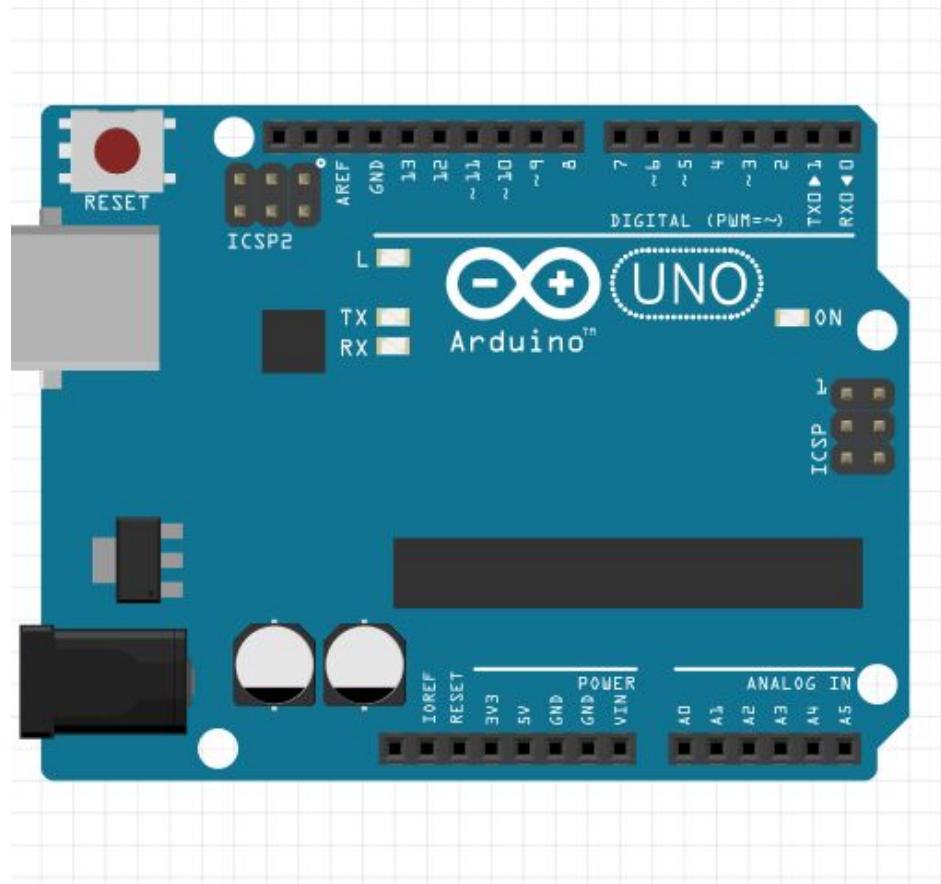


Arduino

Yarelis D. Acevedo
Arianna H. Colón
Tiahra N. Avilés



AGENDA



What is Arduino?

What is an Arduino? Why Arduino? How can I use it and implement it on the lab?



Arduino Software

How Arduinos are programmed



How to get started?



Hands On

Using Tinkercad simulator

What is a microcontroller

To answer this enter to menti.com
Code: 5829 2707

What is a microcontroller?

A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals.



Big computers vs small computers



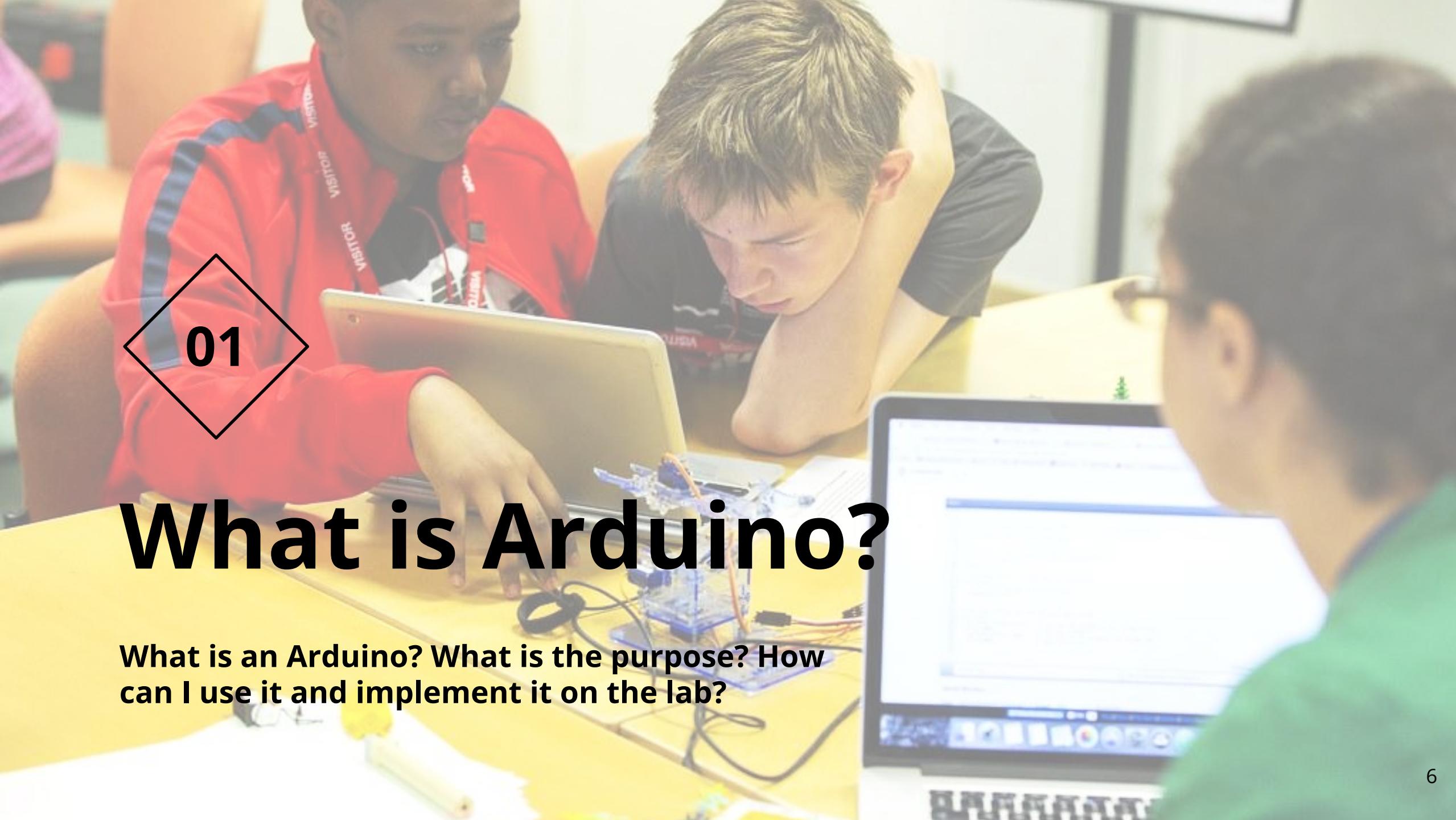
Dell Precision T1500

- **CPU Speed:** 2.93GHz
 - Quad-core!
- **RAM:** 16GB
- **Storage:** 2TB
- **I/O:**
 - USB, Firewire, Serial, PS/2, RJ-45, Audio, etc.



Atmel ATMEGA328P

- **CPU Speed:** 20MHz
 - Not Quad-core 😊
- **RAM:** 2KB
- **Storage:**
 - 32KB Program Memory
 - 1KB EEPROM
- **I/O:**
 - Up to 23 generic I/O
 - 6 of them 'analog-capable'
 - UART/SPI/I²C, etc.

A photograph showing three students in a classroom setting. One student in a red jacket labeled 'VISITOR' is looking at a laptop screen. Another student in a grey shirt is leaning over the laptop, focused on the screen. A third student's back is to the camera, looking at a tablet device. On the table in front of them is a blue breadboard with various electronic components and wires connected, representing an Arduino setup.

01

What is Arduino?

What is an Arduino? What is the purpose? How can I use it and implement it on the lab?

