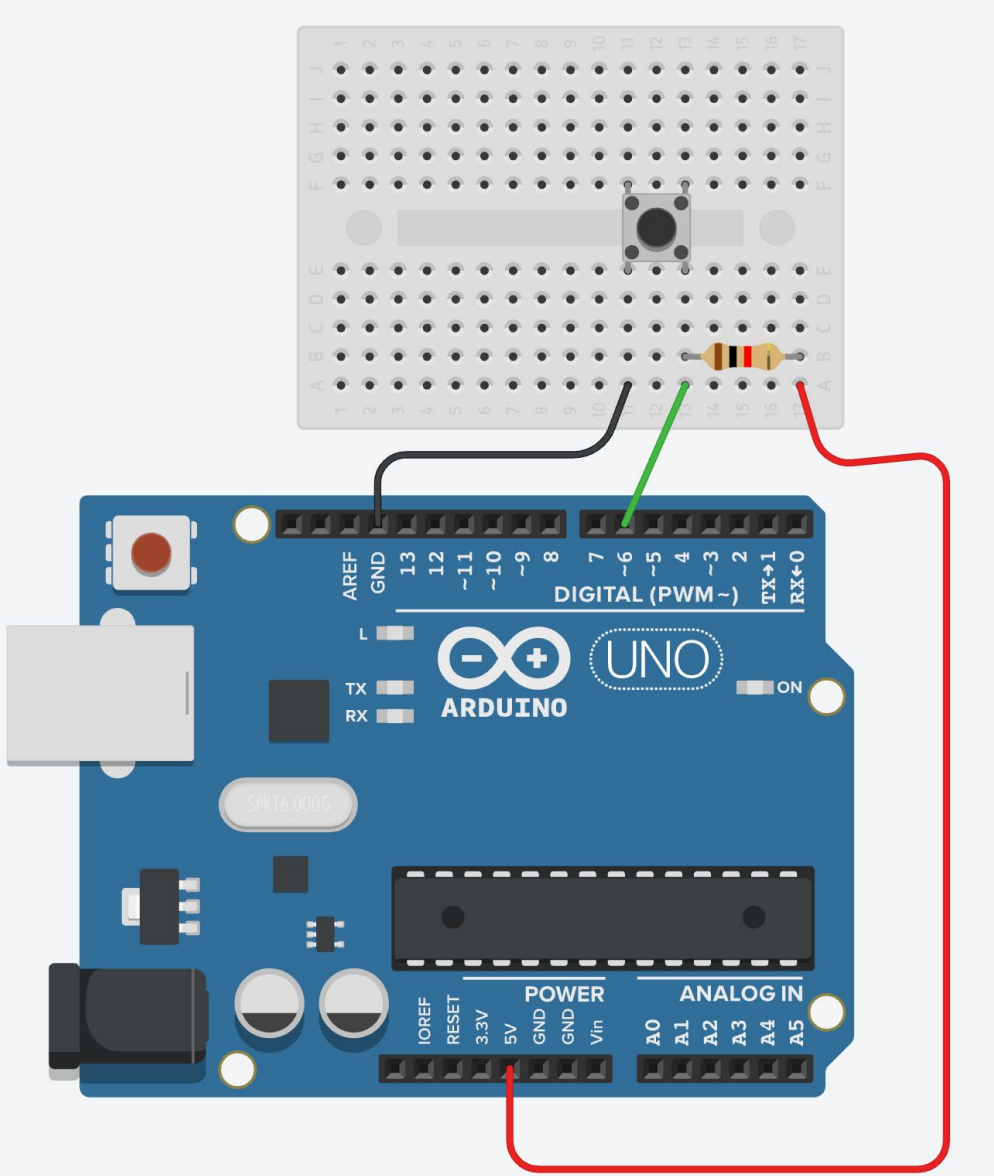
Button Tactile Switch

- Simple circuit (Demo)
- Random Input Values
  - Electromagnetic interference (EMI)
  - Floating PIN while `LOW`



Pull-Up Resistor Project

- Keep floating PIN stable
- Pull-Up
  - On press input is `LOW`
  - 5V - 1k ohm
  - Supported by Arduino MC
  - `pinMode(INPUT_PULLUP, 6)`
- Inverted Logic (Pull-Down) ⚡



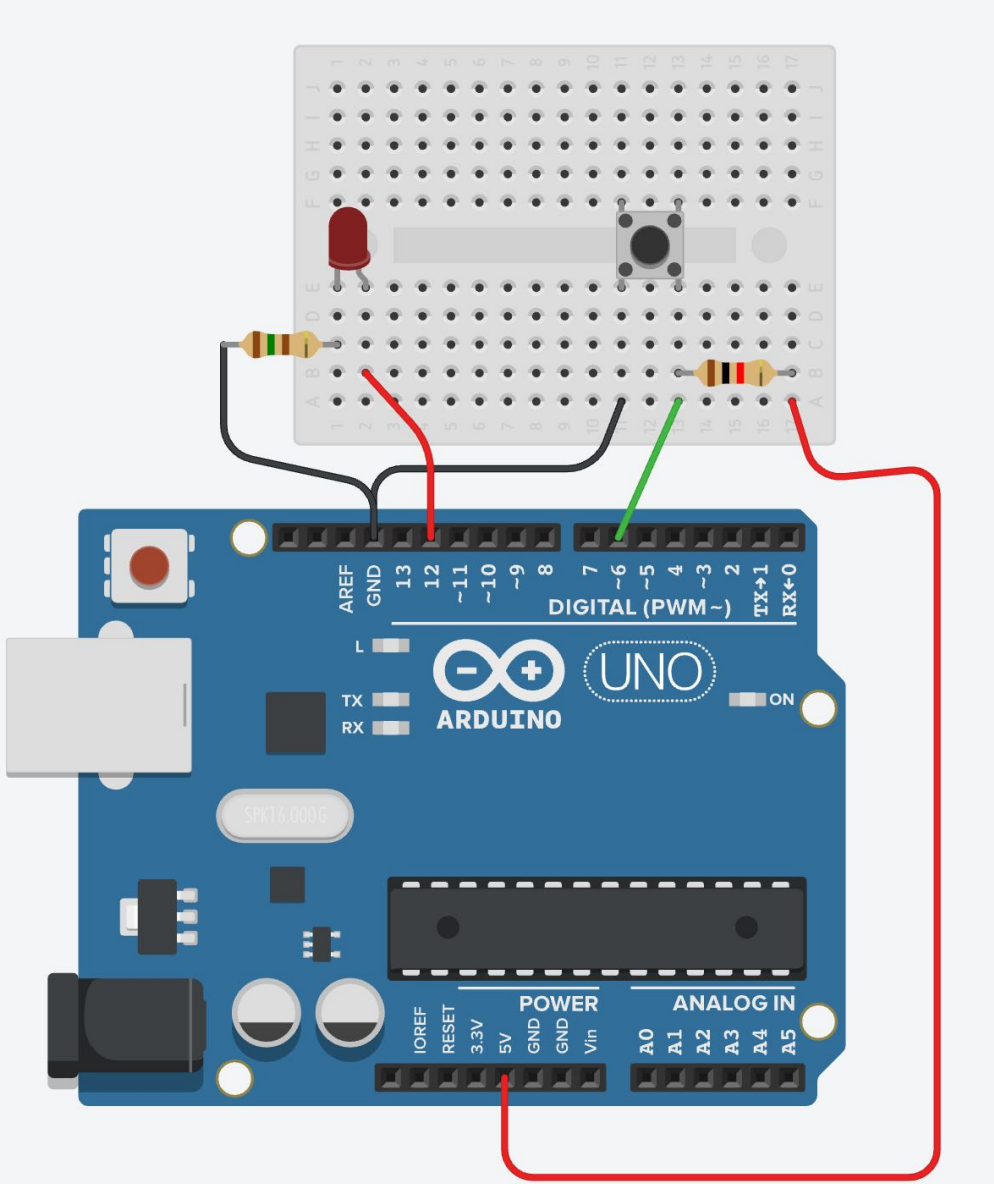
👩‍💻 Task 1: Light Button (10min)

Use a button to **turn on** an LED **when pressed** and turn it off **when released**.

