



Armstrong

School Program 2023-2024

Lesson 1



Armstrong

entertainment meets education



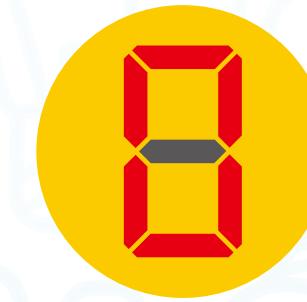
Course Content



Microcontrollers



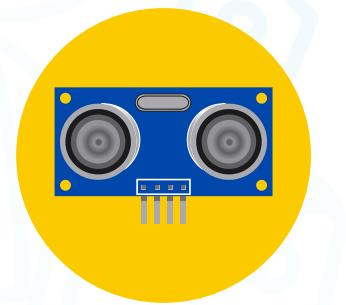
Programming



7 Segment



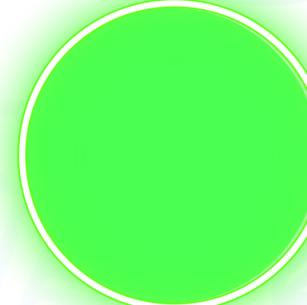
LCD/I₂C



Ultrasonic sensor



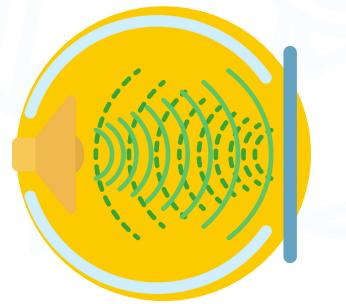
IR remote control



RING LED strip



IR sensor



Sound sensor



ESP webserver

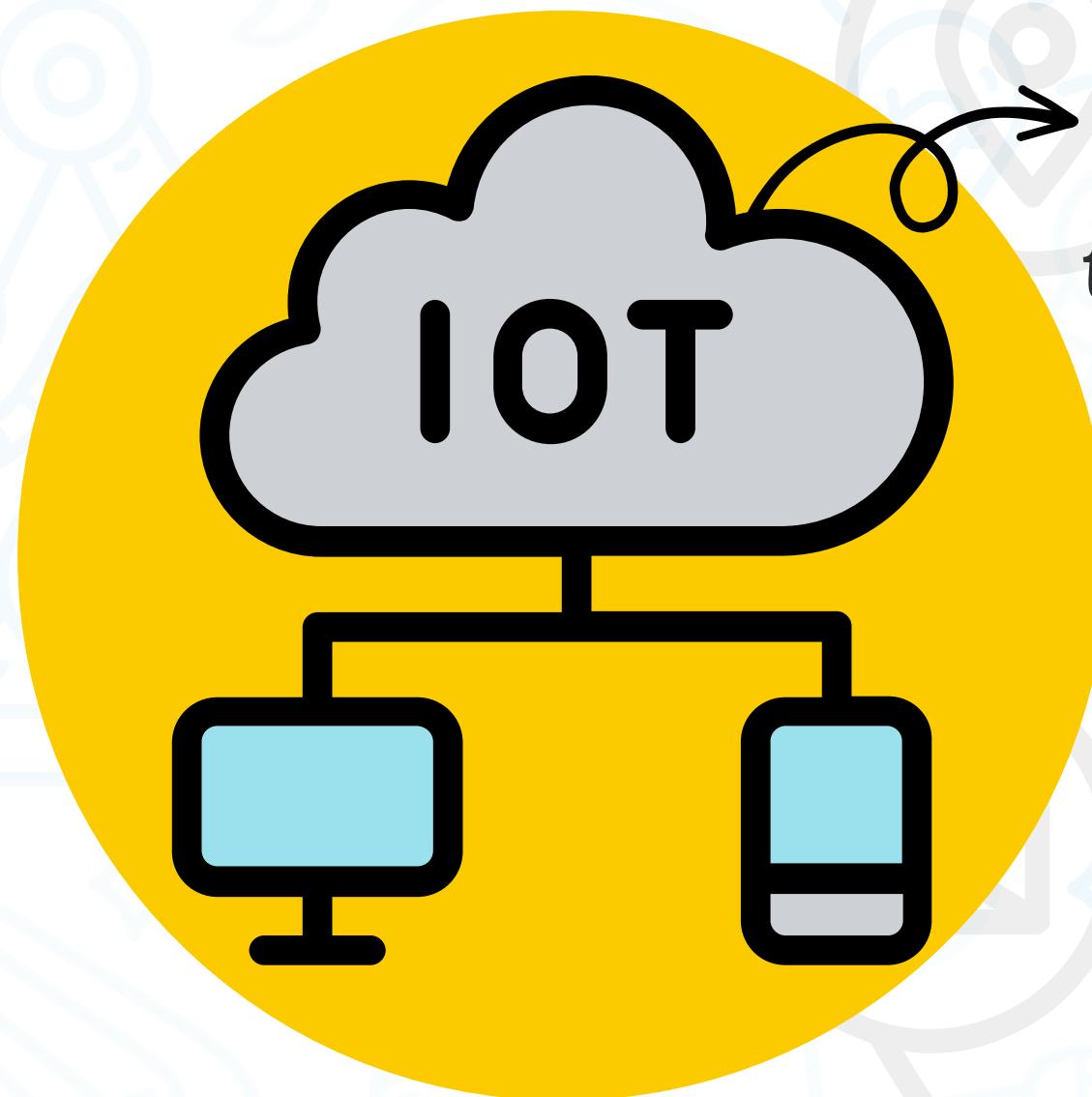


MIT APP Inventor



AC remote control

Arduino Electronics&IoT



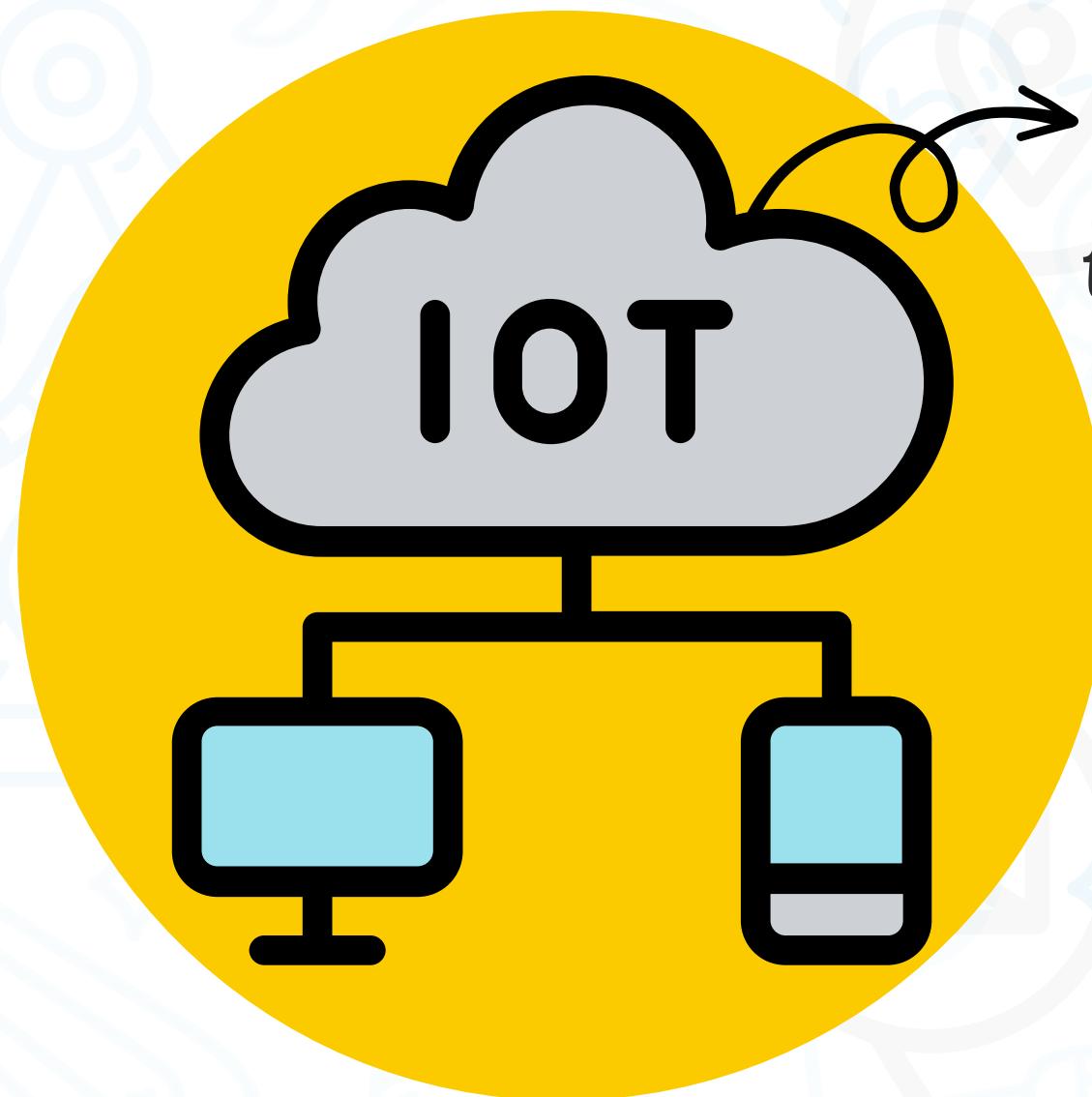
The Internet of Things is like a big team of smart devices and machines that can share information with each other using the internet.



Internet of Things

Automation

Arduino Electronics&IoT



The Internet of Things is like a big team of smart devices and machines that can share information with each other using the internet.