Arduino microcontroller

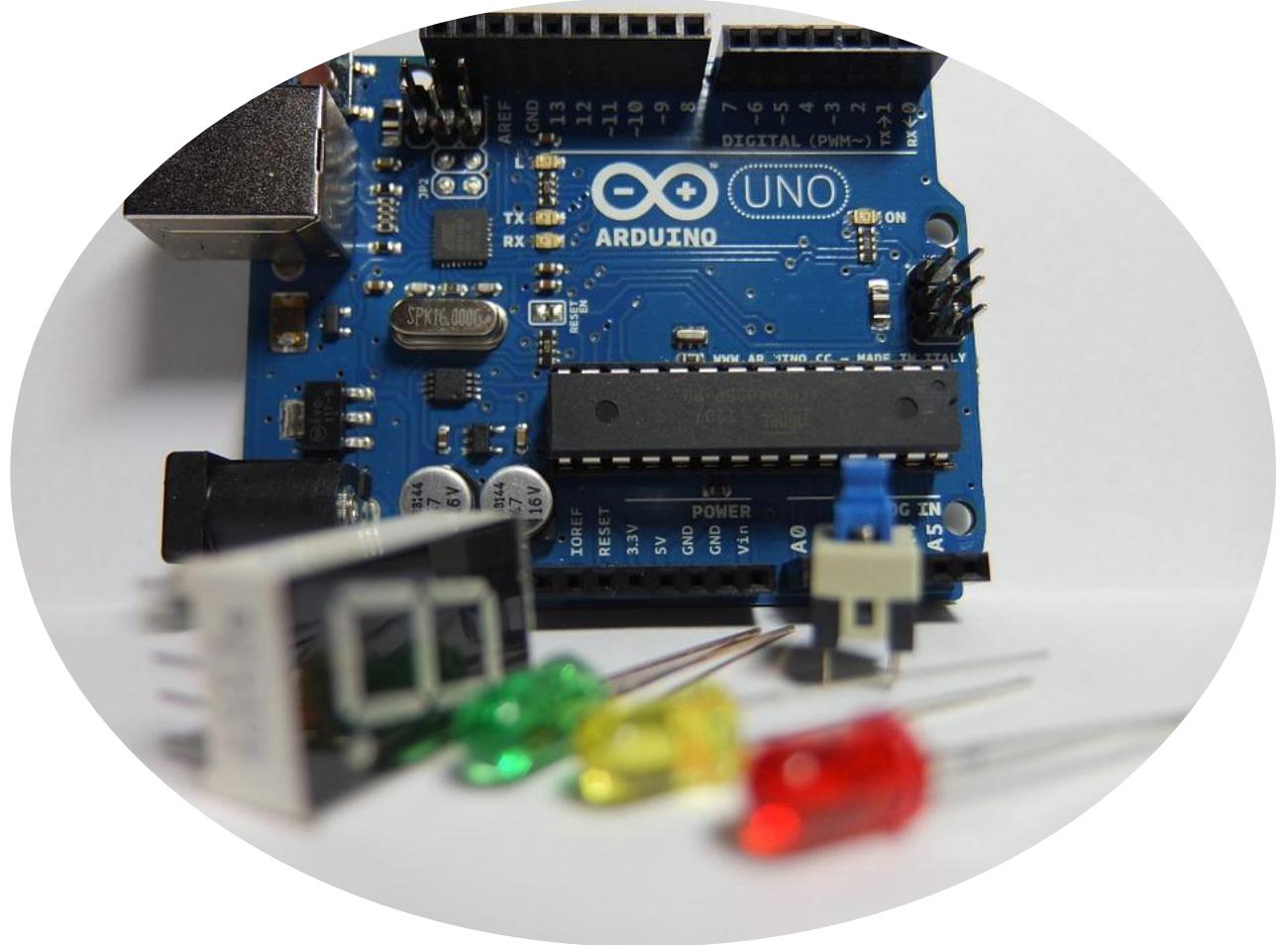


A brief story. The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators.

Arduino Microcontroller

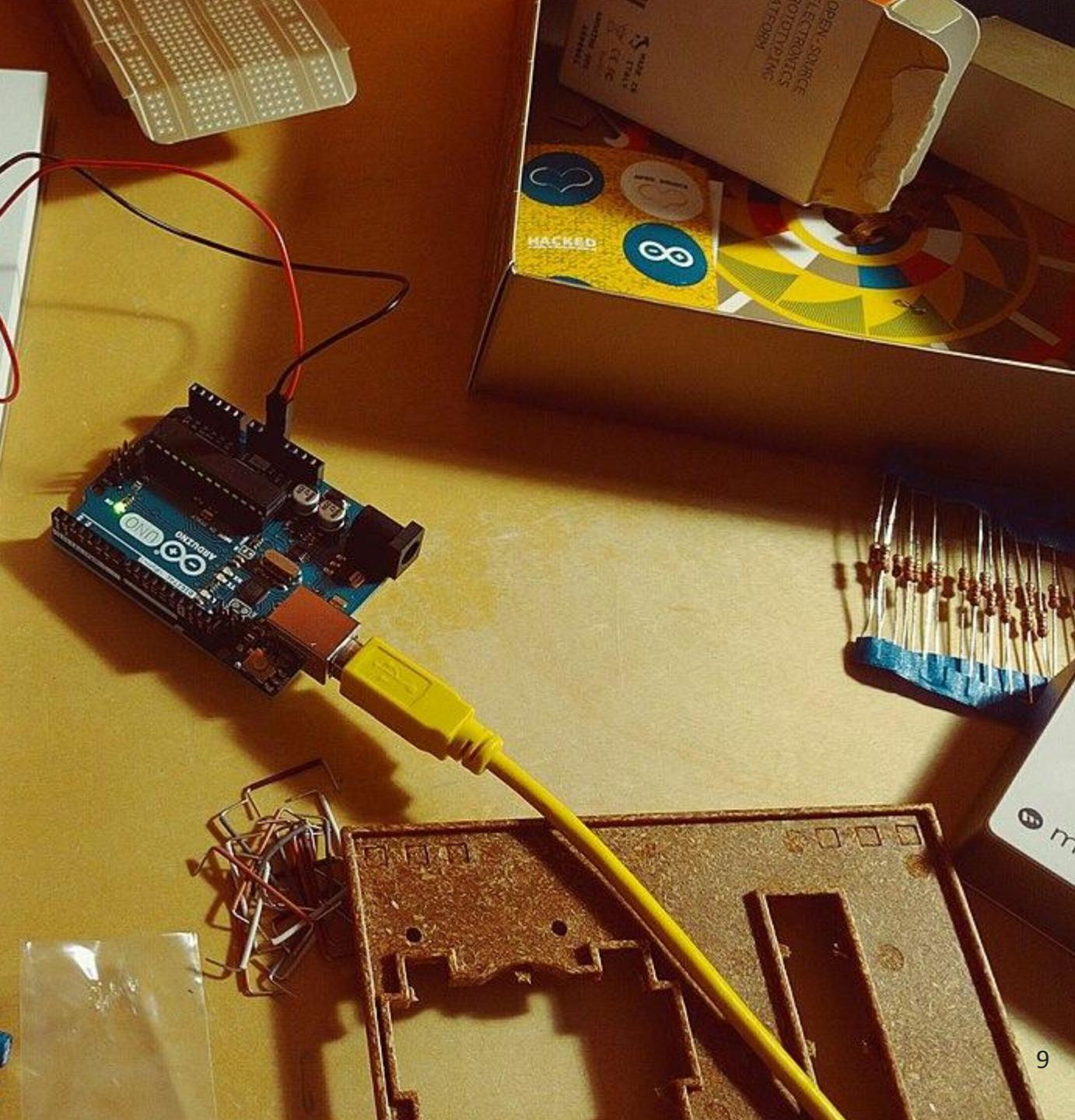
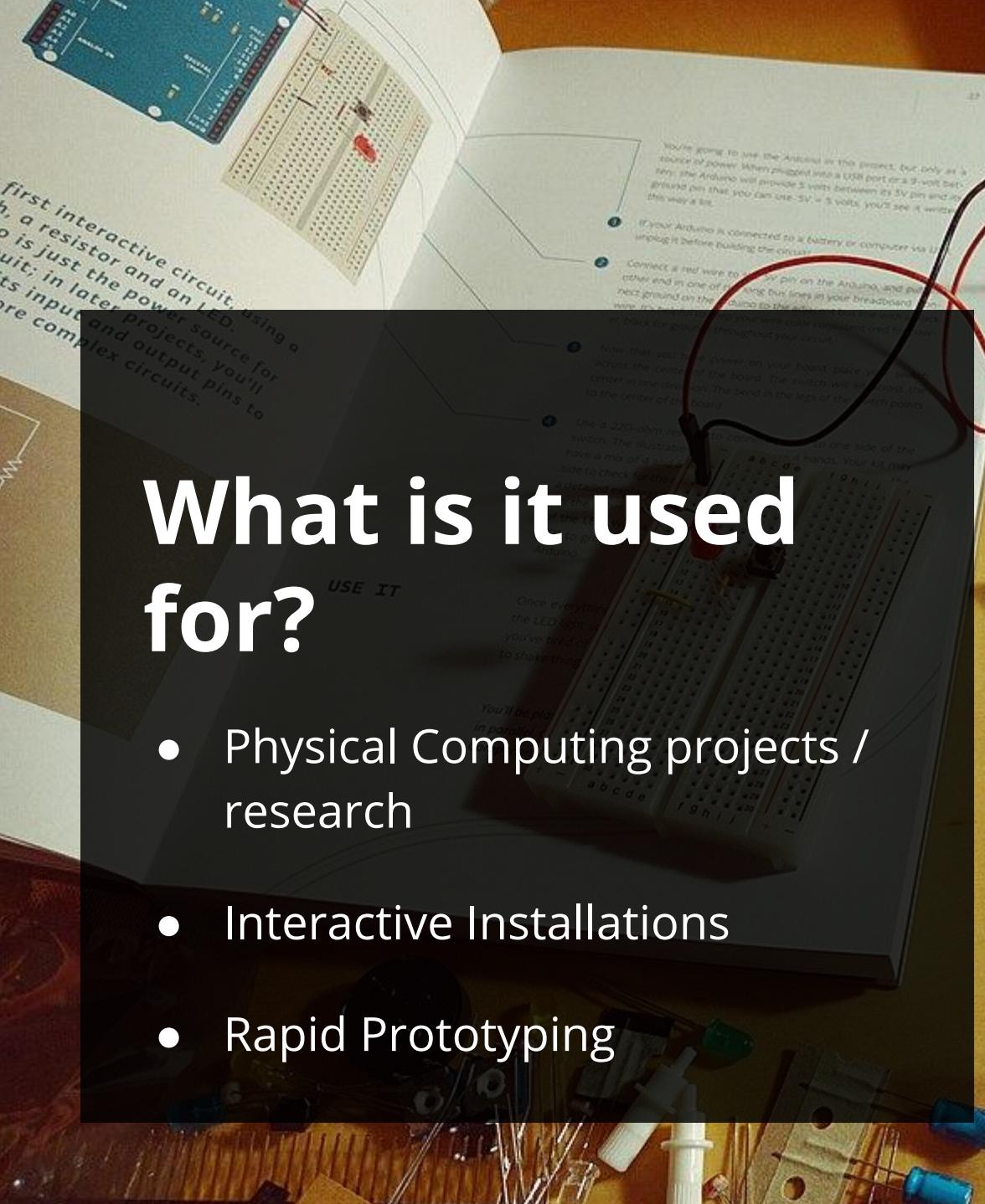