🌻 Task 1: Light Button

Solution

```
void setup()
{
  pinMode(INPUT, 6);
  pinMode(OUTPUT, 12);
}

void loop()
{
  int value = digitalRead(6);
  digitalWrite(12, 1 - value);
  delay(250);
}
```

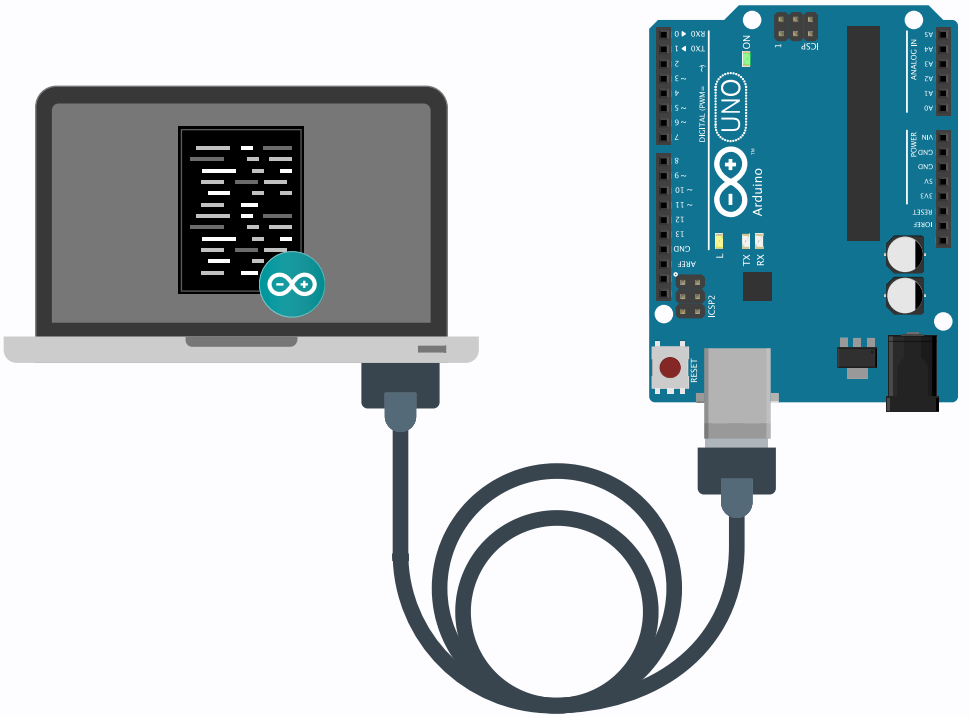


USB 💻


What does USB stand for?
Universal Serial Bus
You're thinking of a bus full of cereal flying through space aren't you?
Yes.
No.
Yes.
©Li Chen
www.exocomics.com

Serial Debugging

```
void setup() {
  // setup serial connection
  Serial.begin(9600);
}

void loop() {
  // print something out
  Serial.print("hello world");
  Serial.print(25);
  Serial.println("interaction");

  int x = 3;
  Serial.println(x);
}
```



👩‍💻 Task 2: Light Switch (10min)

Use your existing wiring to develop a **toggle switch**, which turns **on** the LED on *first button press* and turns it **off** *after the second*.

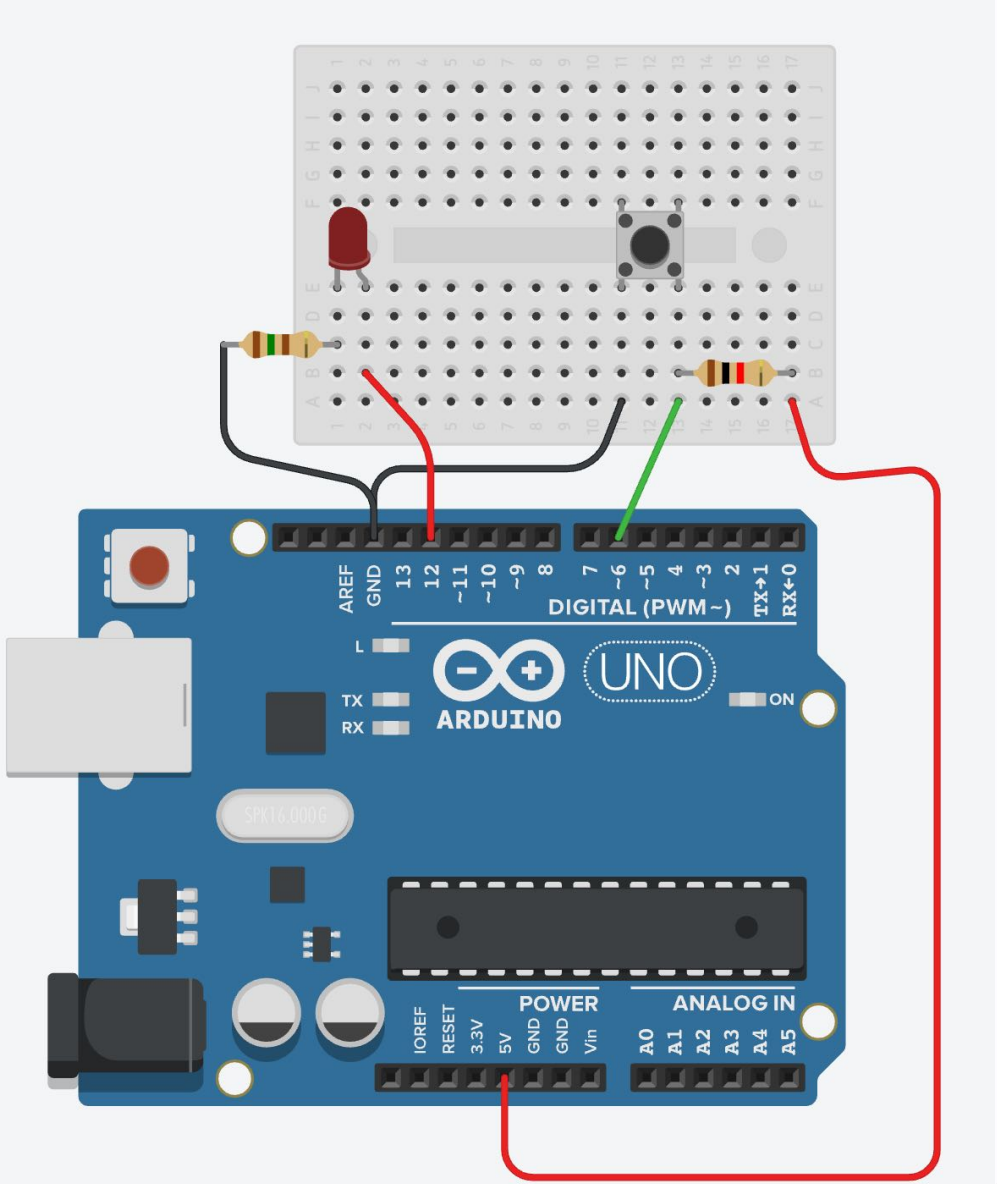
Use the `Serial` to debug your code!

🌻 Task 2: Light Switch

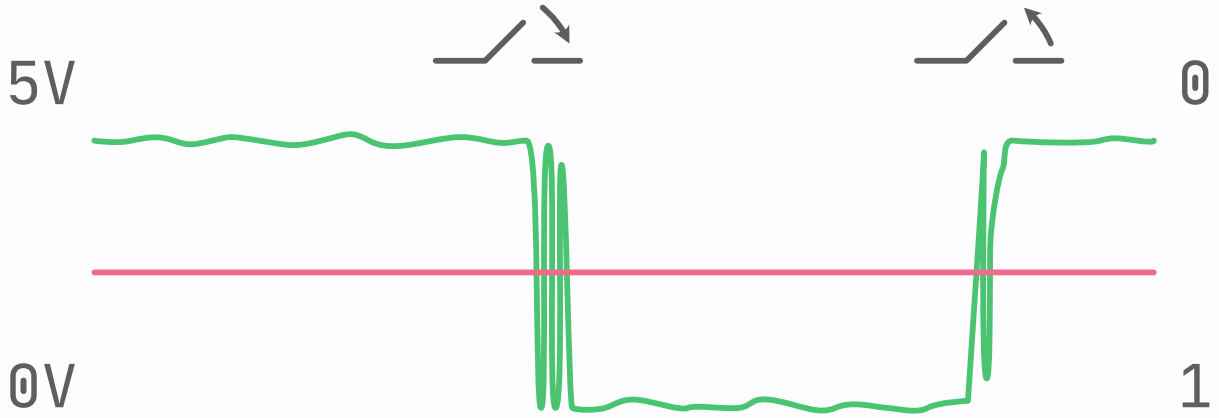
Solution

- Track button and LED state
- Detect if button changed state
  - Input is not equal button state
- If button turns `LOW` (press)
  - Switch LED state
- If button turns `HIGH` (release)
  - Store button state