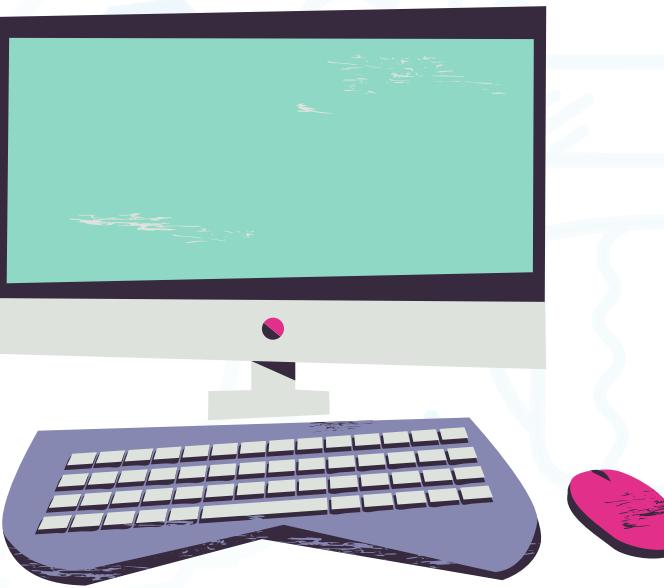
Internet of Things



Automation is the use of technology, machines, or computer programs to perform tasks and process without human intervention.