- Open-source electronics platform based on easy-to-use hardware and software.
- Are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.



What is it used for?

- Physical Computing projects / research
- Interactive Installations
- Rapid Prototyping



What can I do?

0



Sensors

- Push buttons, touchpads, tilt switches
- Variable resistors (Sliders, Volume knobs)
- Photoresistors (sensing light)
- Thermistors (temperature)
- Ultrasound (proximity range finder)



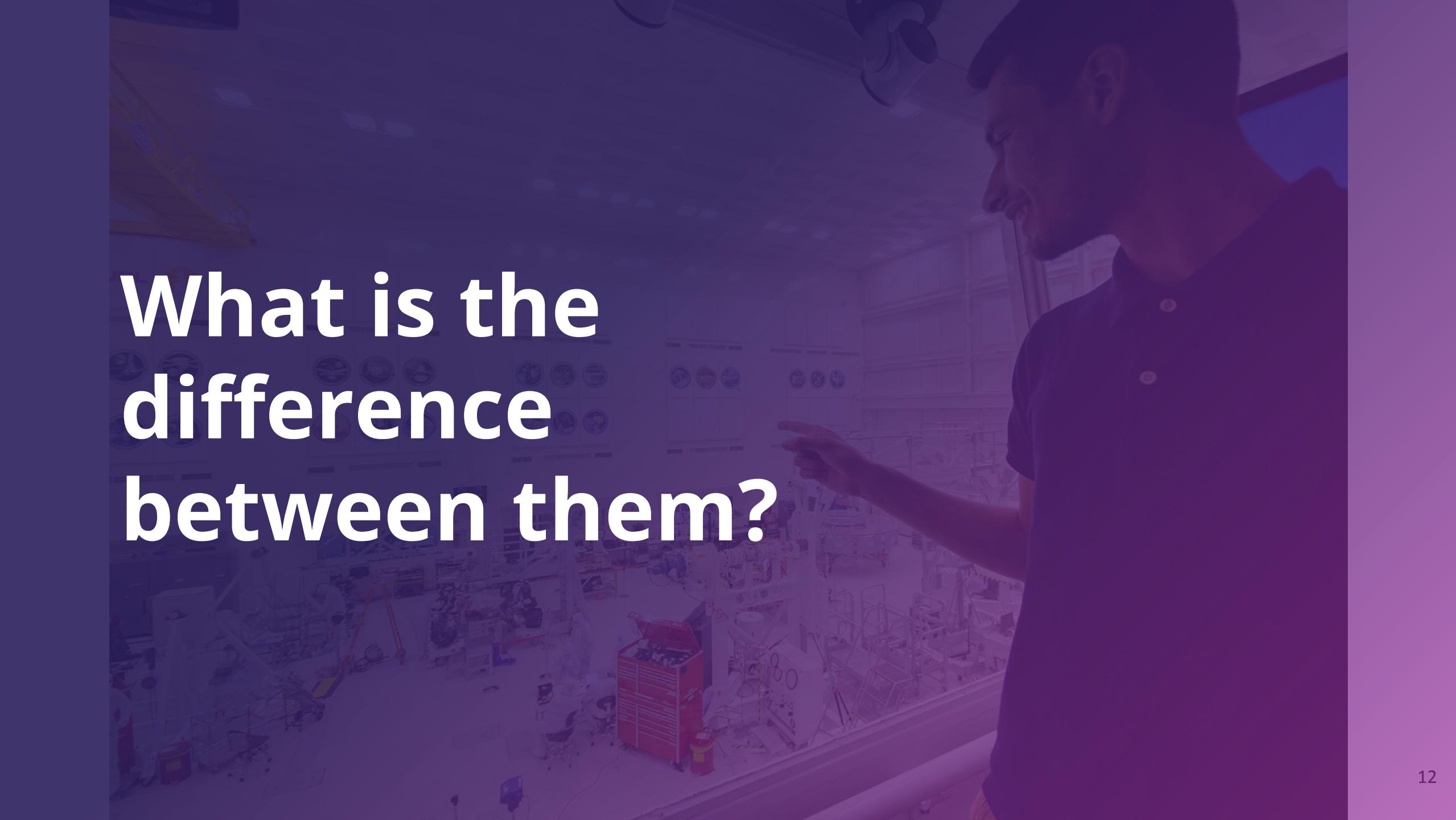
Actuators

- Lights, LED's
- Motors
- Speakers
- Displays (LCD's)



Types of Arduinos



A photograph of a man in a workshop or industrial setting. He is wearing a dark polo shirt and is looking down at a large, complex metal structure, possibly a piece of machinery or a frame. The background shows shelves with various tools and equipment, and the ceiling has industrial lighting fixtures.

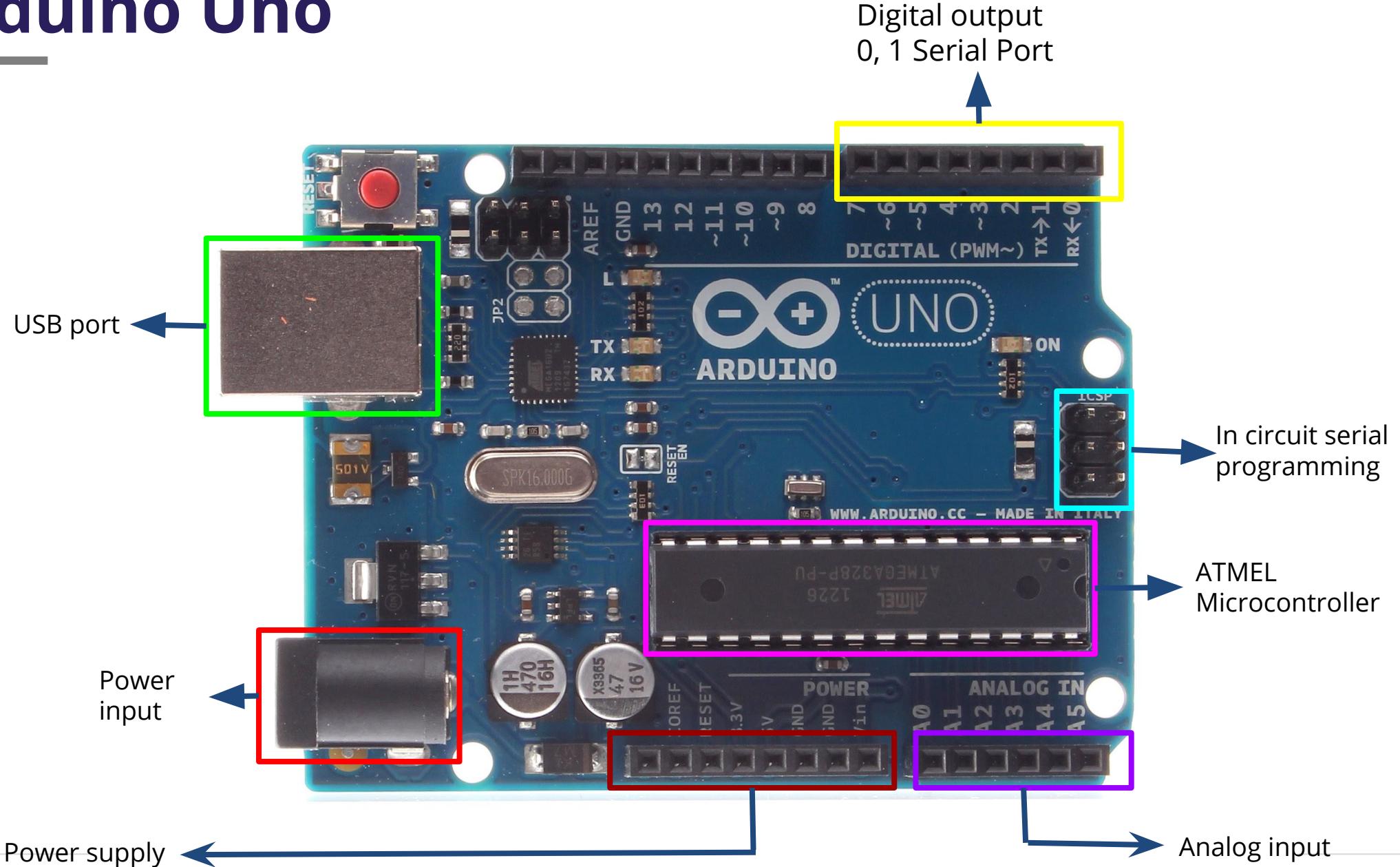
What is the
difference
between them?