Code



Bouncing

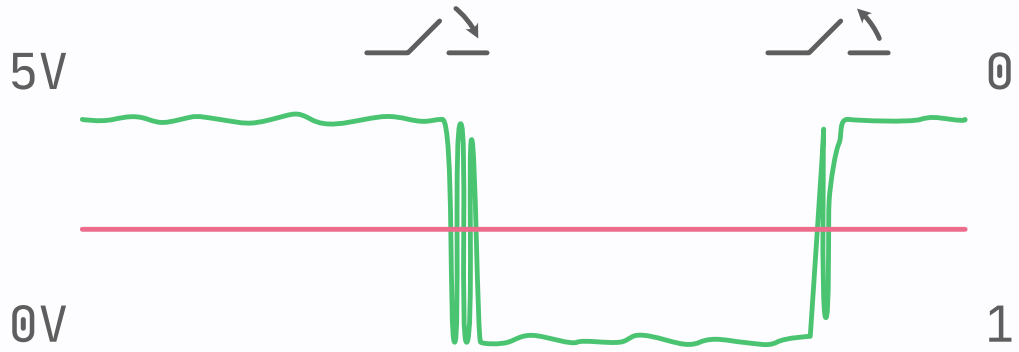


The challenging analog world!

Debouncing

- On button state switch
  - Store button state
  - Store current time
- After a predefined wait time
  - If button state steady
  - Recognize as trigger

Code



👩‍💻 Task 3: Intelligent Light Switch (20min)

Implement an intelligent light switch, which has two modes:

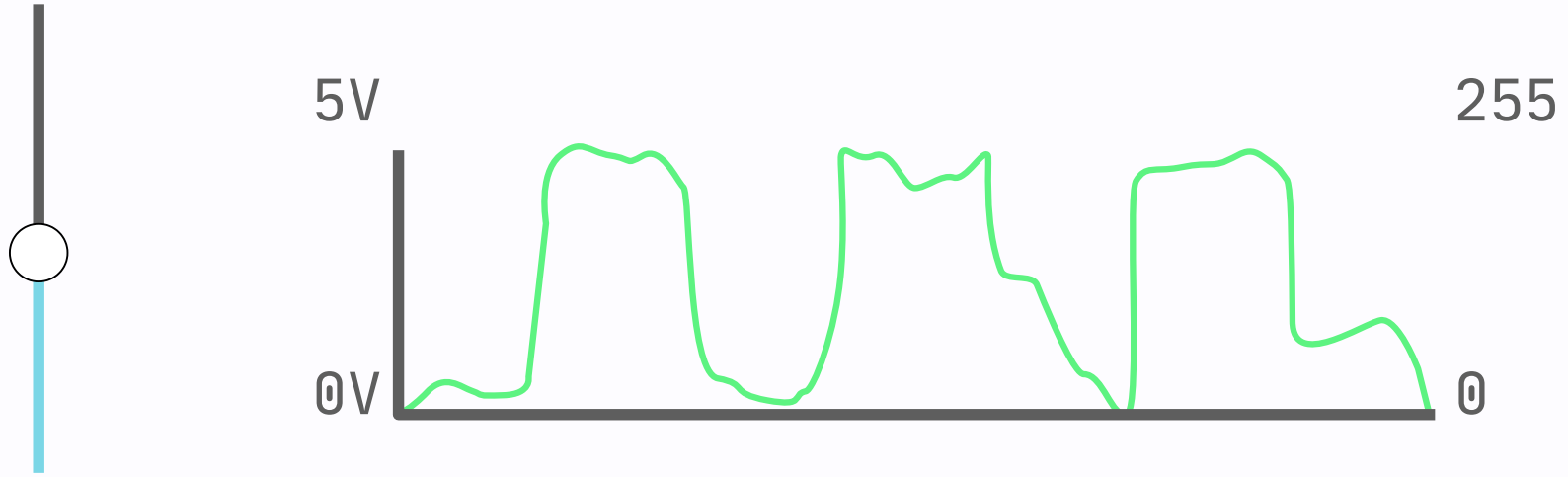
- Single Press: LED **turns on**
- Double Press: LED **starts blinking**
- If either mode is on, **turn off** on next button press.
- *Debounce* the button events!