Automation

Software vs Hardware



Software vs Hardware



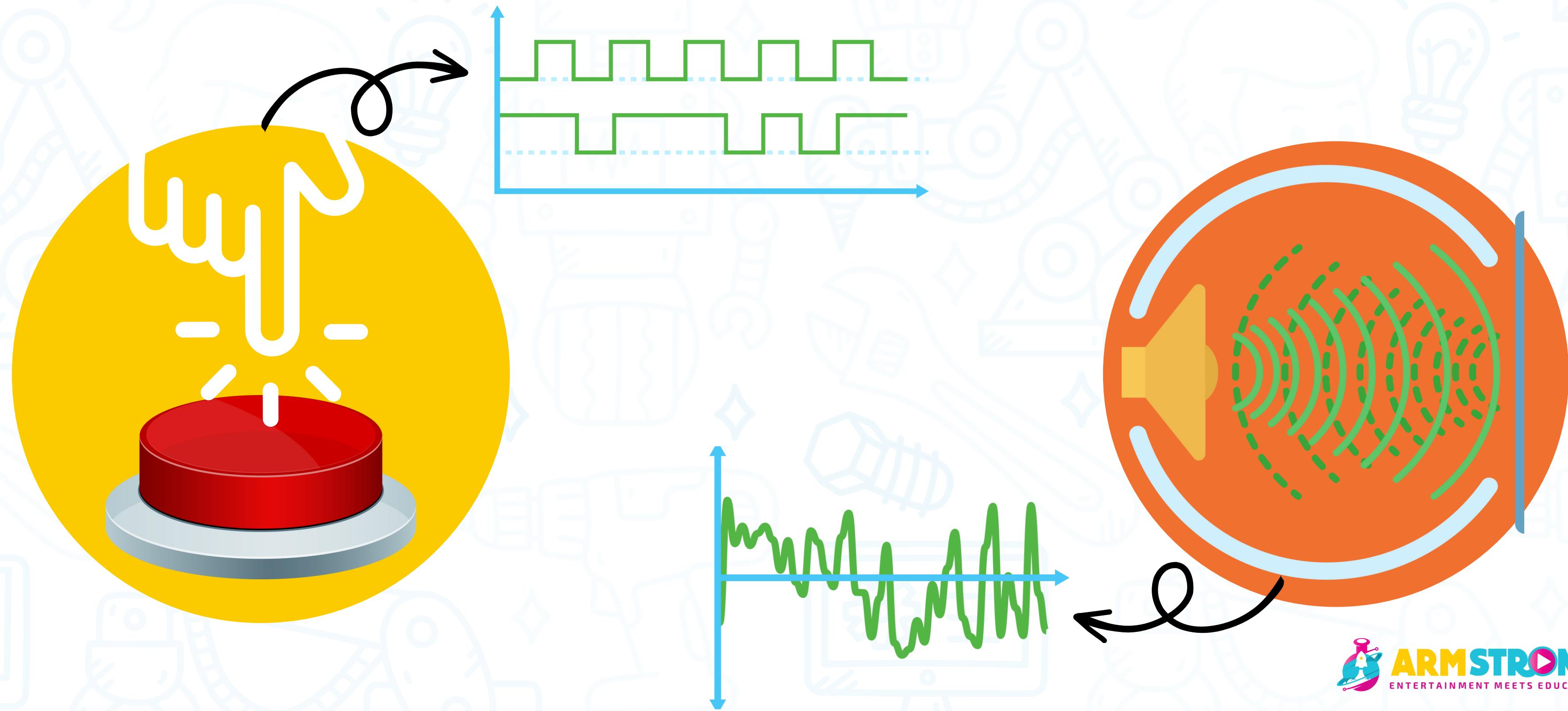
Devices



INPUT

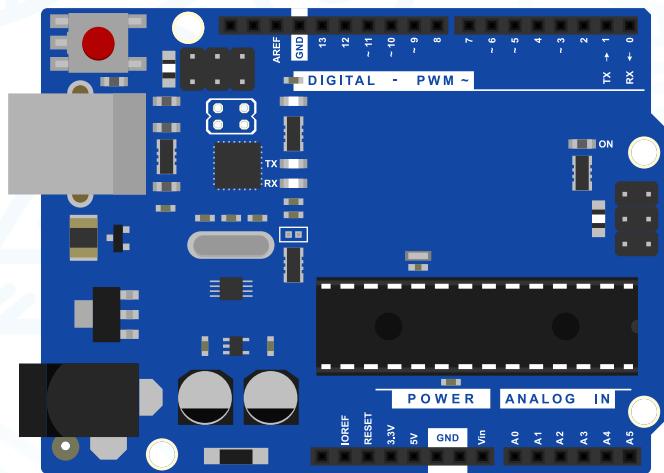
OUTPUT

Digital vs Analog

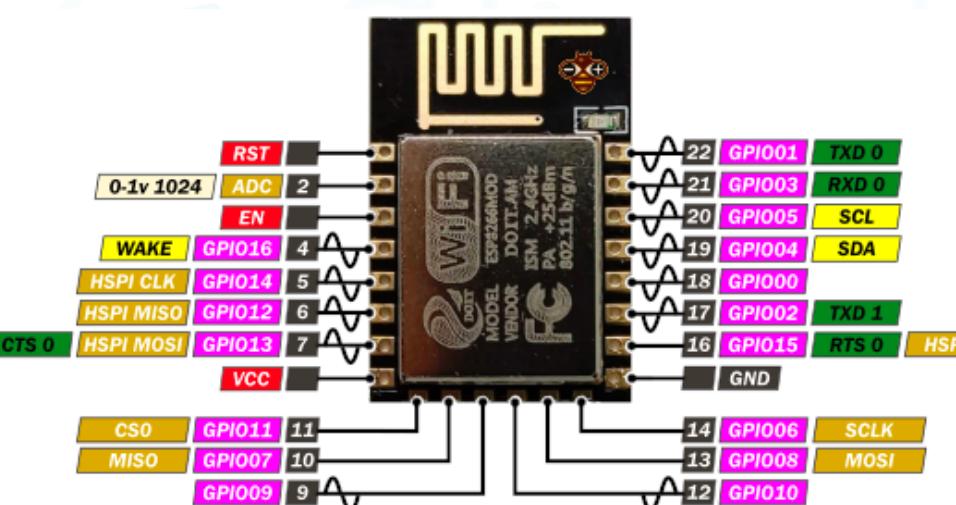


Microcontrollers

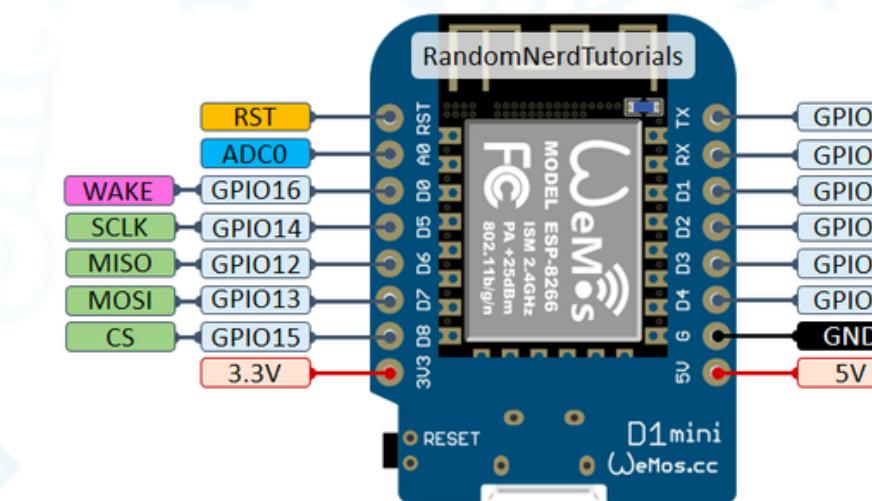
Arduino



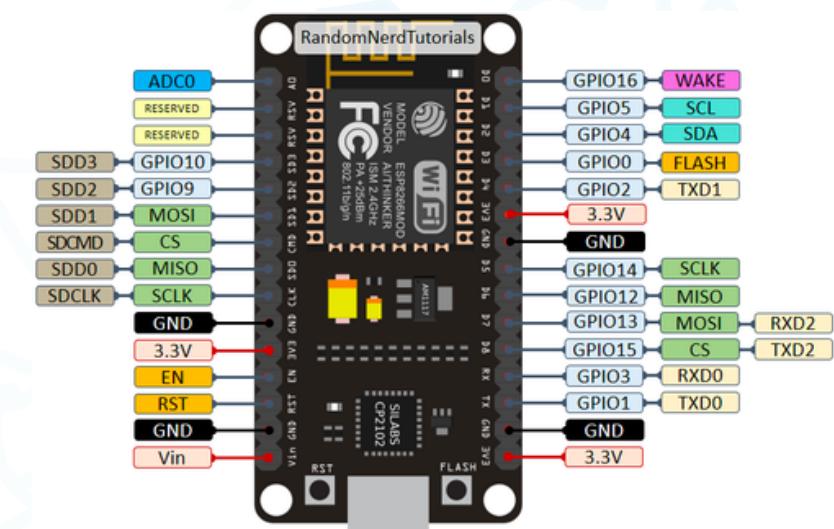
ESP8266



WeMos D1



NOD MCU

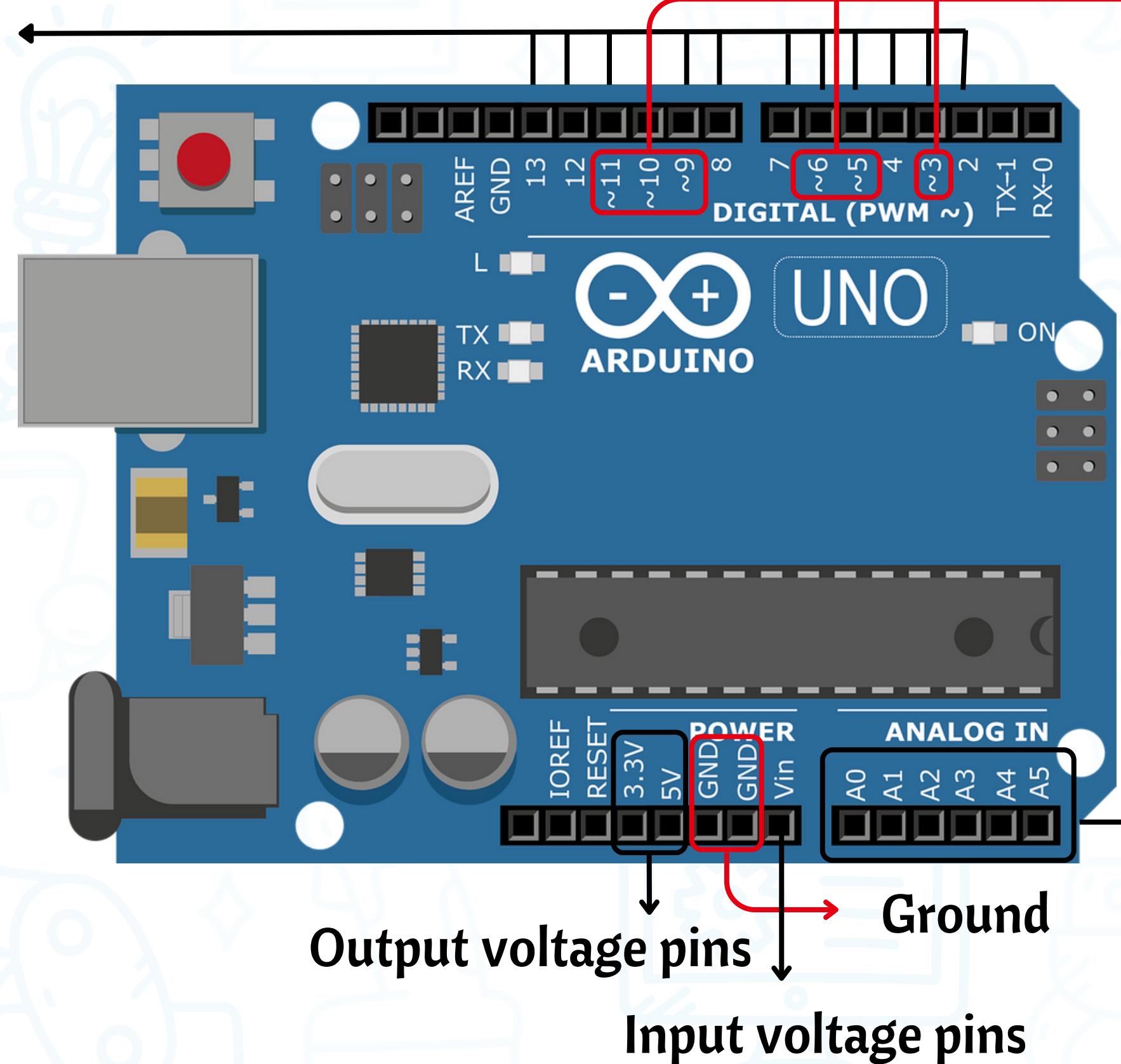


ESP

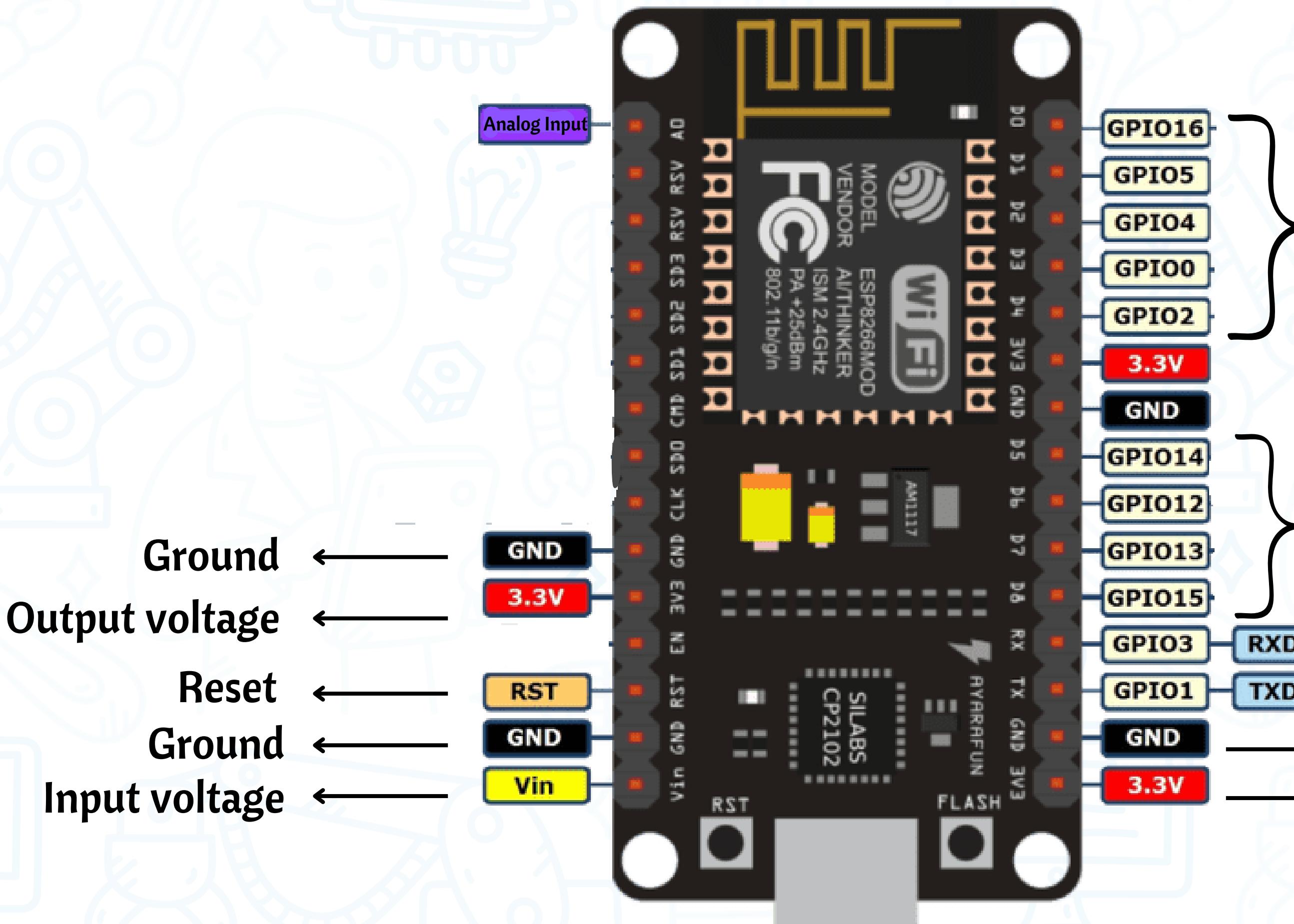
Intro to Arduino

Digital input/output pins

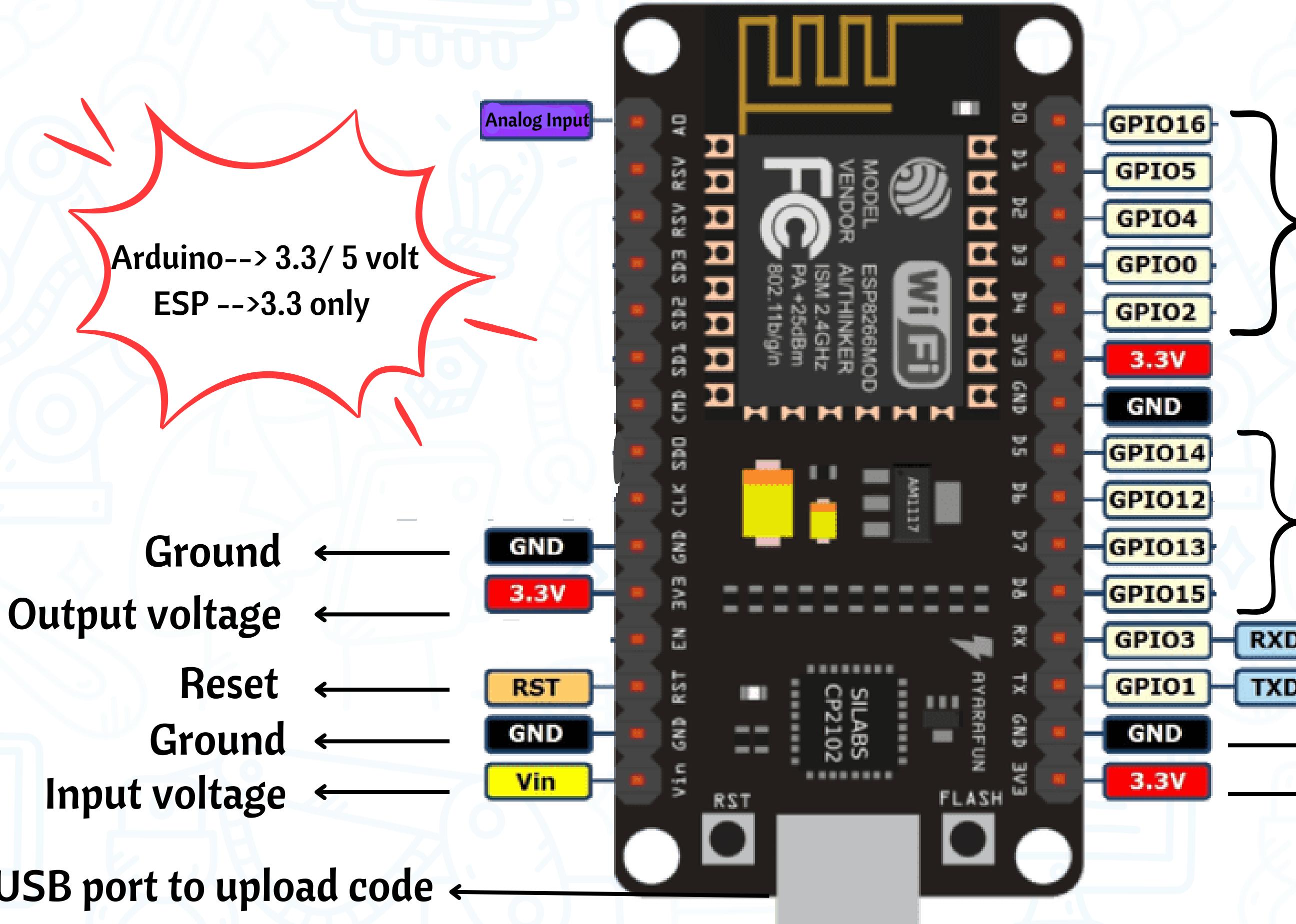
Analog output pins



Intro to ESP8266



Intro to ESP8266

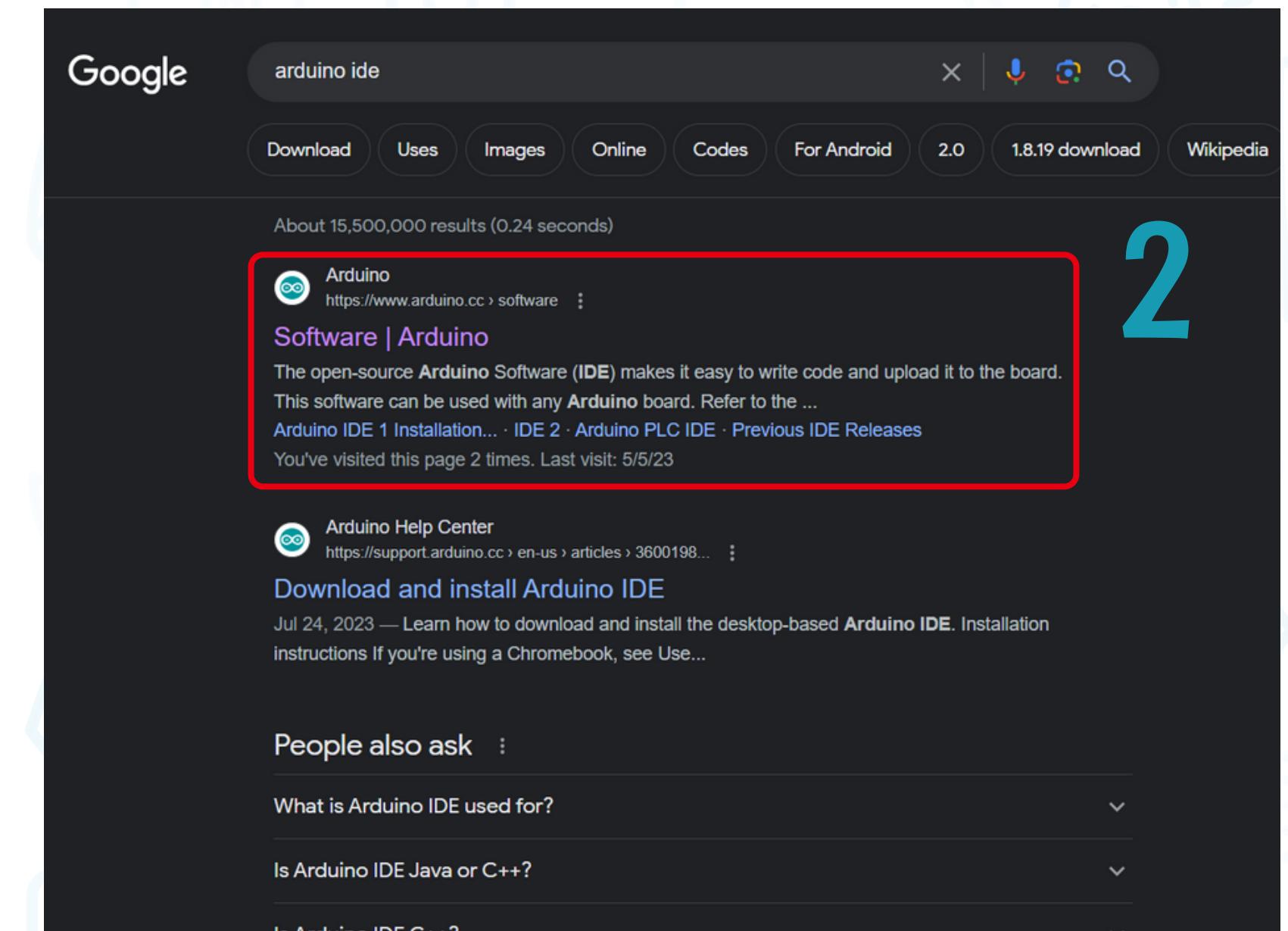
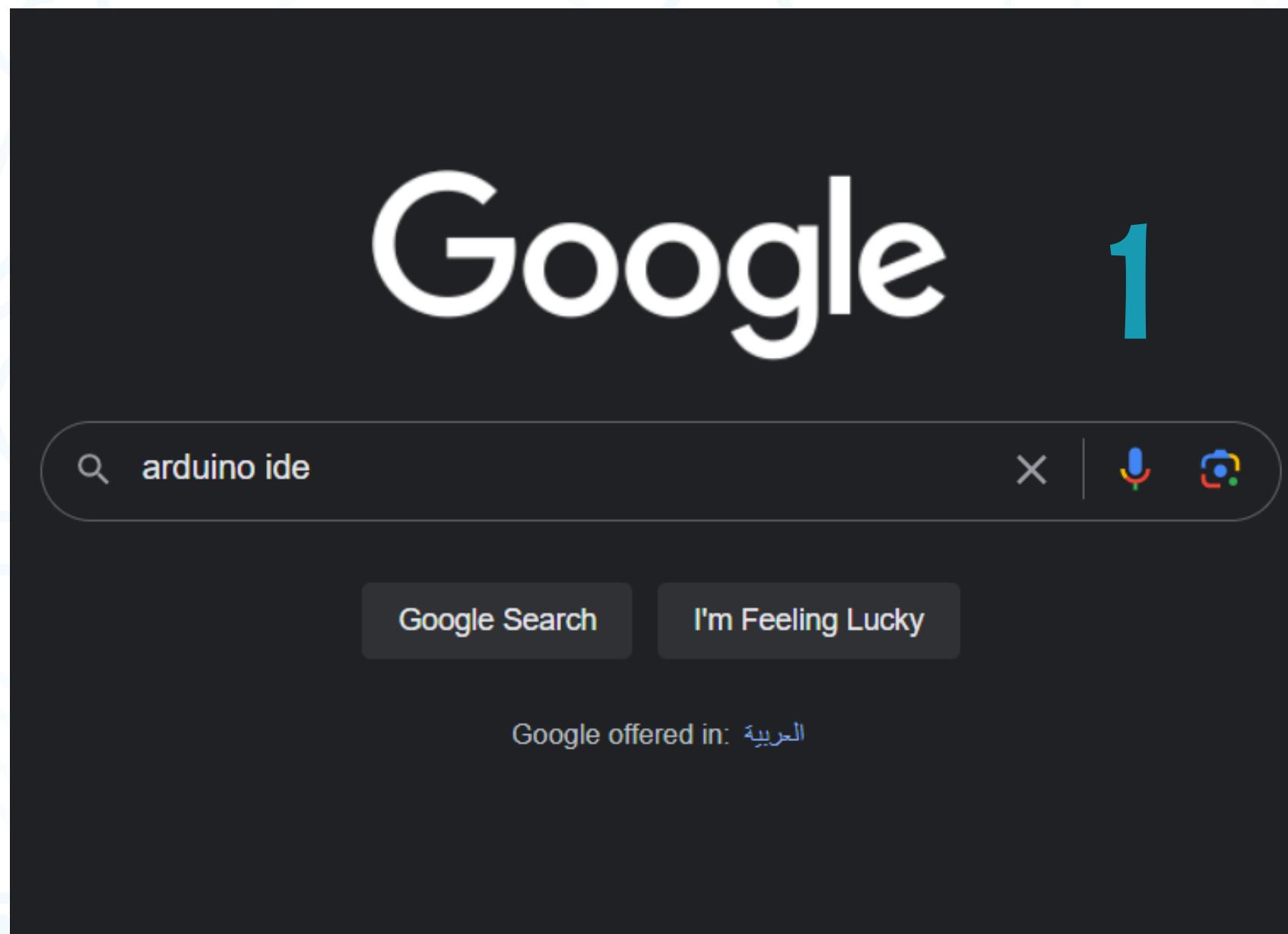


Digital Input/Output pins

Communication Pins

→ Ground
→ Output voltage

Getting started with Arduino IDE



Getting started with Arduino IDE

The screenshot shows the Arduino website's software download page. At the top, there are navigation links for HARDWARE, SOFTWARE (which is highlighted in yellow), CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. Below this, there are two buttons: 'CODE ONLINE' and 'GETTING STARTED'. A search bar labeled 'Search on Arduino' is at the top right. The main content area features a large image of a computer monitor displaying code, with sections for Examples, Libraries, and Serial Monitor. Below the image, there are download options for Windows, Linux, and macOS.