Types of Arduino

Arduino Board	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KB SRAM, 512KB flash	54	12 input, 2 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	20	12 input, 0 output

Arduino Uno

14



Analog and digital pins

15

The Arduino can input and output analog signals as well as digital signals.

An analog signal is one that can take on any number of values, unlike a digital signal which has only two values: HIGH and LOW.



**We will talk
about this later!**

How Arduino is programmed?



Using a software
called Arduino IDE





This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/Blink>

```
/*
// the setup function runs once when you press reset or power the board
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
}
```

02

Arduino Software

How Arduinos are programmed

Arduino Software (IDE)

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Arduino Language

19

- Simplified C/C++
- Based on the wiring project
 - <http://wiring.org.co>
- Peripheral libraries
 - LCD, sensors, 12C, ect.



Useful functions

20

pinMode()	set pin as input or output
digitalWrite()	set a digital pin high/low
digitalRead()	read a digital pin's state
analogRead()	read an analog pin
analogWrite()	write an “analog” PWM value
delay()	wait an amount of time
millis()	get the current time

Sketch

21

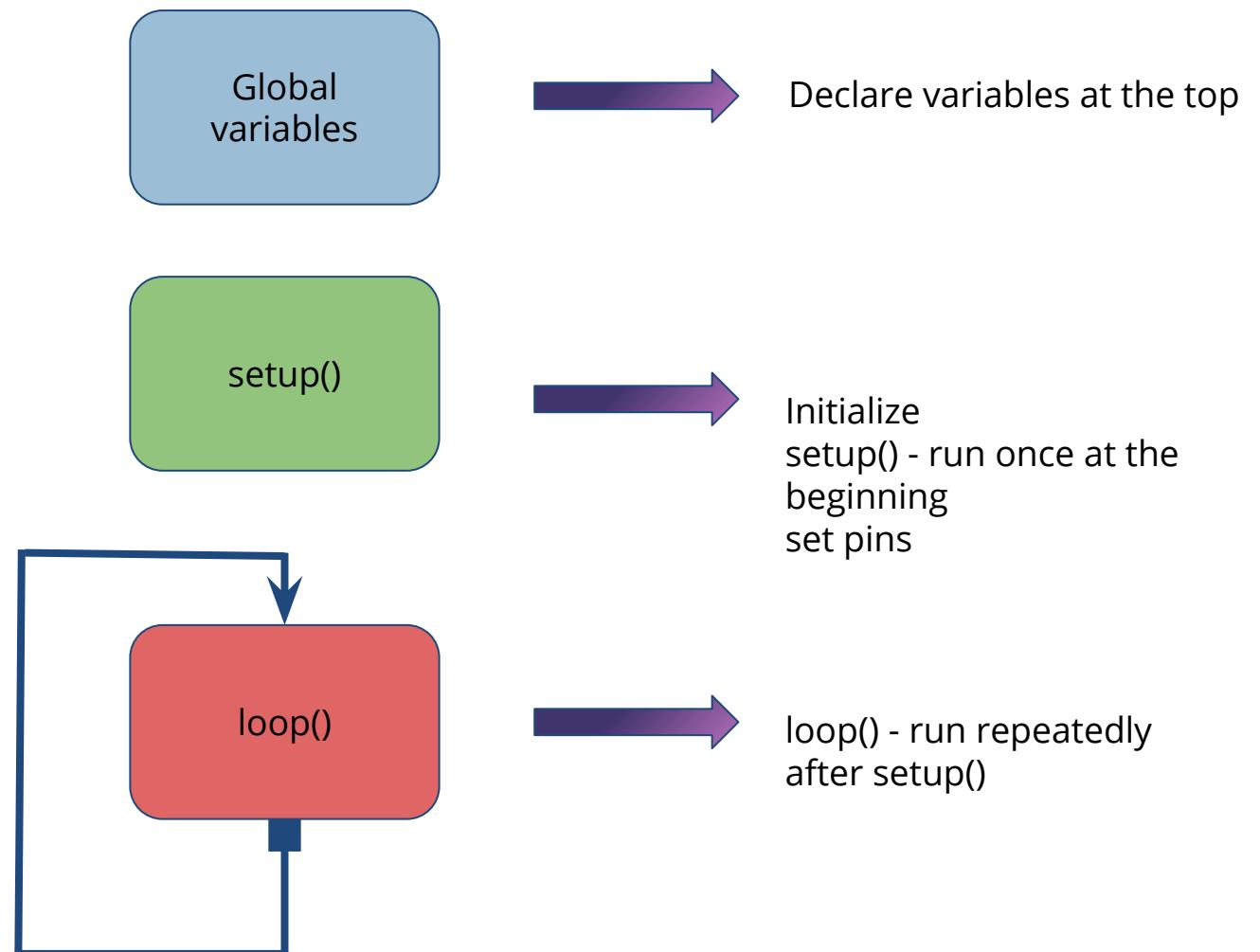
The screenshot shows the Arduino IDE interface. The title bar reads "sketch_may15a | Arduino 1.8.15". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and other functions. The main code area contains:

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
}
```

The status bar at the bottom right says "Arduino Uno".

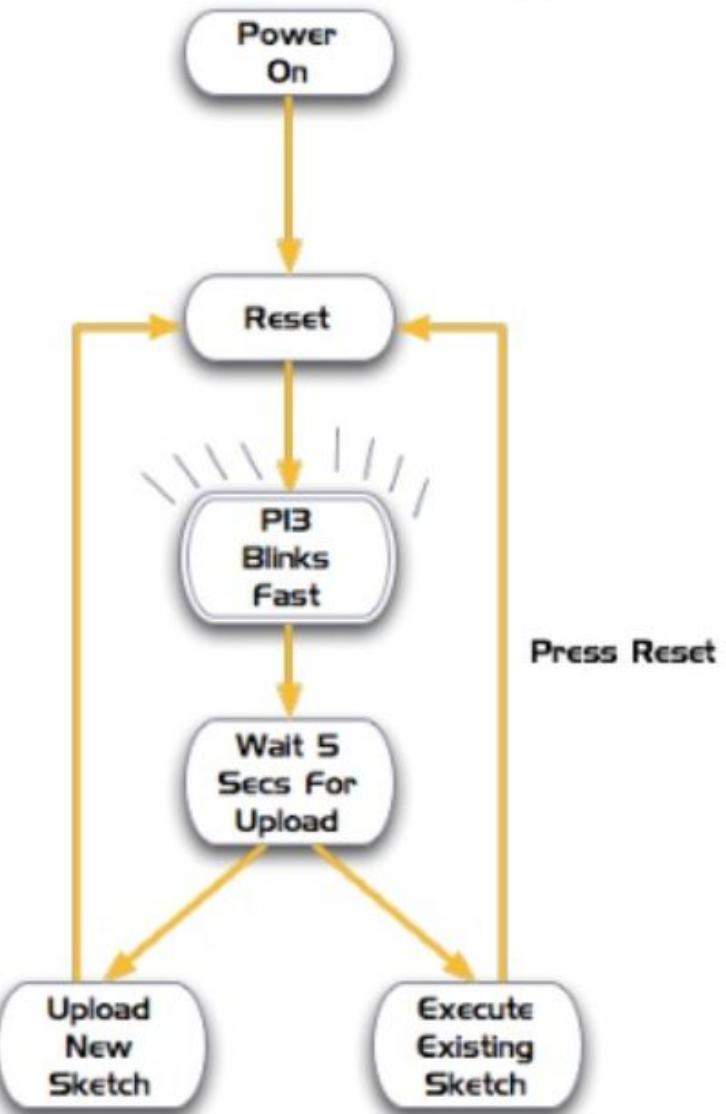


Let's see how a sketch for turning an LED light runs in an Arduino

```
void setup() {  
    // initialize digital pin 13 as an output.  
    // the digital pin 13 has an LED connected on most Arduino  
    // boards.  
    // set the digital pin 13 as an output.  
    pinMode(13, OUTPUT);  
  
    // turn the LED on (HIGH is the voltage level)  
    digitalWrite(13, HIGH); // set the LED on  
  
    // wait for a second  
    delay(2000); // set the LED off  
    // wait for a second  
    delay(2000);  
}
```

Blinking LED

23



The screenshot shows the Arduino IDE interface with the 'Blink' sketch open. The code is as follows:

```
int ledpin = 13;