🌻 Task 3: Intelligent Light Switch Solution

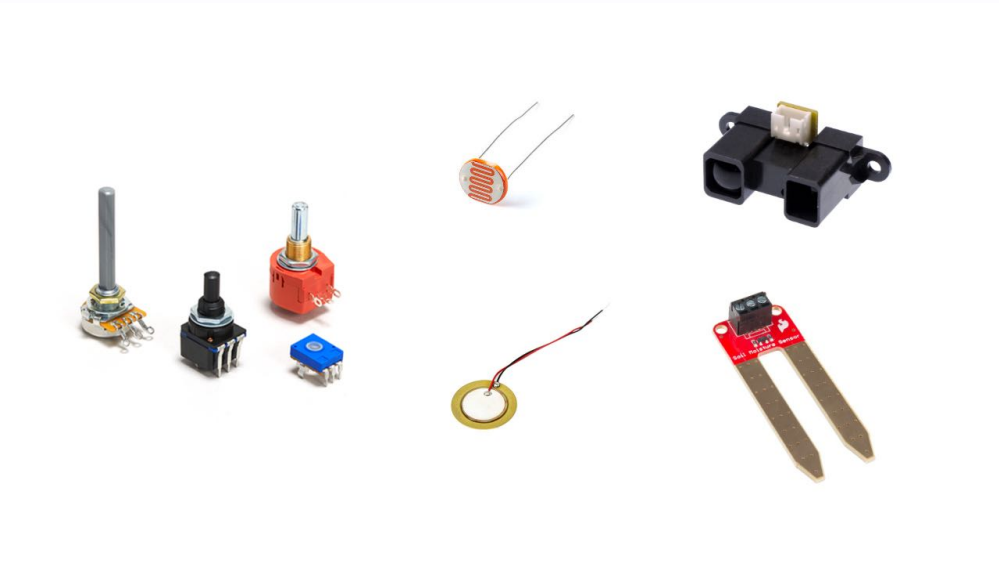
tbd

Source: Unrau

Analog Signal



Analog Sensors

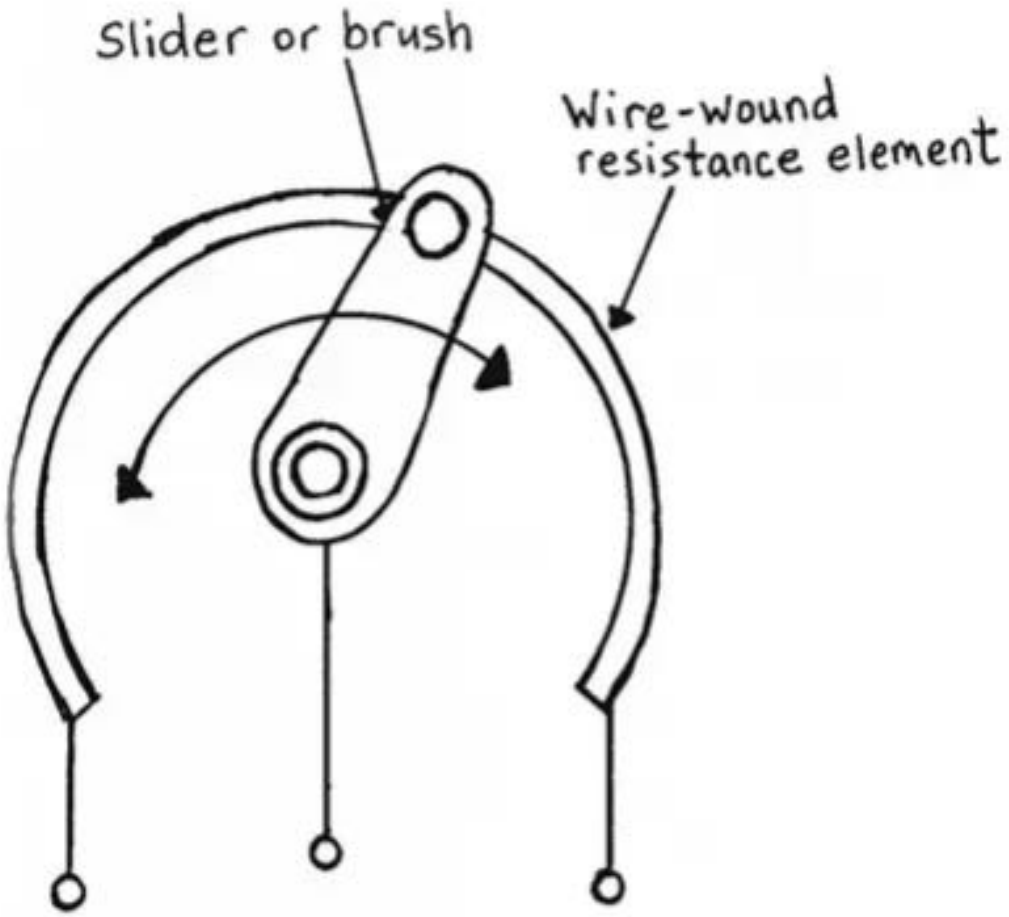


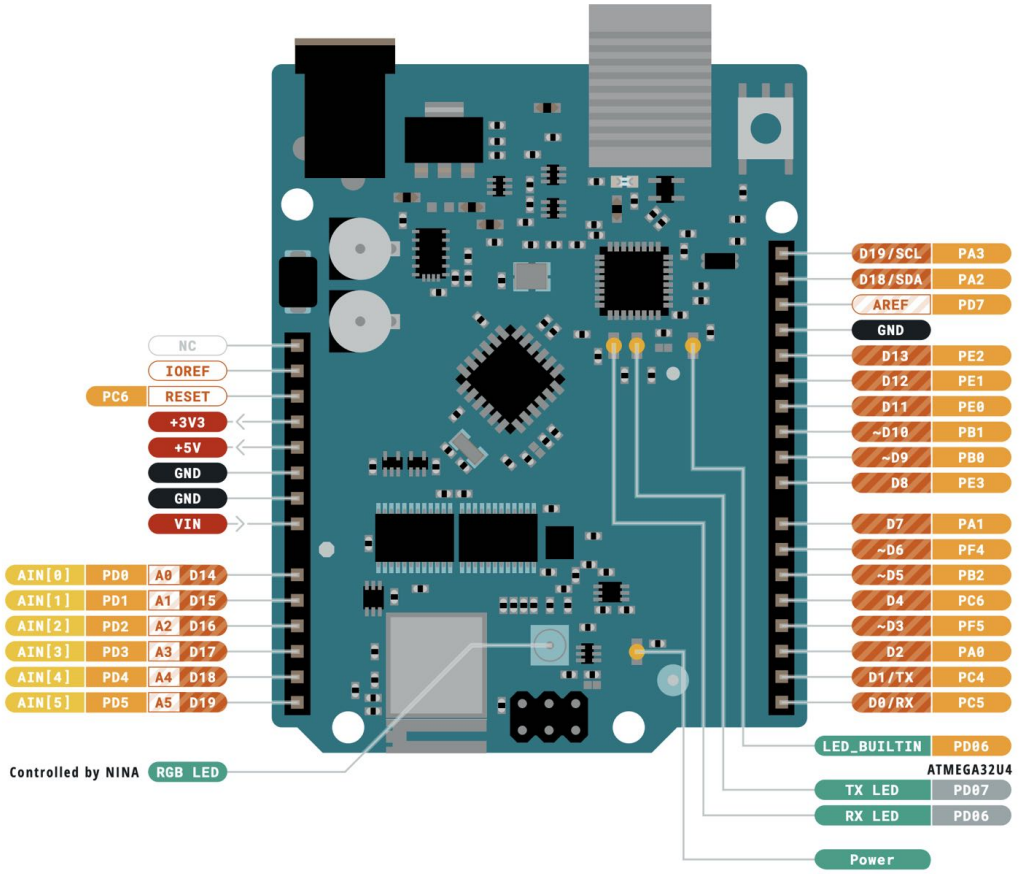
- Variable Resistors

Source: Jiroe

Potentiometer

- 10Ω - 5MΩ
- Datasheet





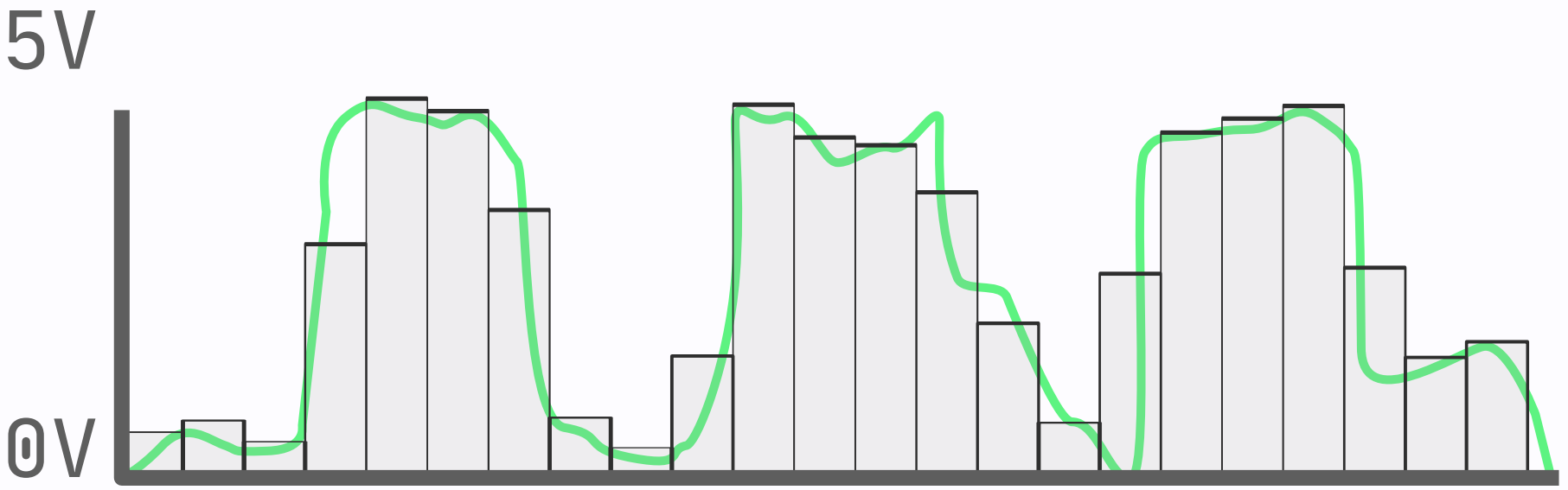
Analog Read

```
int analogRead(pin)
```

- Reads an analog value (discrete)
- 10bit ADC = `1024`
- Returns an `int` (number)
- `0-1023`

Source: Ardunio

Analog to Digital Converter ADC



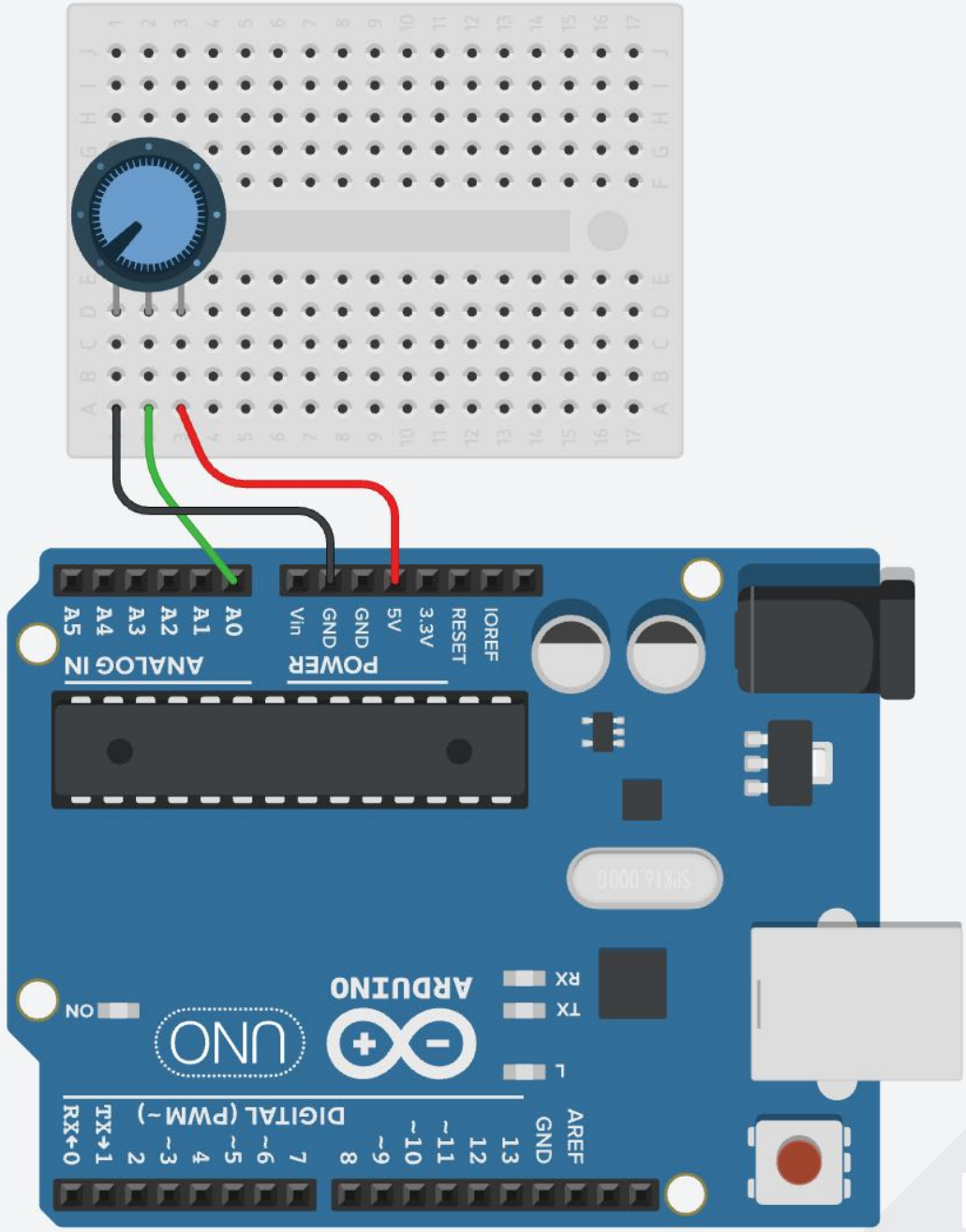
10bit resolution = `2^10` discrete values