Downloads

3

This screenshot shows the download page for Arduino IDE 2.1.1. It features a large image of the IDE interface with the title 'Arduino IDE 2.1.1' above it. Below the image, there is a brief description of the new features and a link to the documentation. A red box highlights the 'DOWNLOAD OPTIONS' section, which lists download links for Windows, Linux, and macOS. At the bottom, there is a link to the source code on GitHub.

The screenshot shows the Arduino website's donation page. At the top, there are navigation links for HARDWARE, SOFTWARE (highlighted in yellow), CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. Below this, there is a section titled 'Support the Arduino IDE' with a message about the download count (74,720,454 times) and a call to action for donations. A red box highlights the '\$50' donation button. Below this, there are two buttons: 'JUST DOWNLOAD' (highlighted with a red box) and 'CONTRIBUTE & DOWNLOAD'. To the right, there is a cartoon illustration of a character holding a bottle of coins.

4

Getting started with Arduino IDE

Arduino IDE Setup

License Agreement

Please review the license terms before installing Arduino IDE.

Press Page Down to see the rest of the agreement.

Terms of Service

The Arduino software is provided to you "as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted, even if advised of the possibility or likelihood of such damages.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Arduino IDE.

Arduino IDE 2.1.1

I Agree

Cancel

5

Arduino IDE Setup

Choose Installation Options

Who should this application be installed for?

- Anyone who uses this computer (all users)
- Only for me

There is already a per-user installation. (C:\Users\armstrong\AppData\Local\Programs\Arduino IDE)
Will reinstall/upgrade.

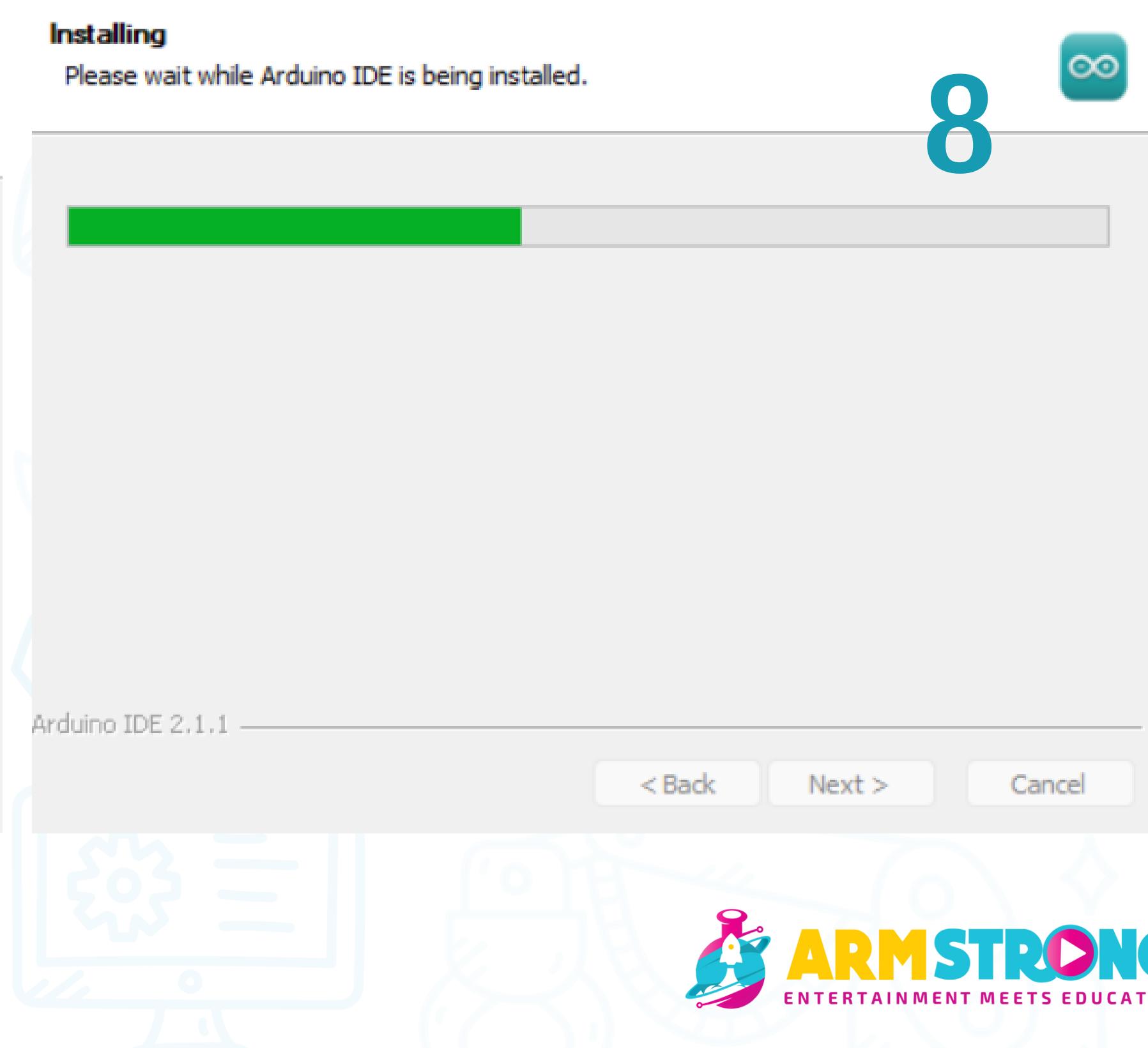
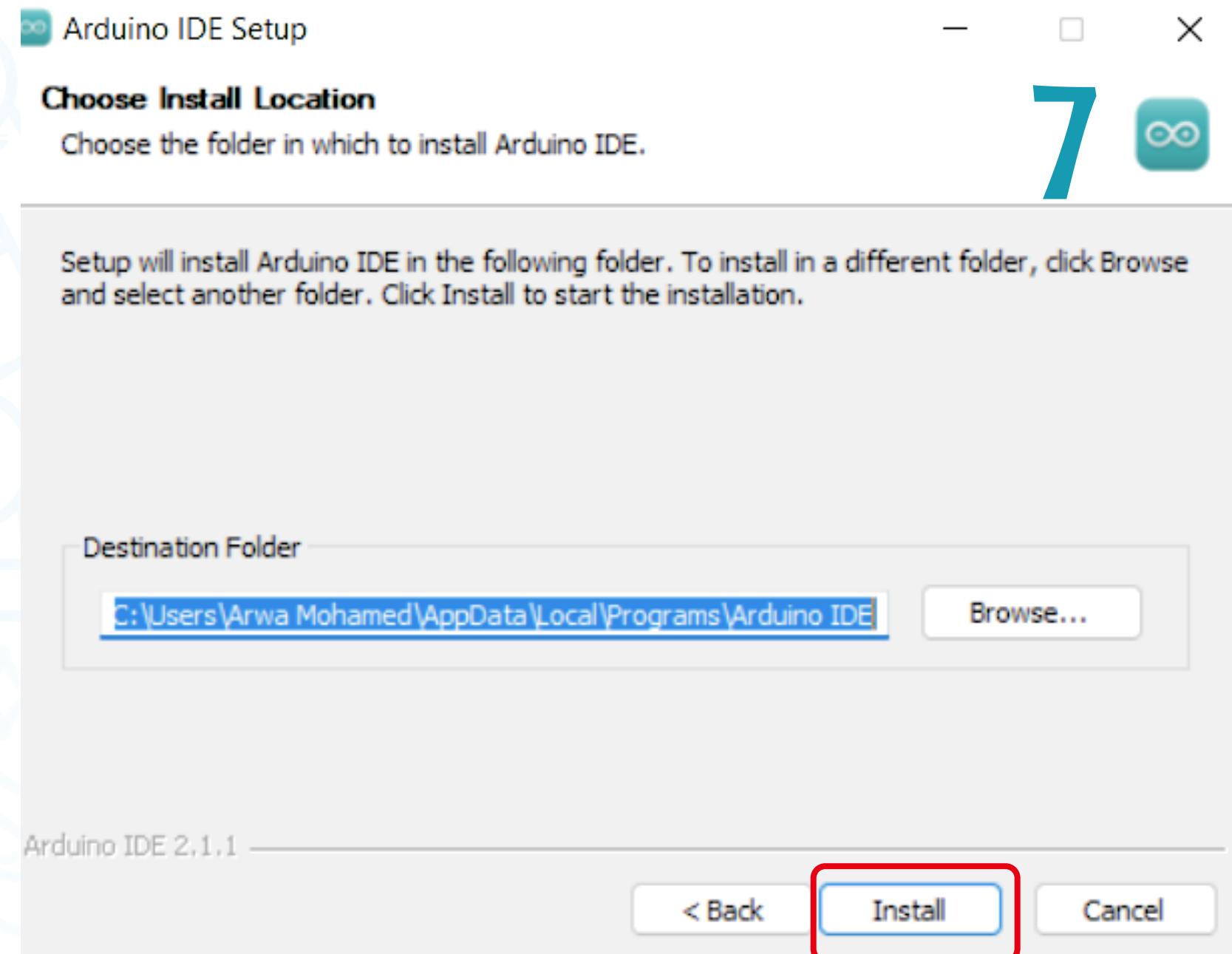
Arduino IDE 2.1.1