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(5000);                 // wait for 5 seconds
  digitalWrite(ledpin, LOW);    // turn the LED off by making the voltage LOW
  delay(5000);                 // wait for 5 seconds
}
```

At the bottom of the IDE, the status bar displays:

- Done compiling.
- Sketch uses 936 bytes (2%) of program storage space. Maximum is 32256 bytes.
- Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

Global variables

Global variables



```
int ledPin = 13;  
- LED connected to  
the control pin 13
```

Setup()

25



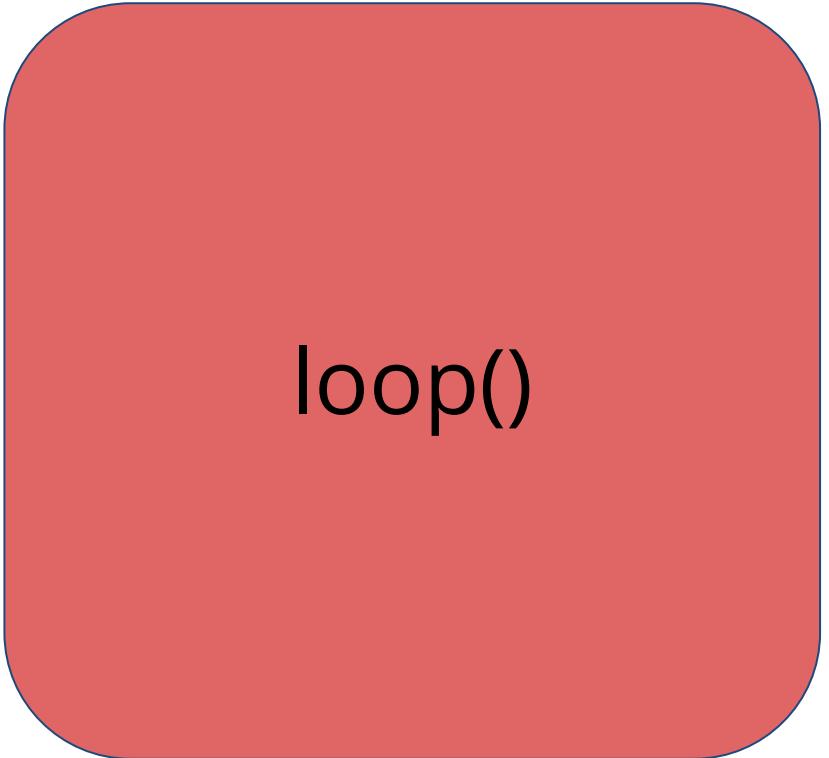
setup()



pinMode(ledPin, OUTPUT);

loop()

26



```
digitalWrite(ledPin, HIGH);  
delay(5000);  
digitalWrite(ledPin, LOW);  
delay(5000);
```

Verify**Upload****New
sketch****Opens****Save****Search**

```
File Edit Sketch Tools Help  
sketch_may15a  
void setup () {  
    // put your setup code here, to run once:  
}  
  