👩‍💻 Task 4: Light Dimmer

(10min)

Read the value of the potentiometer and control the **brightness** of an LED with it.

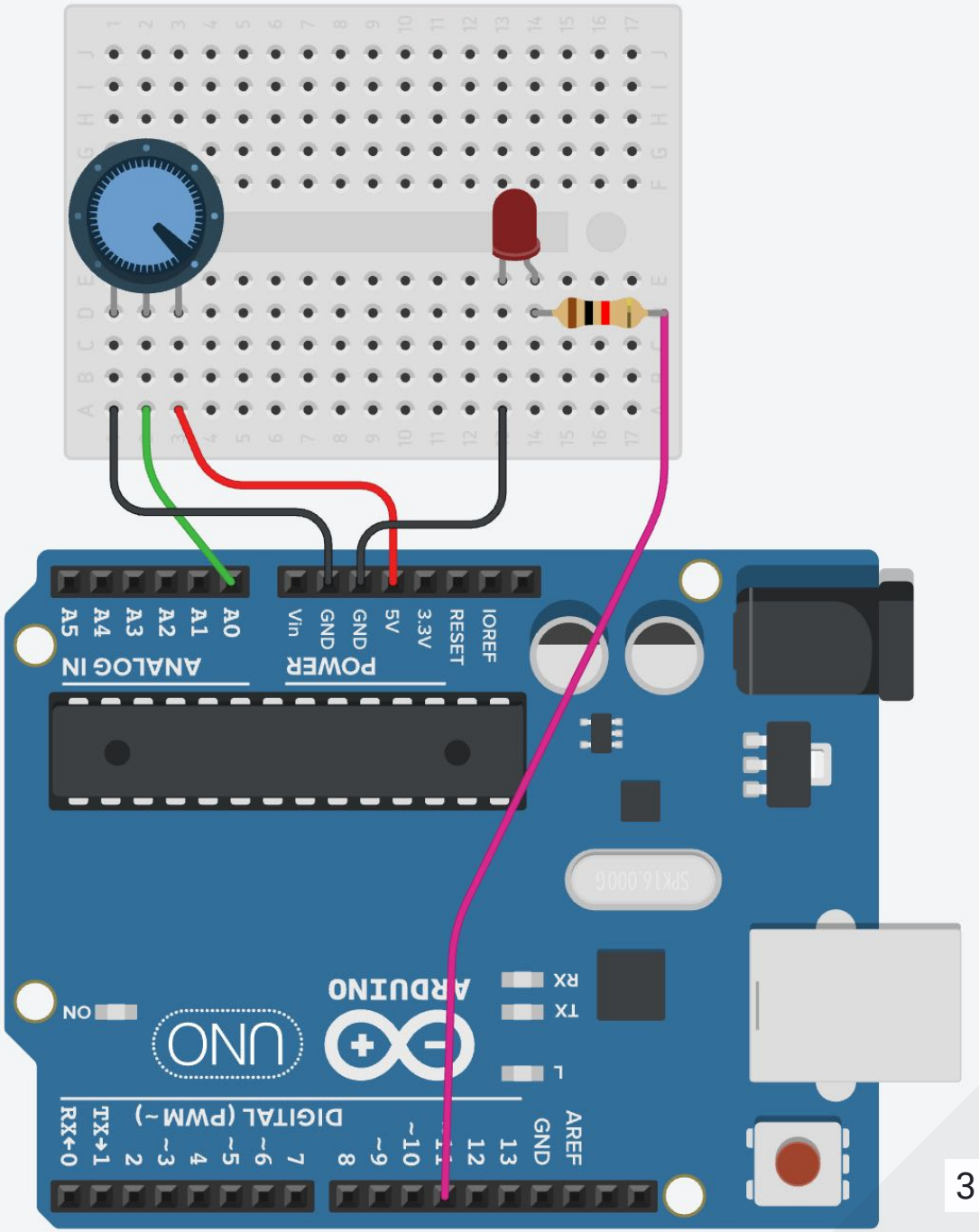
- `analogWrite(...)` for dimming
- Use `map(...)` function