< Back

Next >

Cancel

Getting started with Arduino IDE

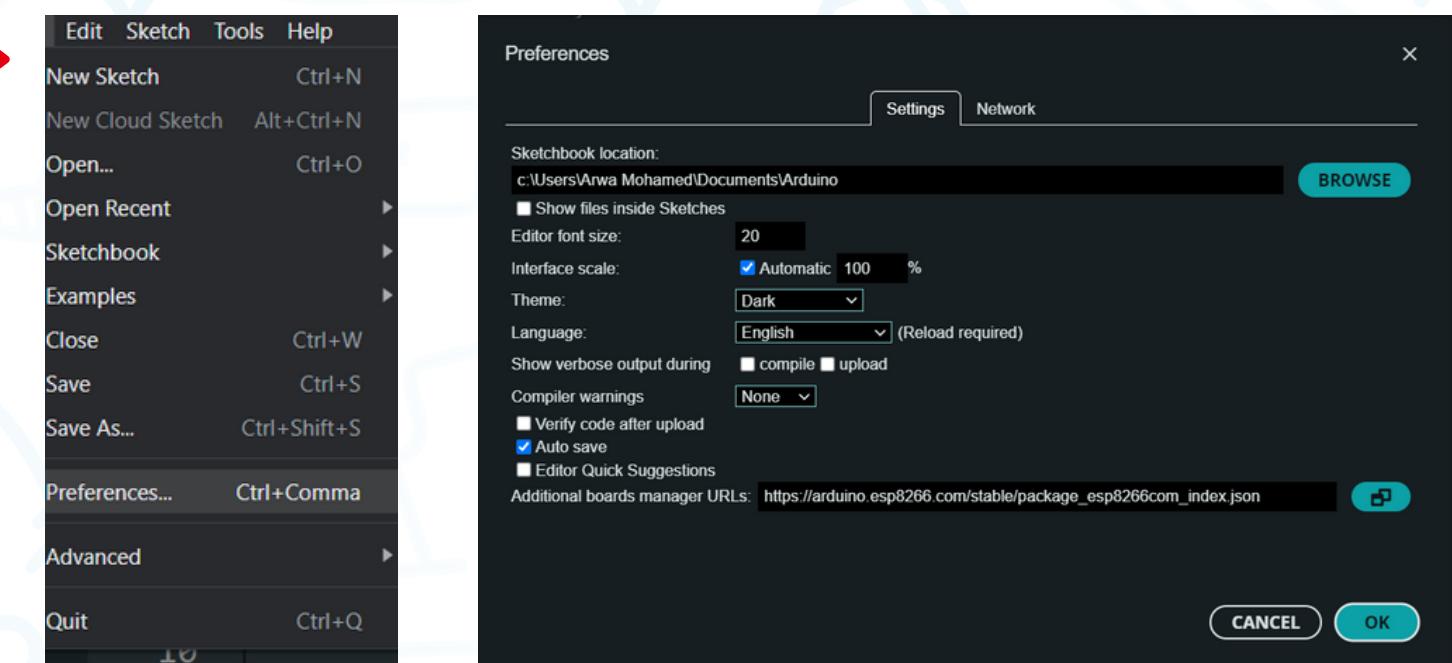
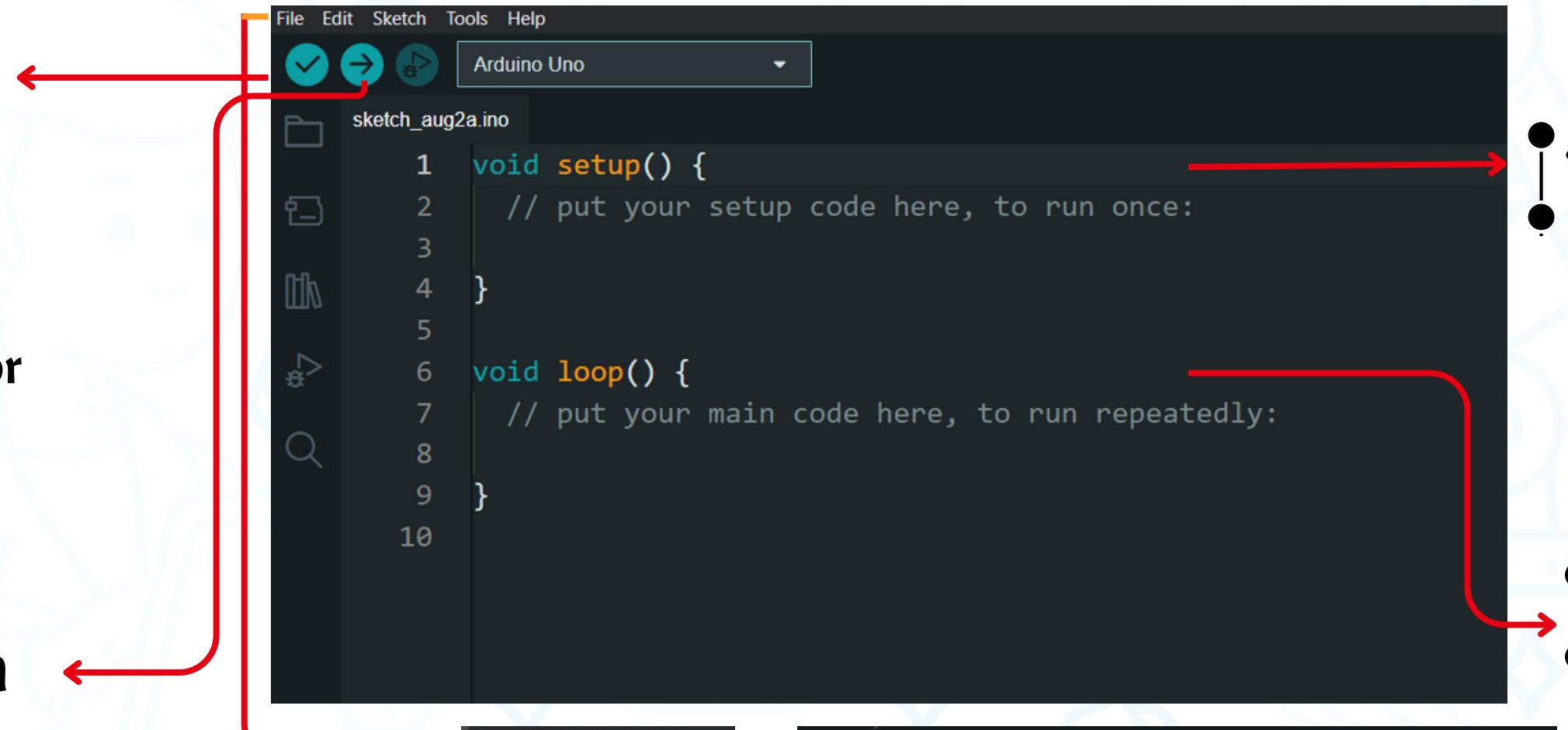


Getting started with Arduino IDE

Verify Button

- Syntax error
- Logic error
- Runtime error

Upload Button

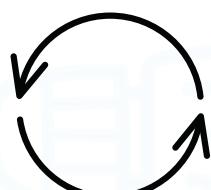


Void setup()

- getting ready to start any project on the Arduino
- happens only once at the beginning, each time we start a new project .

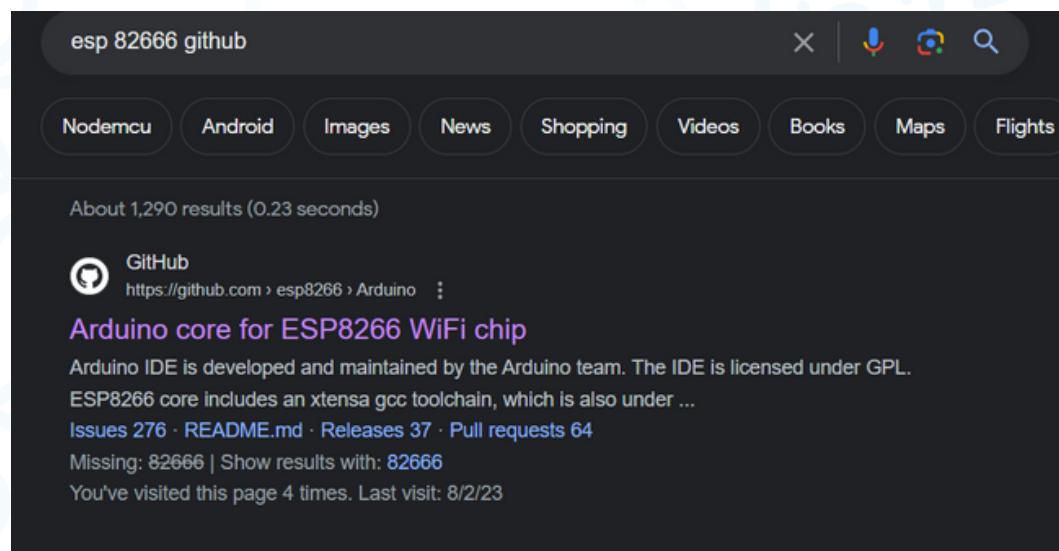
Void loop()

- put all the Arduino code to do repeatedly.
- As long as the Arduino is powered on, it will keep playing the loop and doing those instructions we told it to do.