void loop () {  
    // put your main code here, to run repeatedly:  
}
```

03

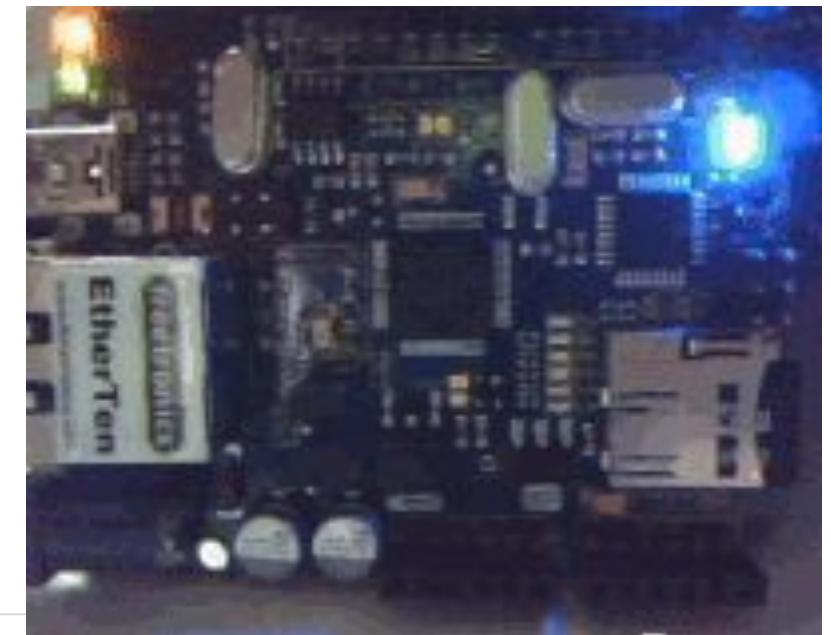
How to get started?

A black and white photograph of a classroom interior. In the foreground, there are several rows of wooden desks and chairs. The room has a chalkboard on the right wall featuring a complex chemical reaction diagram with various structures and arrows. To the left of the chalkboard is a large window with a decorative mural of a landscape scene. The overall atmosphere is that of a traditional classroom setting.

If you have the
board

How to get started?

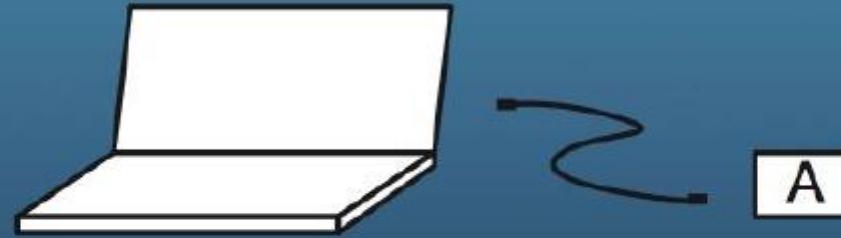
- Arduino board
 - USB cable
 - DC power supplies
- Download the Arduino's software (Arduino IDE)
 - Read carefully
 - Instructions to install and setup the Arduino board with the computer and software
 - Download the Arduino IDE software
 - <http://www.arduino.cc>
- Plug it in!



Plug in it into the computer

Writing and Downloading Code

Write sketch on PC



Download sketch to Arduino



Online

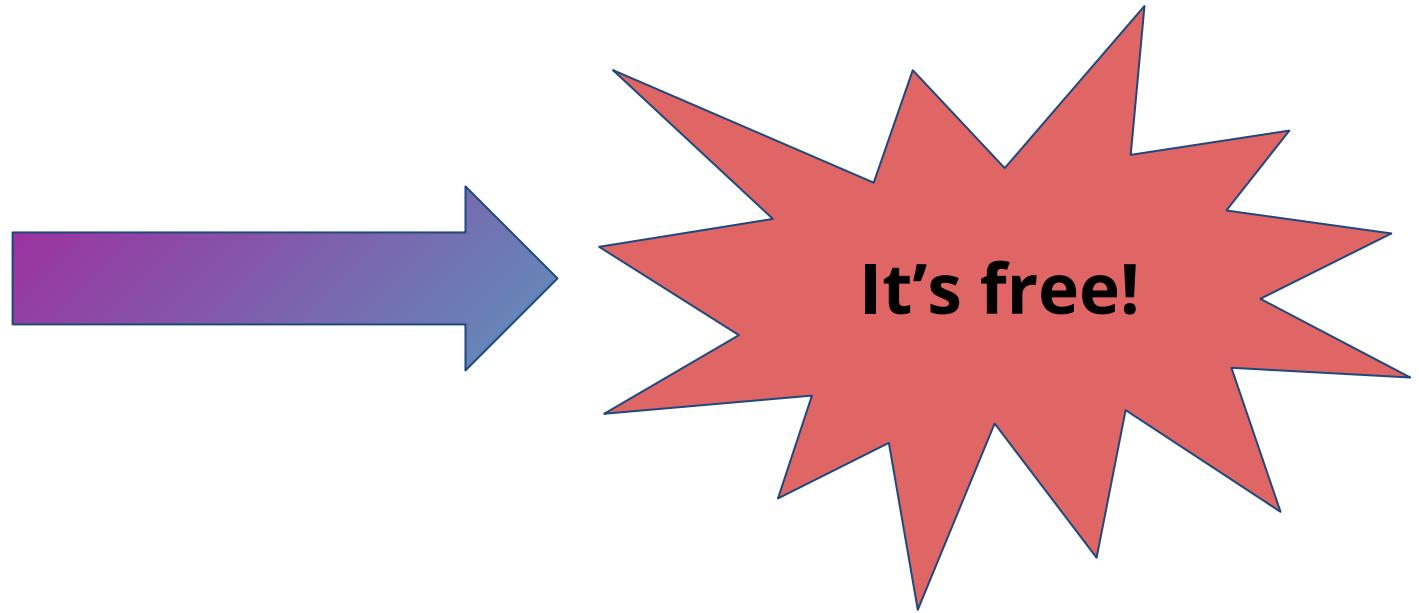


Tinkercad is a free, online 3D modeling program that runs in a web browser, known for its simplicity and ease of use. Since it became available in 2011 it has become a popular platform for creating models for 3D printing as well as an entry-level introduction to constructive solid geometry in schools.



AUTODESK®
TINKERCAD®

Make an
account in
tinkercad



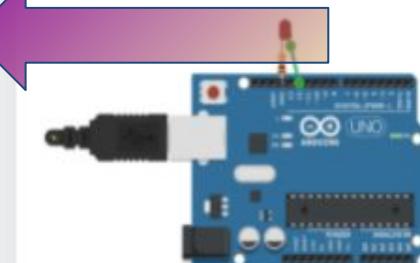
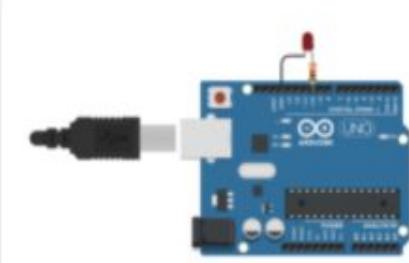
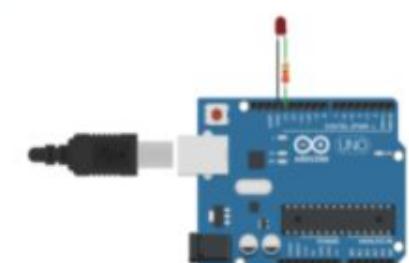
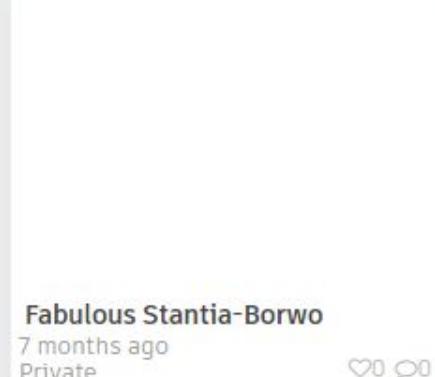
AUTODESK®
TINKERCAD®[Gallery](#)[Blog](#)[Learn](#)[Teach](#)

yarelis.acevedo

 Search designs...[3D Designs](#)[Circuits](#)[Codeblocks](#)NEW[Lessons](#)[Your Classes](#)[Projects](#)[+ Create project](#)[Tinkercad Lesson Plans](#)

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

Circuits

[Create new Circuit](#)[Incredible Snicket](#)9 minutes ago
Private [Copy of Arduino Blink](#)11 minutes ago
Private [Copy of Blink \(Blocks\)](#)2 months ago