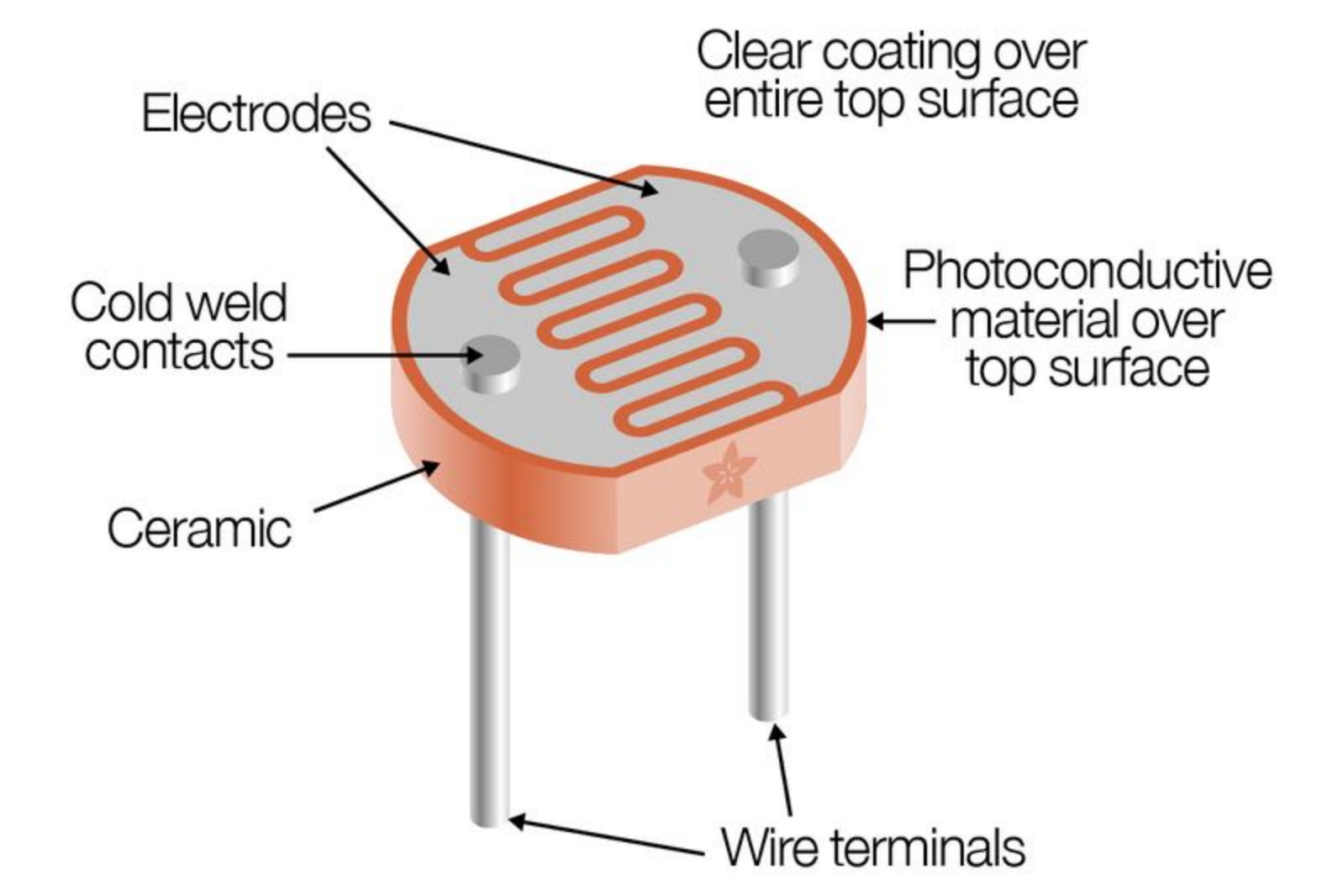
🌻 Task 4: Light Dimmer

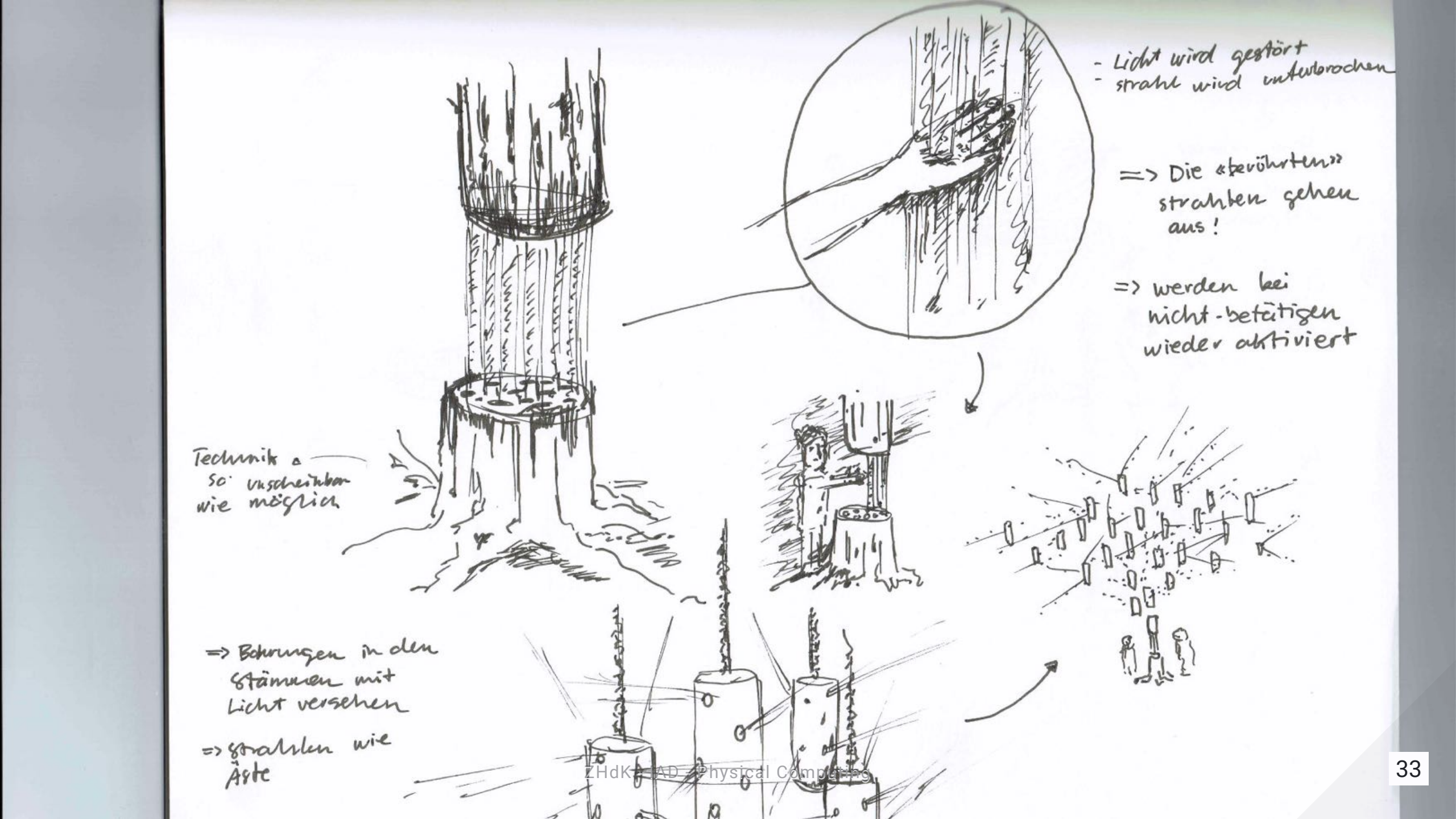
Solution

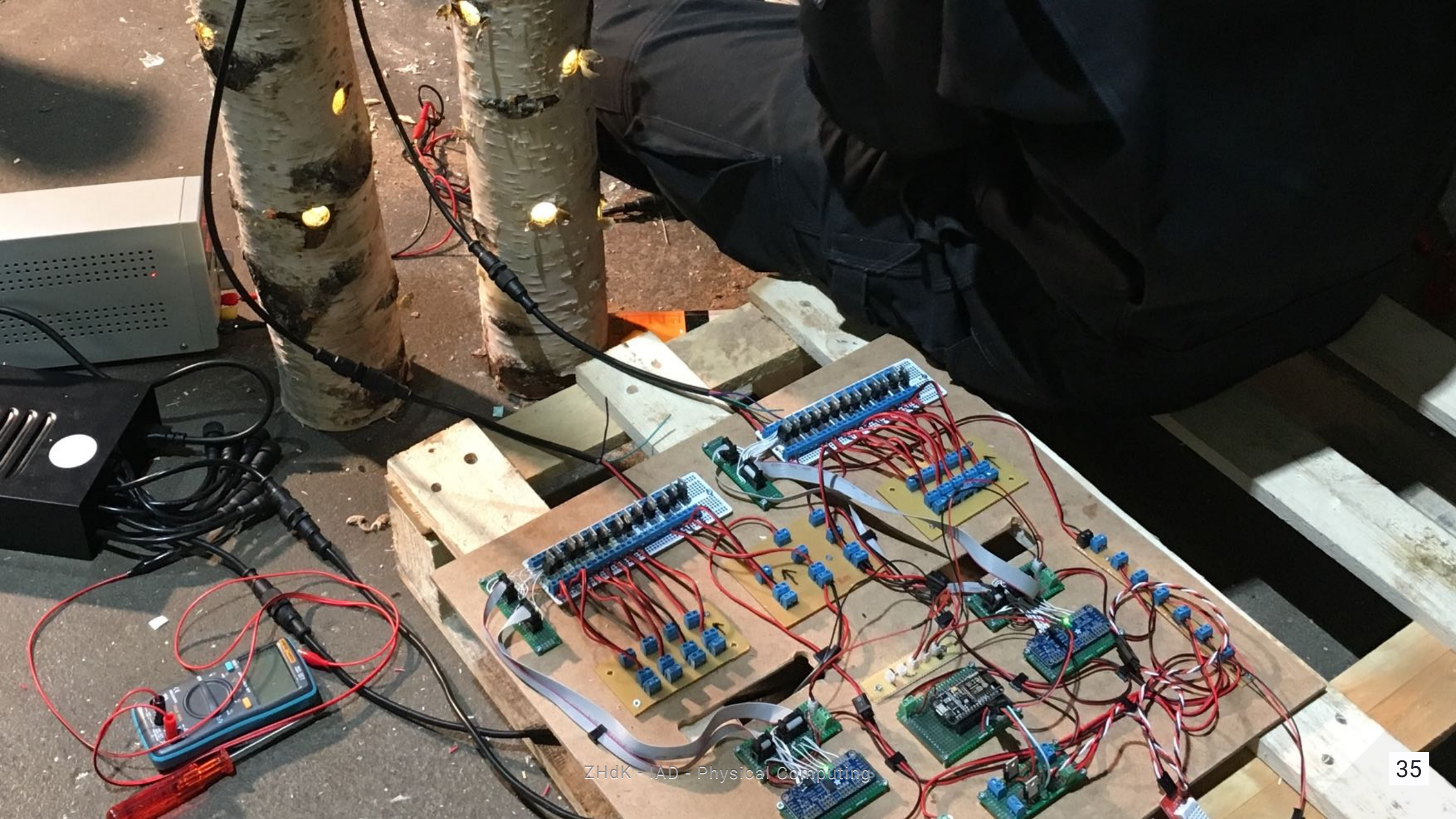
```
void setup() {
  pinMode(11, OUTPUT);
}

void loop() {
  int v = analogRead(0);
  int b = map(v, 0, 1024, 0, 255);
  analogWrite(11, b);
  delay(100);
}
```