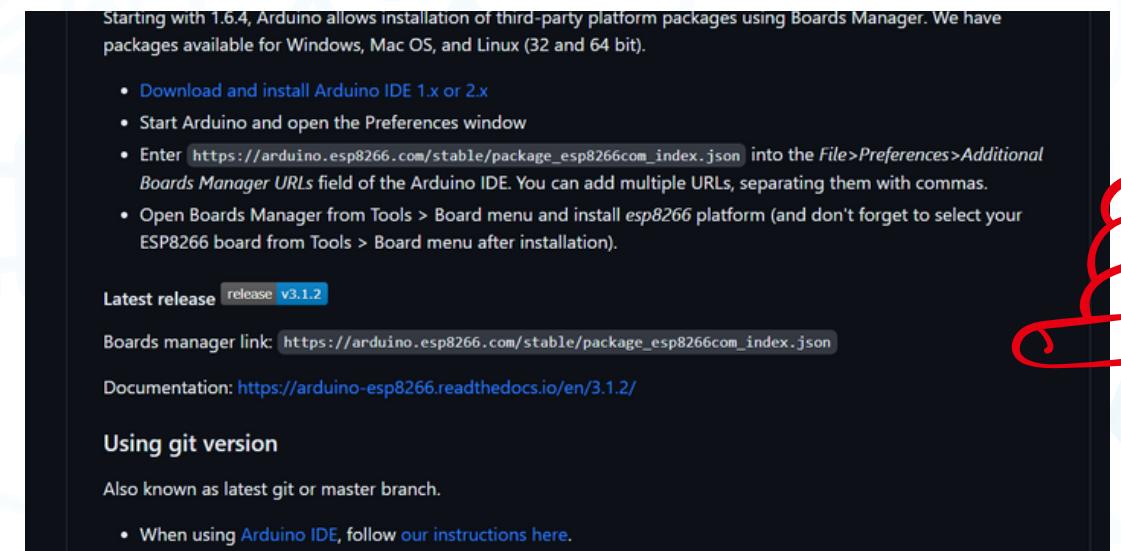


Getting started with Arduino IDE

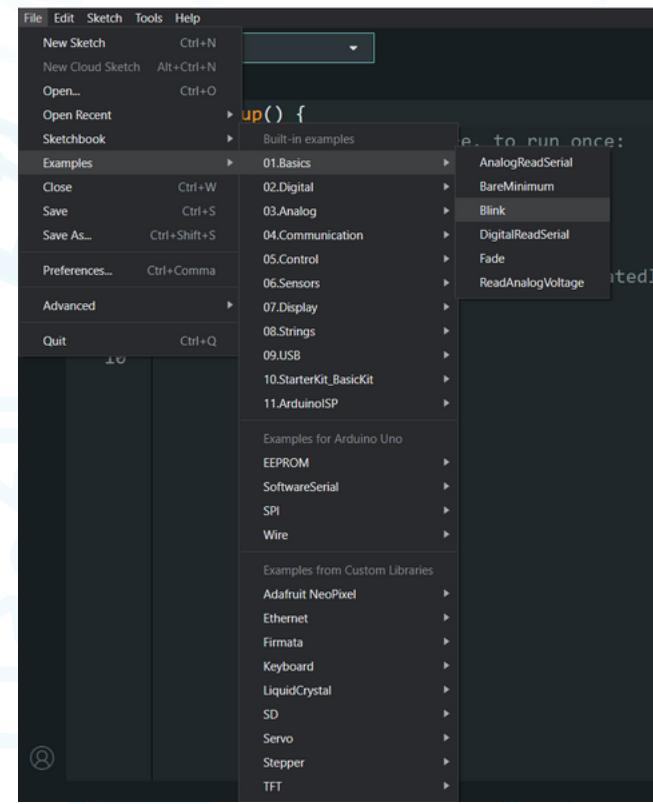
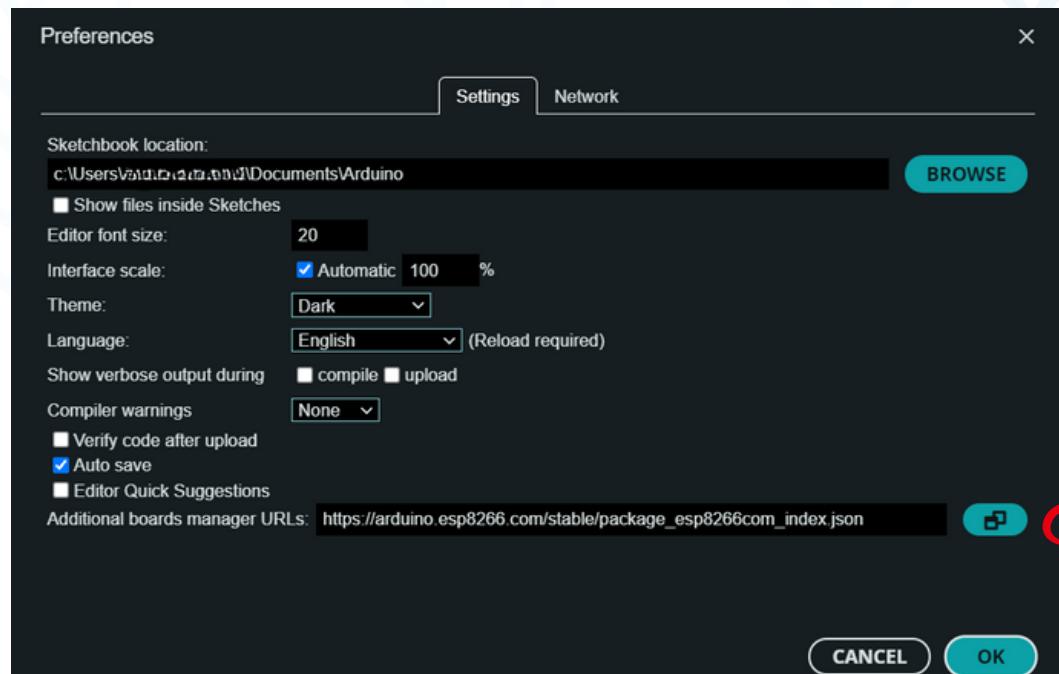
1 Go to Google ->Search for ESP8266 Github



2 Open the official ESP8266 GitHub -> You will find a link to copy



3 Open preferences--> add the link in the additional board



4 Open examples for the ESP and choose the **Blinking example** to start with

Blinking Example

pinMode(,)

Pin number

INPUT,OUTPUT

digitalWrite(,)

Pin number

HIGH,LOW

```
1//  
2  ESP8266 Blink by Simon Peter  
3  Blink the blue LED on the ESP-01 module  
4  This example code is in the public domain  
5  
6  The blue LED on the ESP-01 module is connected to GPIO1  
7  (which is also the TXD pin; so we cannot use Serial.print() at the same time)  
8  
9  Note that this sketch uses LED_BUILTIN to find the pin with the internal LED  
10 */  
11  
12 void setup() {  
13   pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output  
14 }  
15  
16 // the loop function runs over and over again forever  
17 void loop() {  
18   digitalWrite(LED_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level  
19   // but actually the LED is on; this is because  
20   // it is active low on the ESP-01)  
21   delay(1000);  
22   digitalWrite(LED_BUILTIN, HIGH);  
23   delay(2000);  
24 }
```

delay(milliseconds)

/*comments*/

//comments



Remember

HIGH= on, LOW=off

In programming a pin is **HIGH** when it outputs electricity (led on) and **LOW** when it doesn't output electricity.



delay(milliseconds) Concept

delay(2000)

2000 milliseconds = 2 second

delay(1000)

1000 milliseconds = 1 second

delay(100)