Private [Fabulous Stantia-Borwo](#)7 months ago
Private  Select



Funky Amur

All changes saved



Code

Start Simulation

Export

Share



Components
Basic

Search



Resistor



LED



Pushbutton



Potentiometer

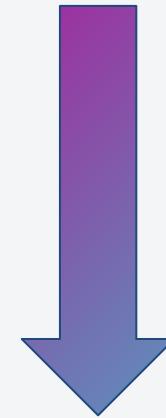


Capacitor



Slideswitch

Scroll down
until you find
the Arduino
Uno



tinkercad.com/things/et4jxNfHWr4-funky-amur/edit#circuits

Funky Amur

All changes saved

Code Start Simulation Export Share

Components Basic Search

Arduino Uno R3

Name 1

Find the resistor and the LED

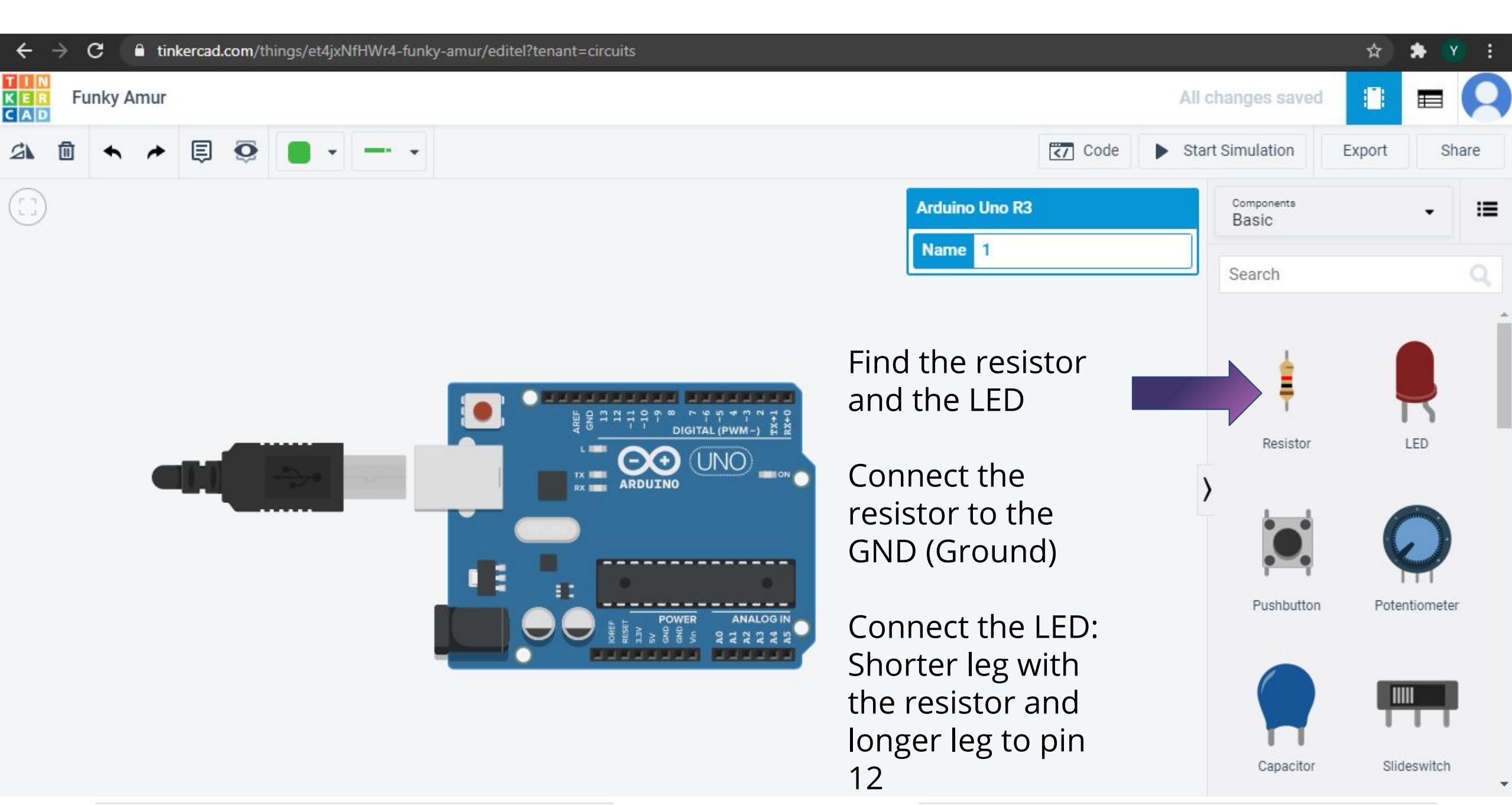
Connect the resistor to the GND (Ground)

Connect the LED: Shorter leg with the resistor and longer leg to pin 12

Resistor LED

Pushbutton Potentiometer

Capacitor Slideswitch



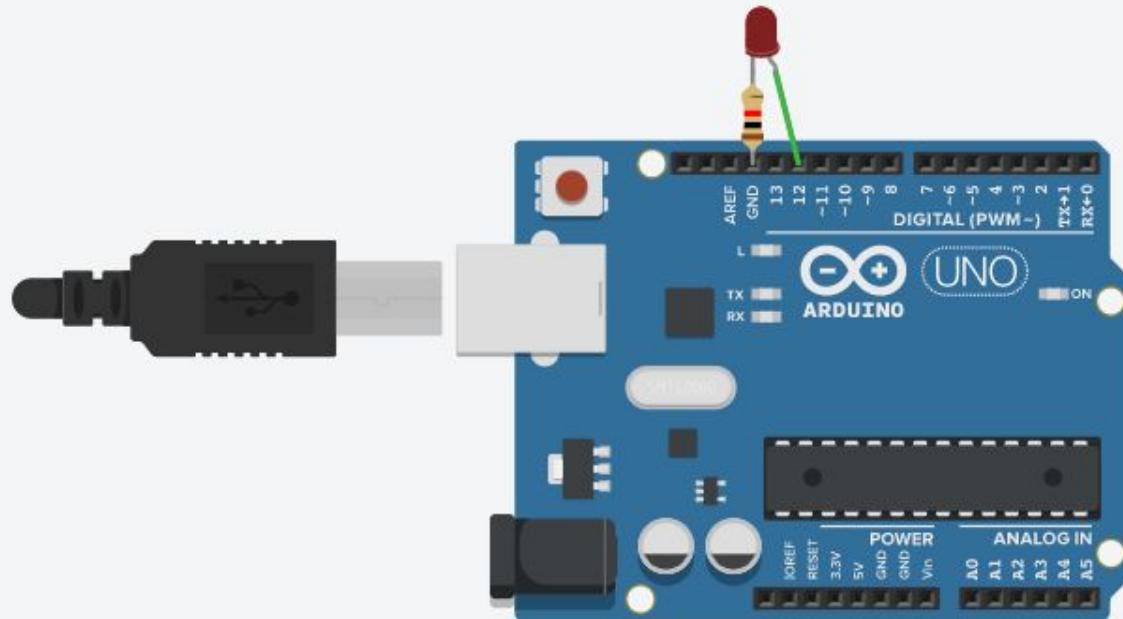


Code

Start Simulation

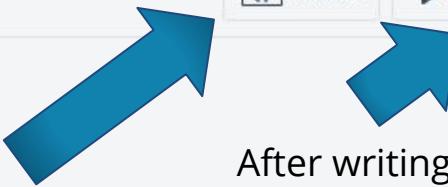
Export

Share



Write the code

After writing the code, start the simulation here to see the circuit work with the Arduino!



Resistor



LED



Pushbutton



Potentiometer



Capacitor



Slideswitch

tinkercad.com/things/et4jxNfHWr4-funky-amur/editel?tenant=circuits

Funky Amur

All changes saved

Code Start Simulation Export Share

Text

1 // C++ code
2 //
3 void setup()
4 {
5 pinMode(13, OUTPUT);
6 }
7
8 void loop()
9 {
10 digitalWrite(13, HIGH);
11 delay(1000); // Wait for 1000 millisecond(s)
12 digitalWrite(13, LOW);
13 delay(1000); // Wait for 1000 millisecond(s)
14 }

Serial Monitor

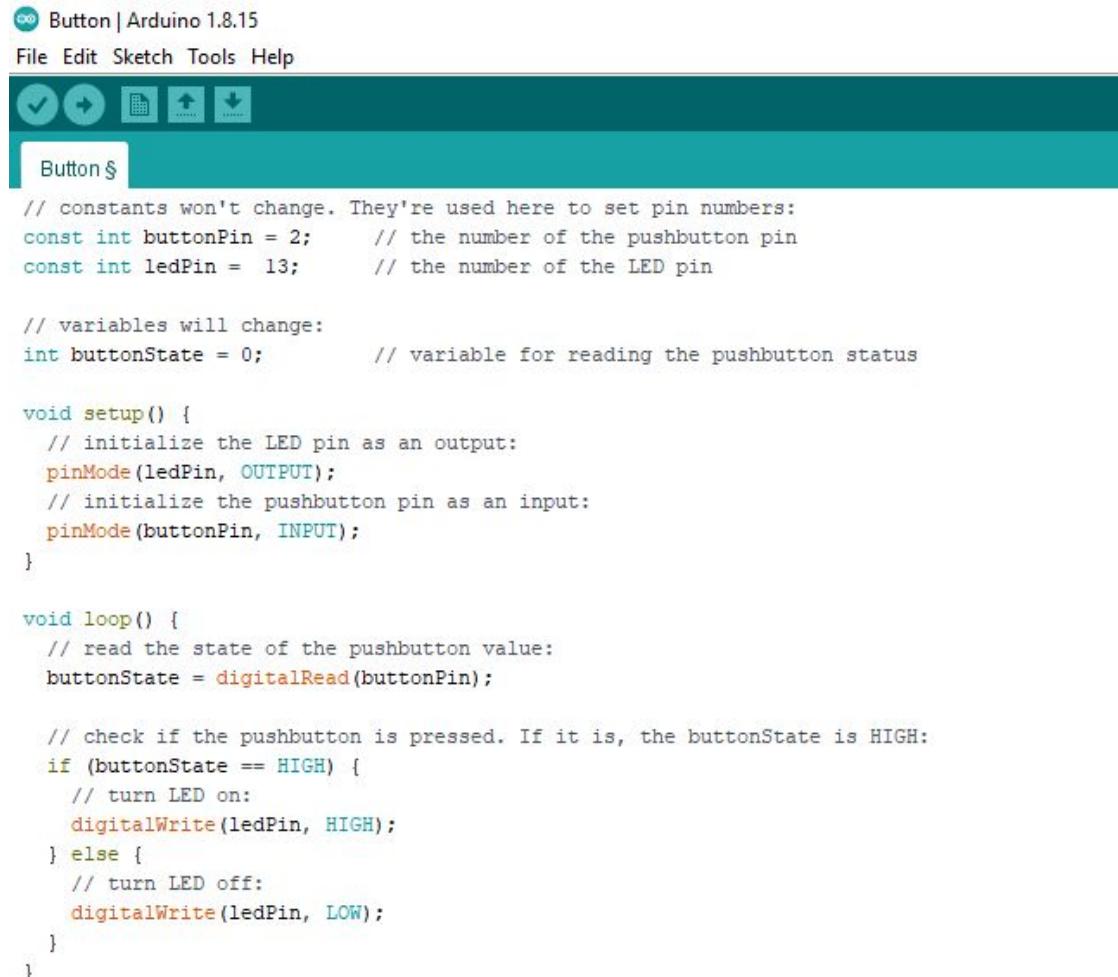
The image shows an Arduino Uno R3 microcontroller connected to a breadboard. A USB cable is connected to the Arduino's USB port. On the breadboard, there is a single LED connected to digital pin 13 via a resistor. The Arduino board has its power pins (5V, GND, and VIN) connected to the breadboard. The breadboard also features a 3.3V and 3.2V voltage regulator. Various other pins and components are visible on the breadboard.



Examples

Button

41



The screenshot shows the Arduino IDE interface with a sketch named "Button". The code is as follows:

```
// constants won't change. They're used here to set pin numbers:  
const int buttonPin = 2;      // the number of the pushbutton pin  
const int ledPin = 13;        // the number of the LED pin  
  
// variables will change:  
int buttonState = 0;          // variable for reading the pushbutton status  
  
void setup() {  
    // initialize the LED pin as an output:  
    pinMode(ledPin, OUTPUT);  
    // initialize the pushbutton pin as an input:  
    pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
    // read the state of the pushbutton value:  
    buttonState = digitalRead(buttonPin);  
  
    // check if the pushbutton is pressed. If it is, the buttonState is HIGH:  
    if (buttonState == HIGH) {  
        // turn LED on:  
        digitalWrite(ledPin, HIGH);  
    } else {  
        // turn LED off:  
        digitalWrite(ledPin, LOW);  
    }  
}
```

Moving a stepper motor



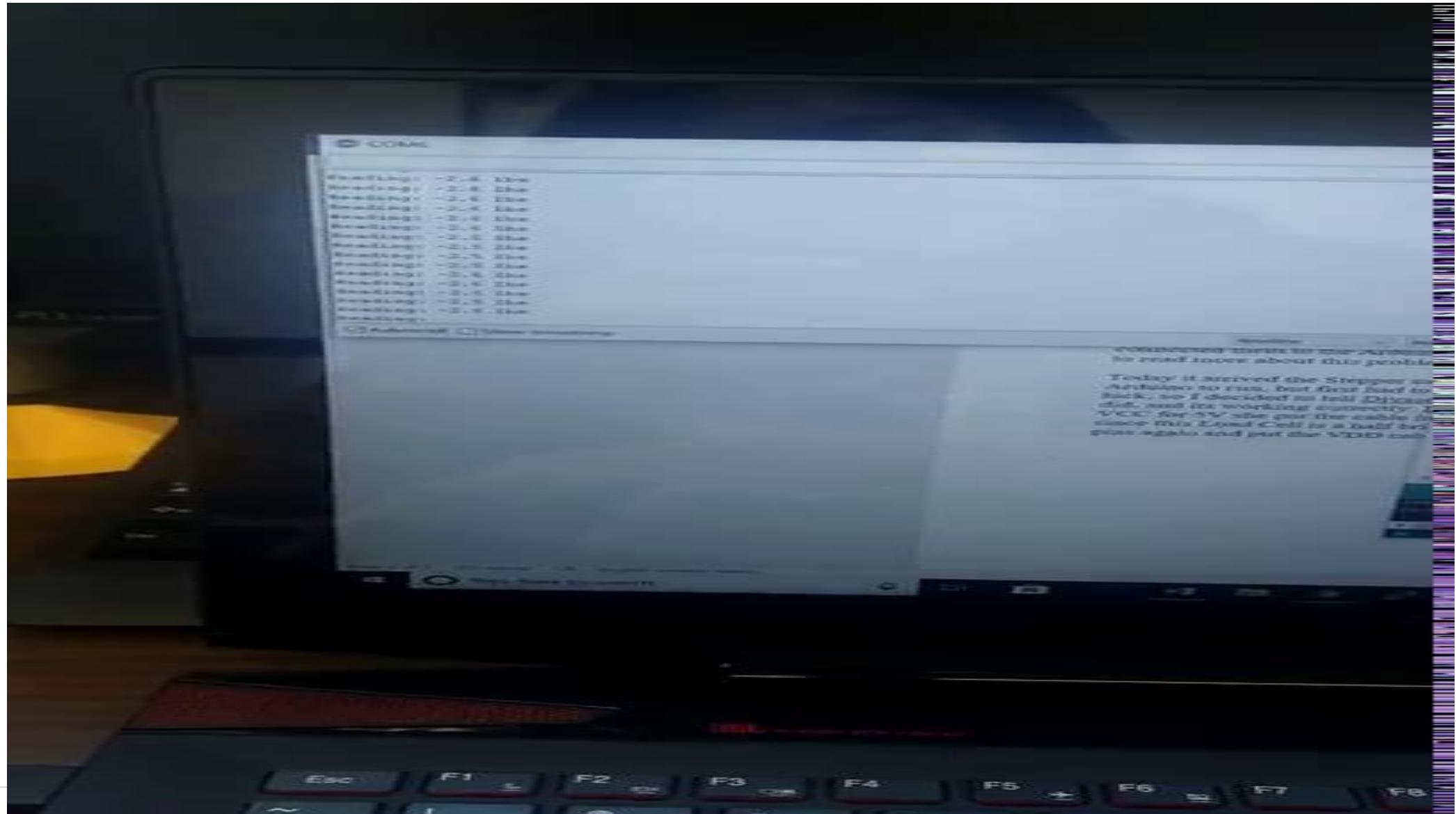
Moving a stepper motor using a potentiometer

43



Using a load cell

44



More projects



You can find more
projects at:

<https://create.arduino.cc/projecthub>

My
Questions



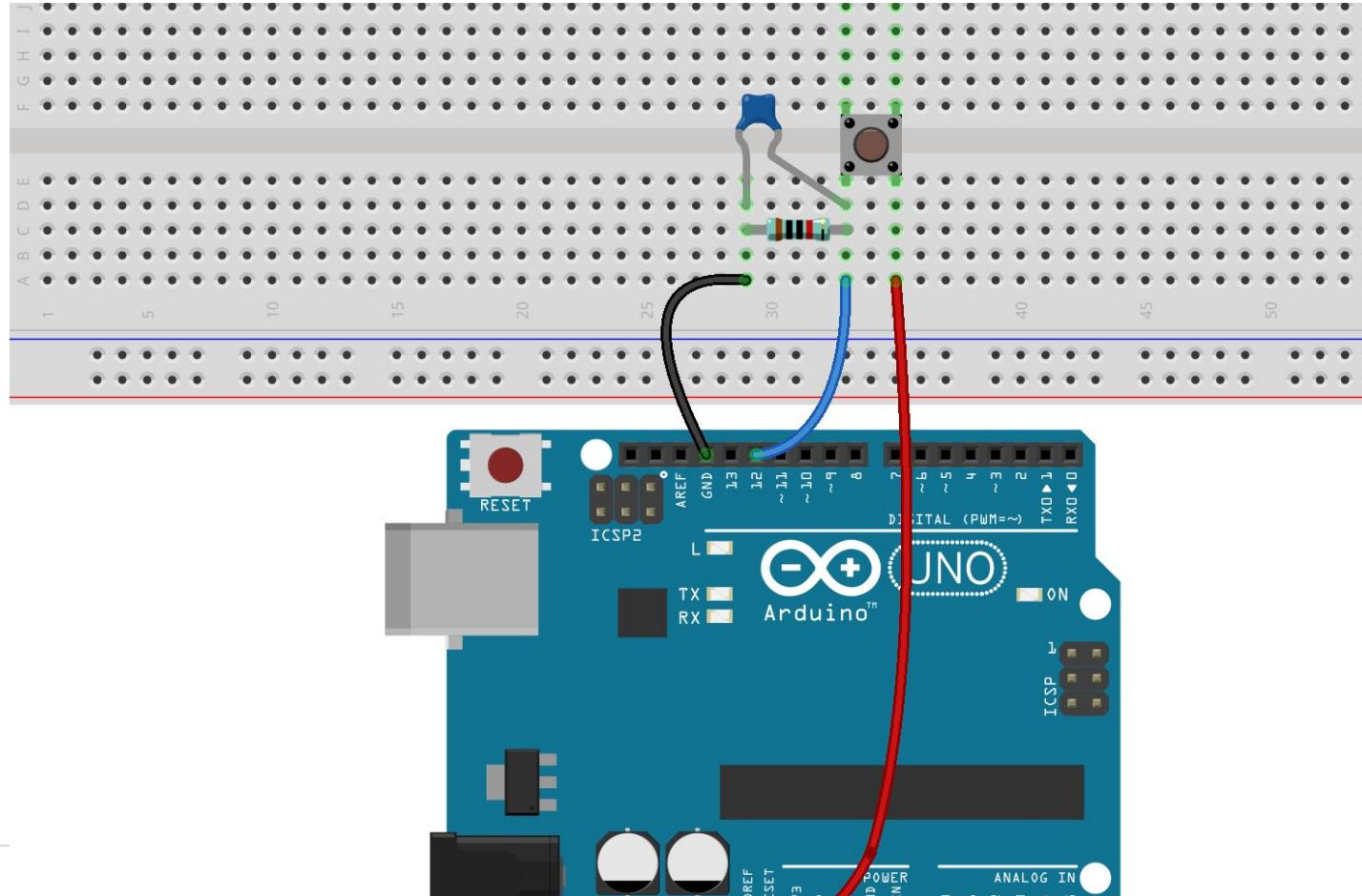
04

Hands On

Challenge!!

48

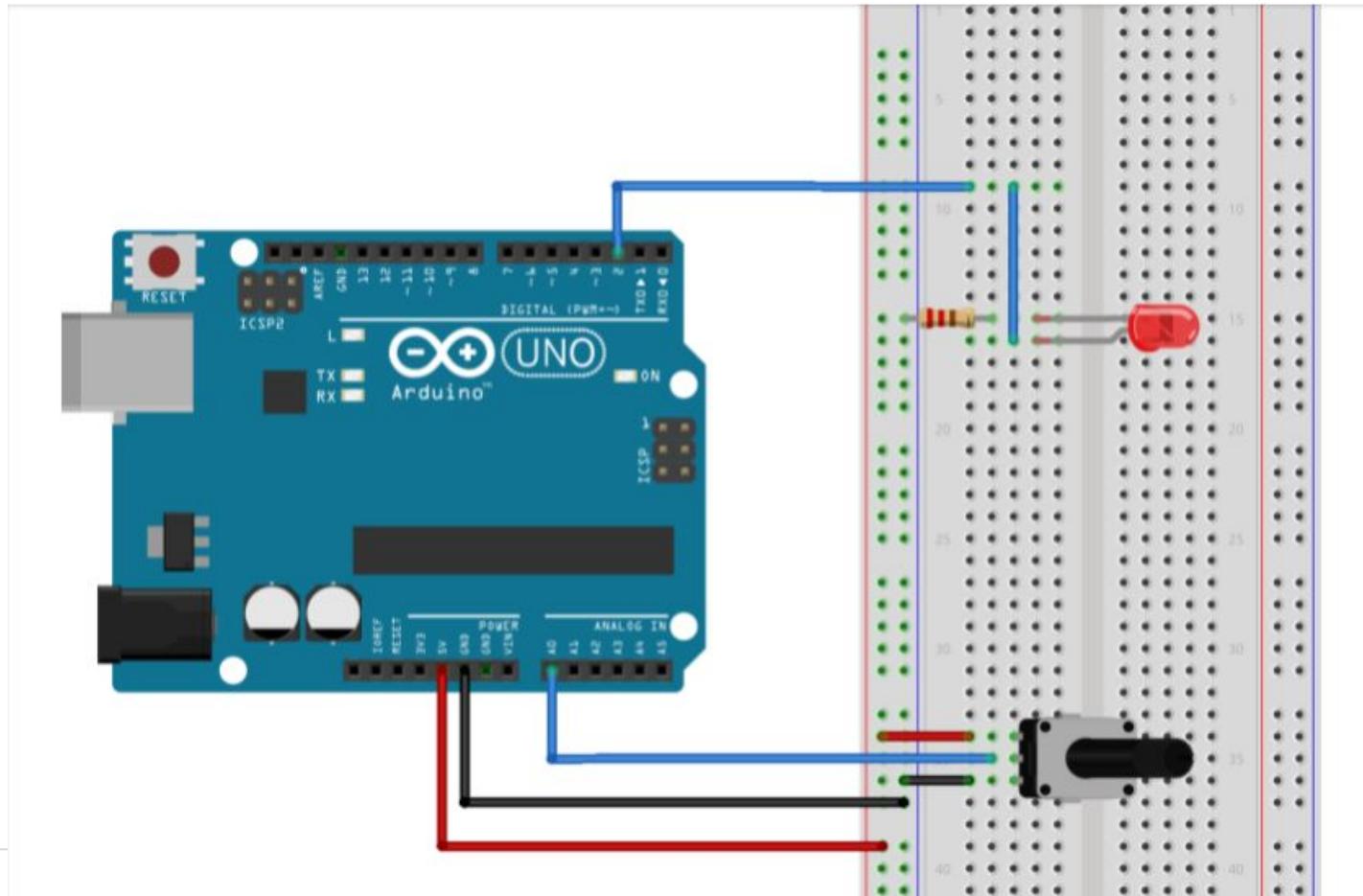
LED Control Using a button



Challenge!!

49

LED Brightness Control Using a Potentiometer



Challenge!!

50

Scrolling LED