Photoresistor

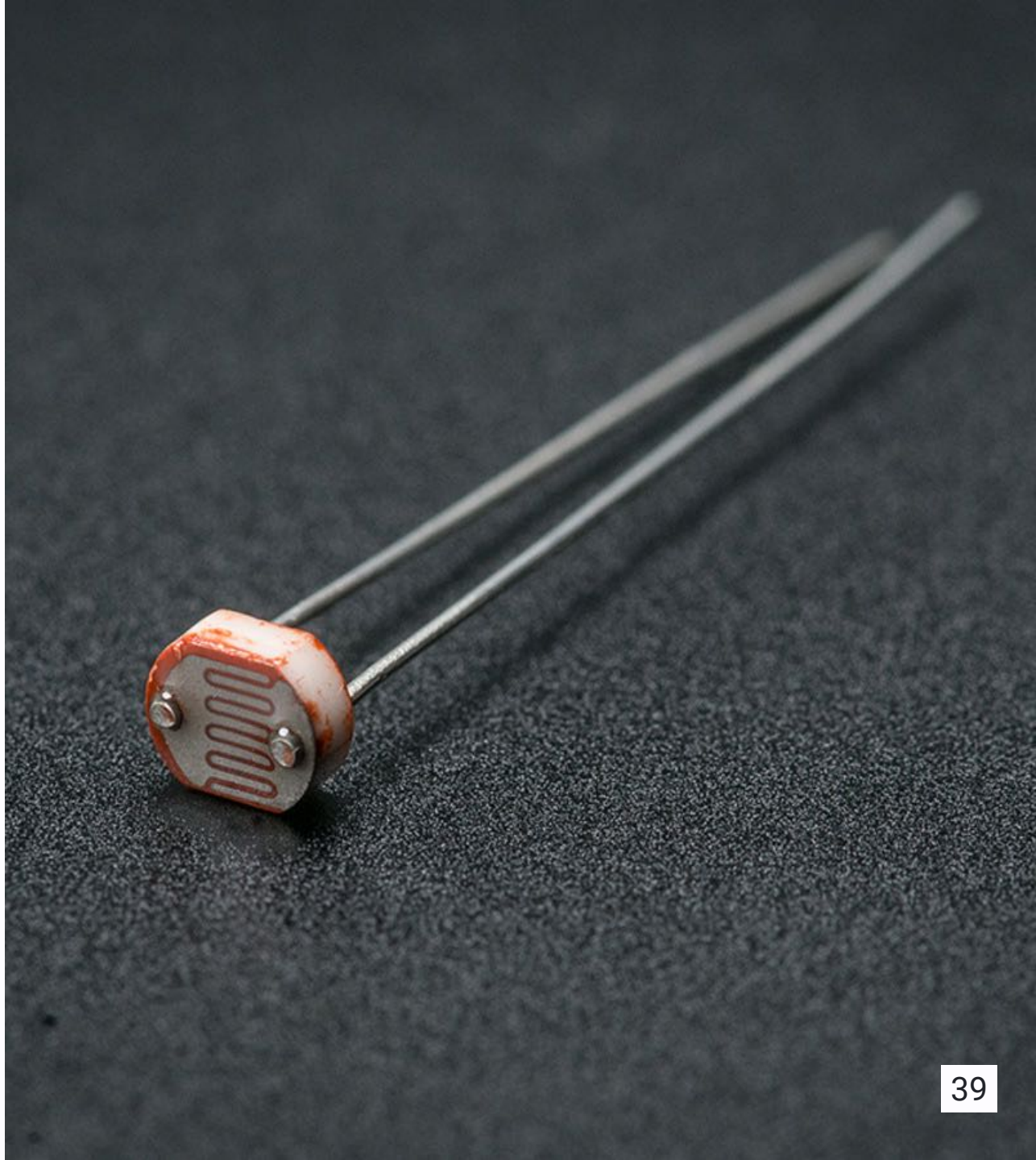



- Licht wird gestört
- strahl wird unterbrochen
=> Die «berührten» strahlen gehen aus!
=> werden bei nicht-betätigen wieder aktiviert
Technik so unscheinbar wie möglich
=> Bohrungen in den Stämmen mit Licht versehen
=> strahlen wie Äste

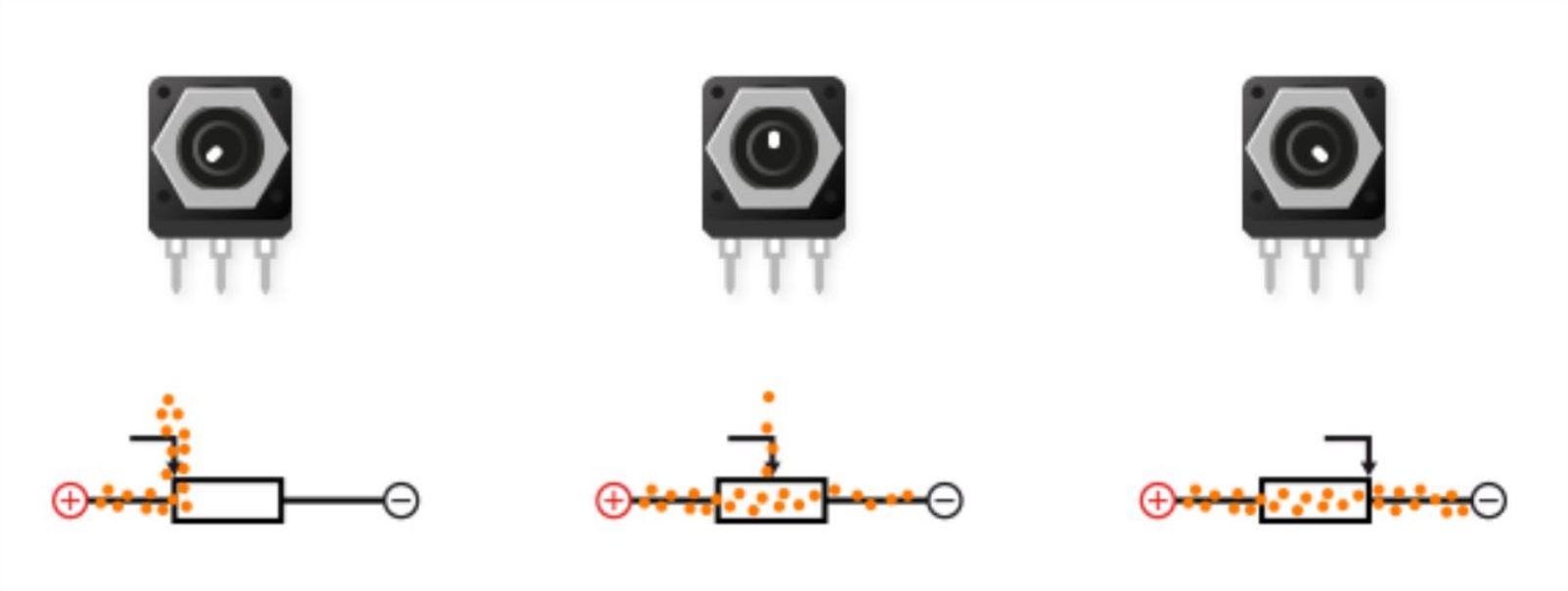


How to measure resistance `Ω`?

- Most sensors are **variable resistors** (`Ohm`)
  - Influence **current** (`A`)!
- Arduino can only measure **voltage** (`V`)
- Let's use a trick: **Voltage Divider**

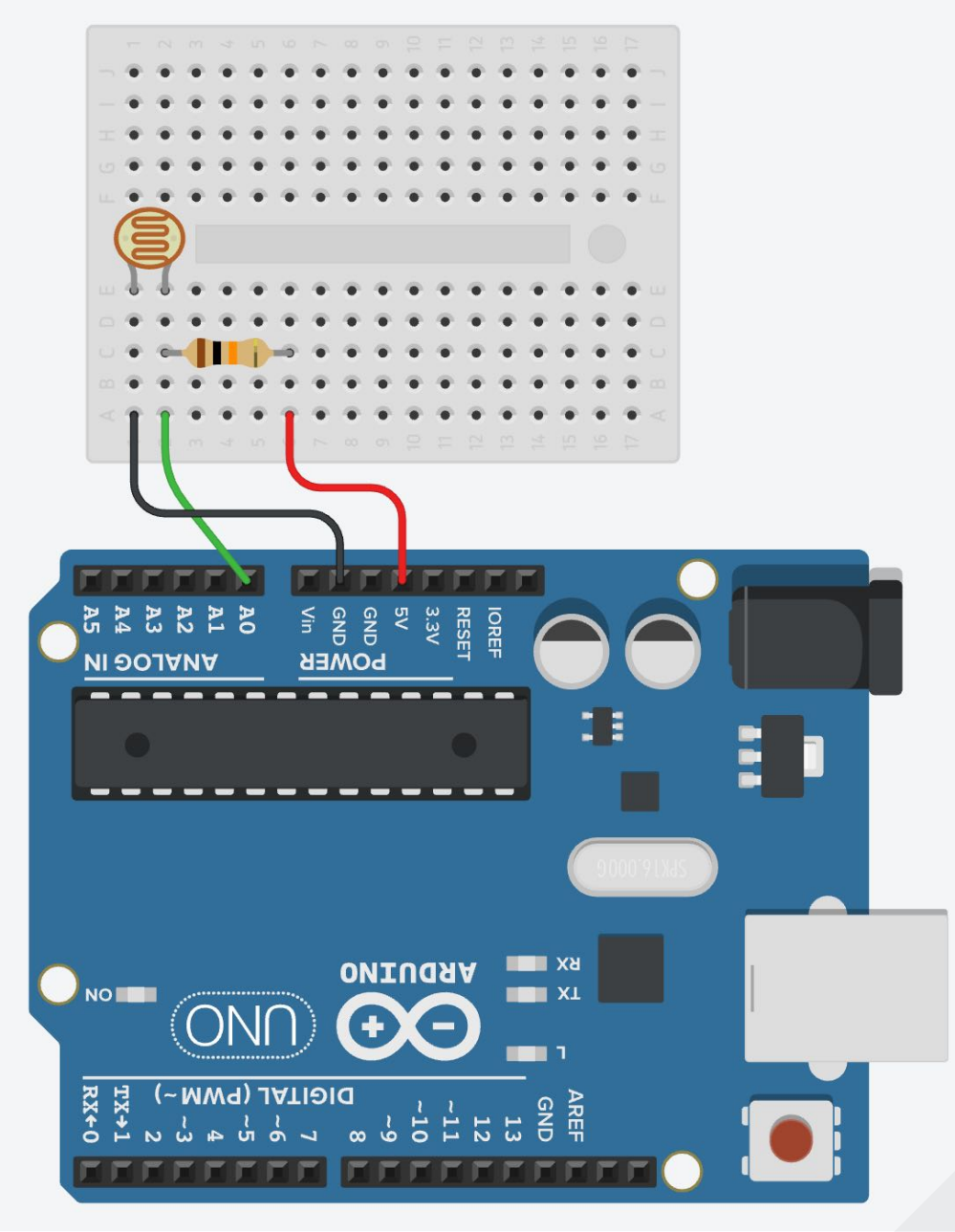


Voltage Divider



Photoresistor Circuit

- More light = more resistance
- More light = less voltage
- Less voltage = lower values
- 10 KOhm



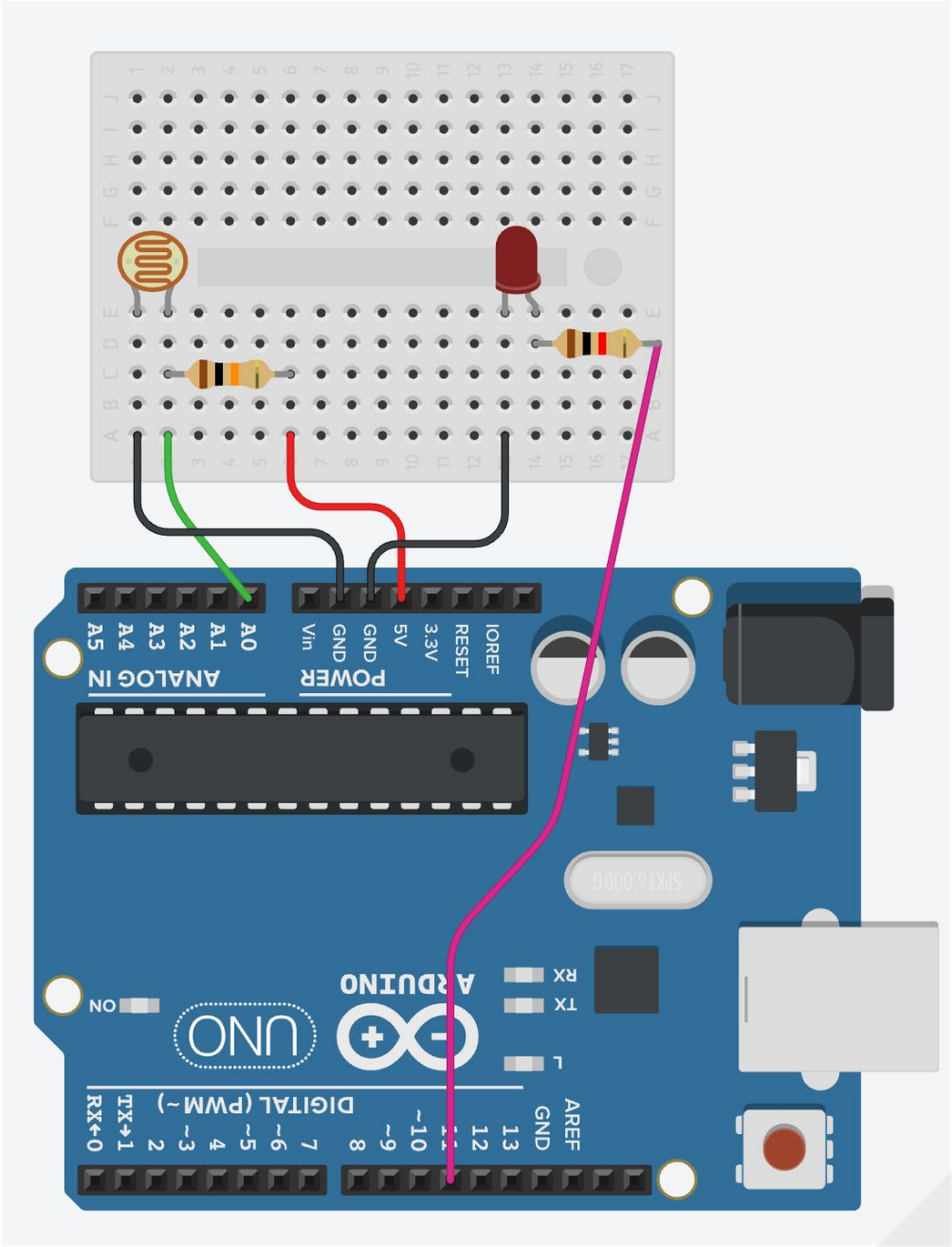
👩‍💻 Task 5: Reactive Light (10min)

Develop an adaptive light sensor which turns on the LED when the environment is too dark.

- Optional: Think about how to **fade out** and **fade in** the LED

Task 5: Reactive Light Solution

```
void setup() {
  pinMode(11, OUTPUT);
}

void loop() {
  int value = analogRead(0);
  int brightness = map(value, 0, 102
  analogWrite(11, brightness);
  delay(100);
}
```



Signal Fluctuation

The challenging analog world!

Moving Average

```
ma_new = alpha * new_sample + (1-alpha) * ma_old
```

```
float alpha = 0.01;
float ma = 1.0f;

void updateAverage(float value) {
  ma = alpha * value + (1-alpha) * ma;
}
```

👩‍💻 Task 6: Intelligent Light Dimmer (15min)

- Use **photoresistor** to adapt the LED brightness to the environment
- Use a **potentiomter** to set different LED modes
  - Left: Off
  - 1/4: Blink every 1s
  - 1/2: Blink every 0.5s
  - 3/4: Blink every 0.250s
  - Right: Light up LED all the time

Use a moving average for input smoothing!

🌻 Task 6: Intelligent Light Dimmer Solution

tbd

Questions?