100 milliseconds = 0.1second



before **delay(1000)**



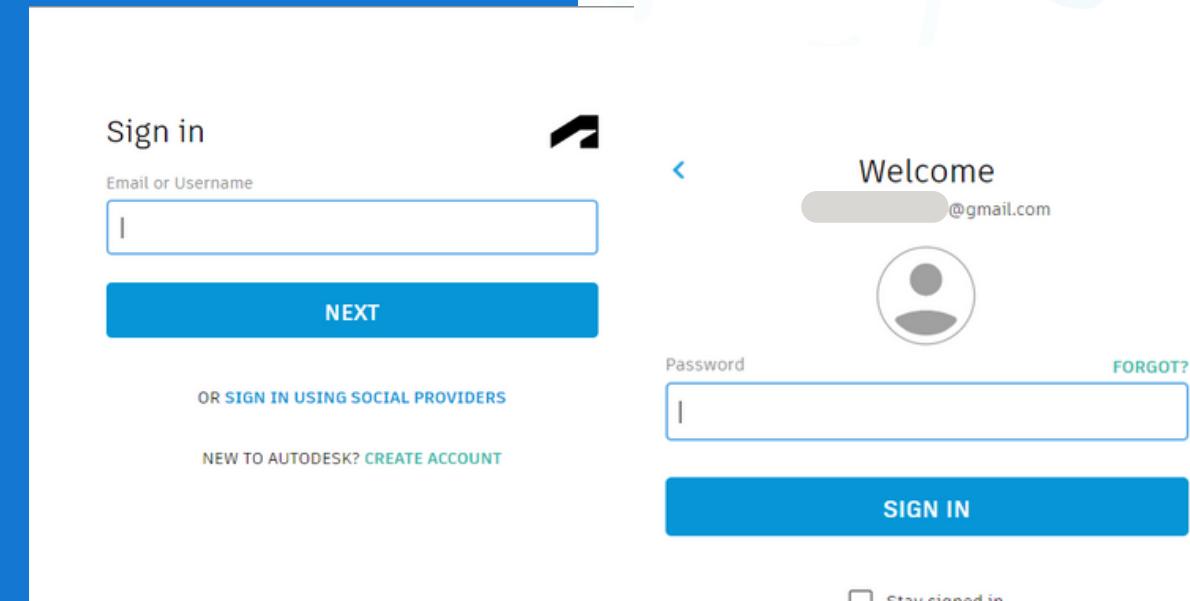
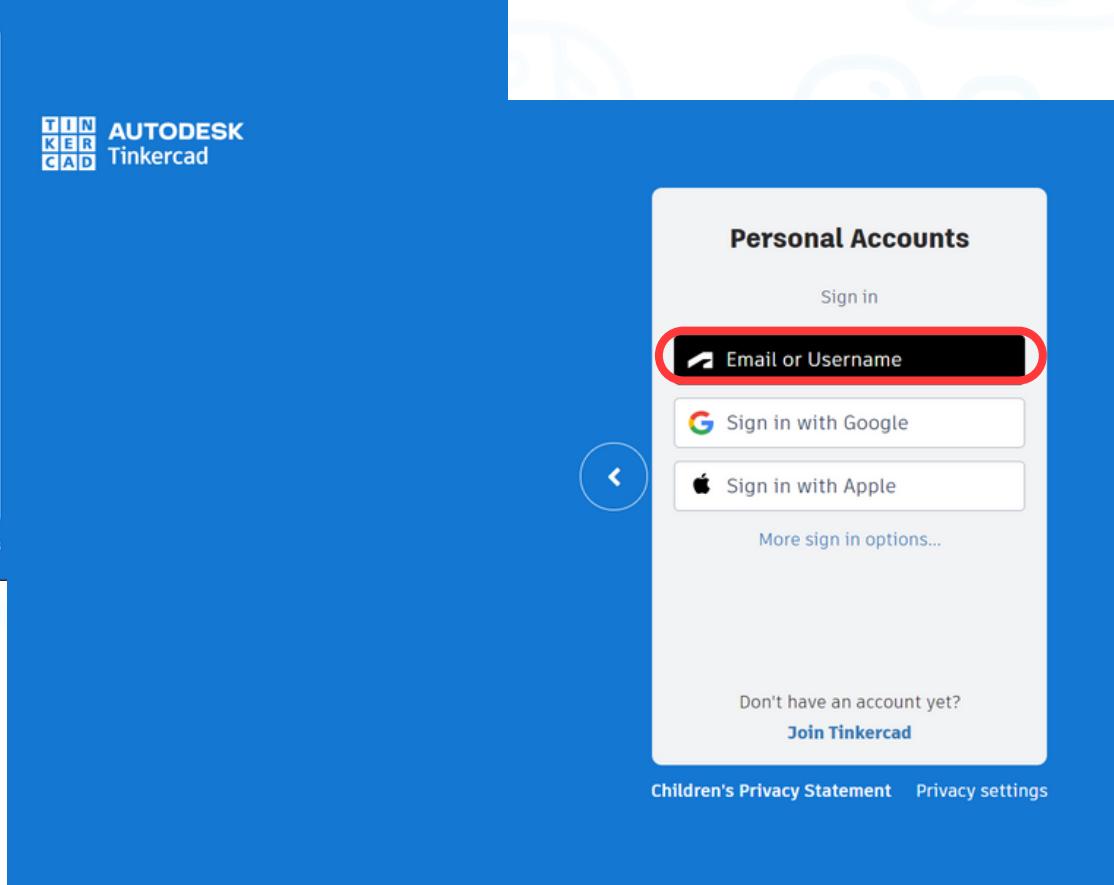
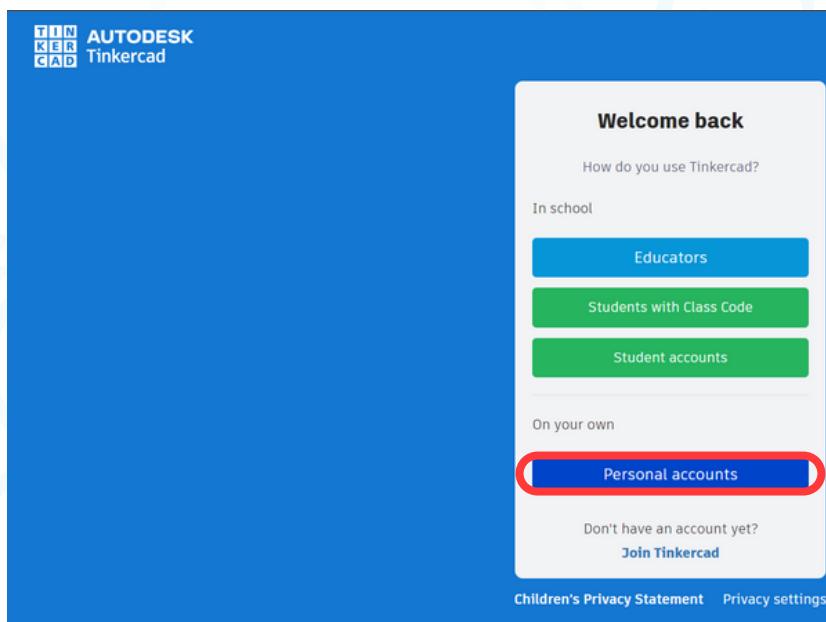
delay(1000)



after **delay(1000)**

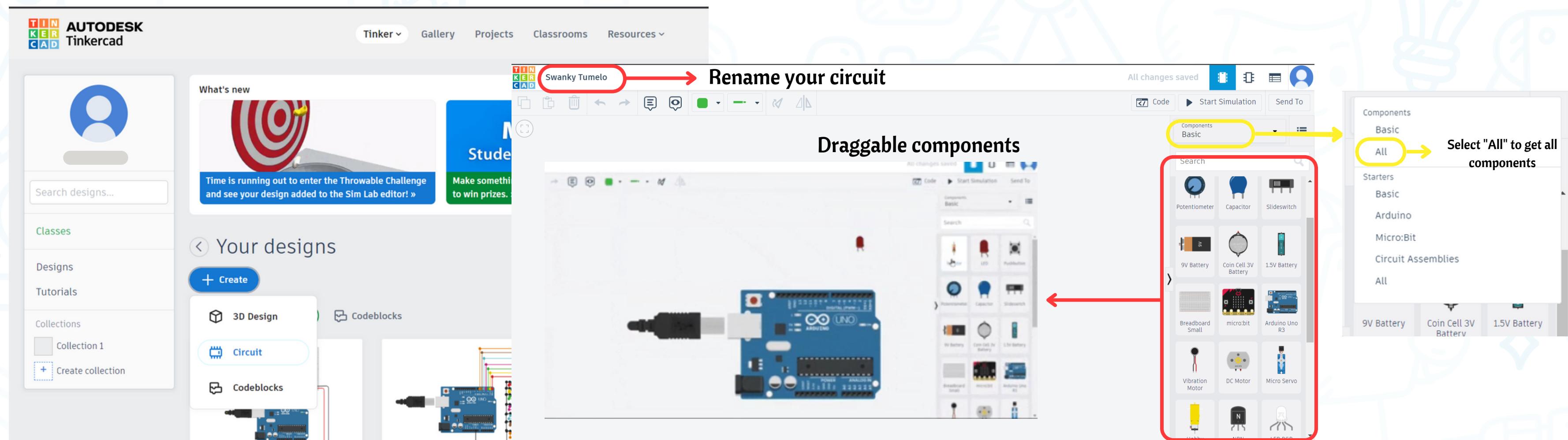
Getting started with Tinkercad

Step 1: sign in.



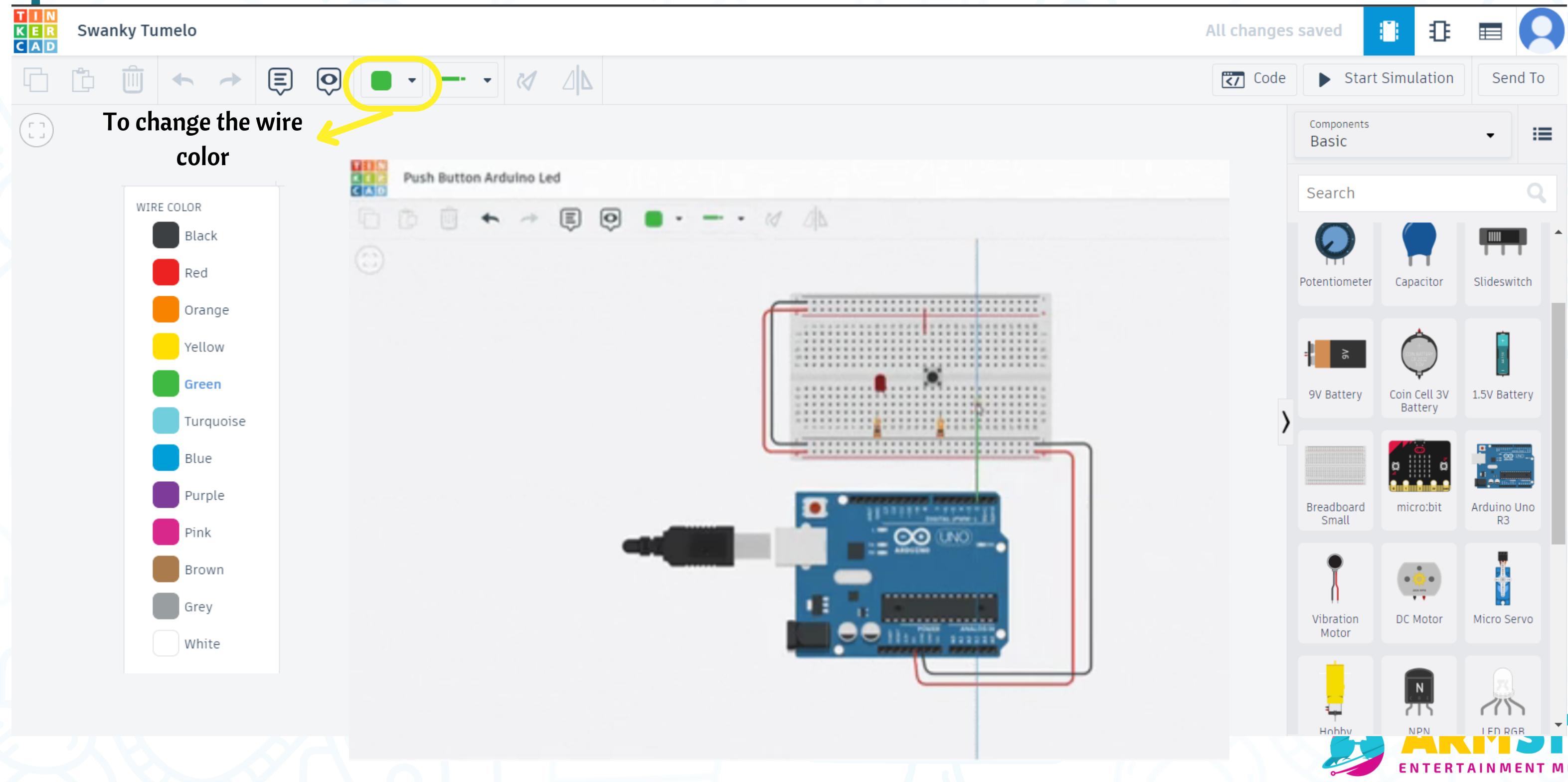
Getting started with Tinkercad

Step 2: create new circuit.



Getting started with Tinkercad

Step 3: connect and simulate.



Let's start our first Example

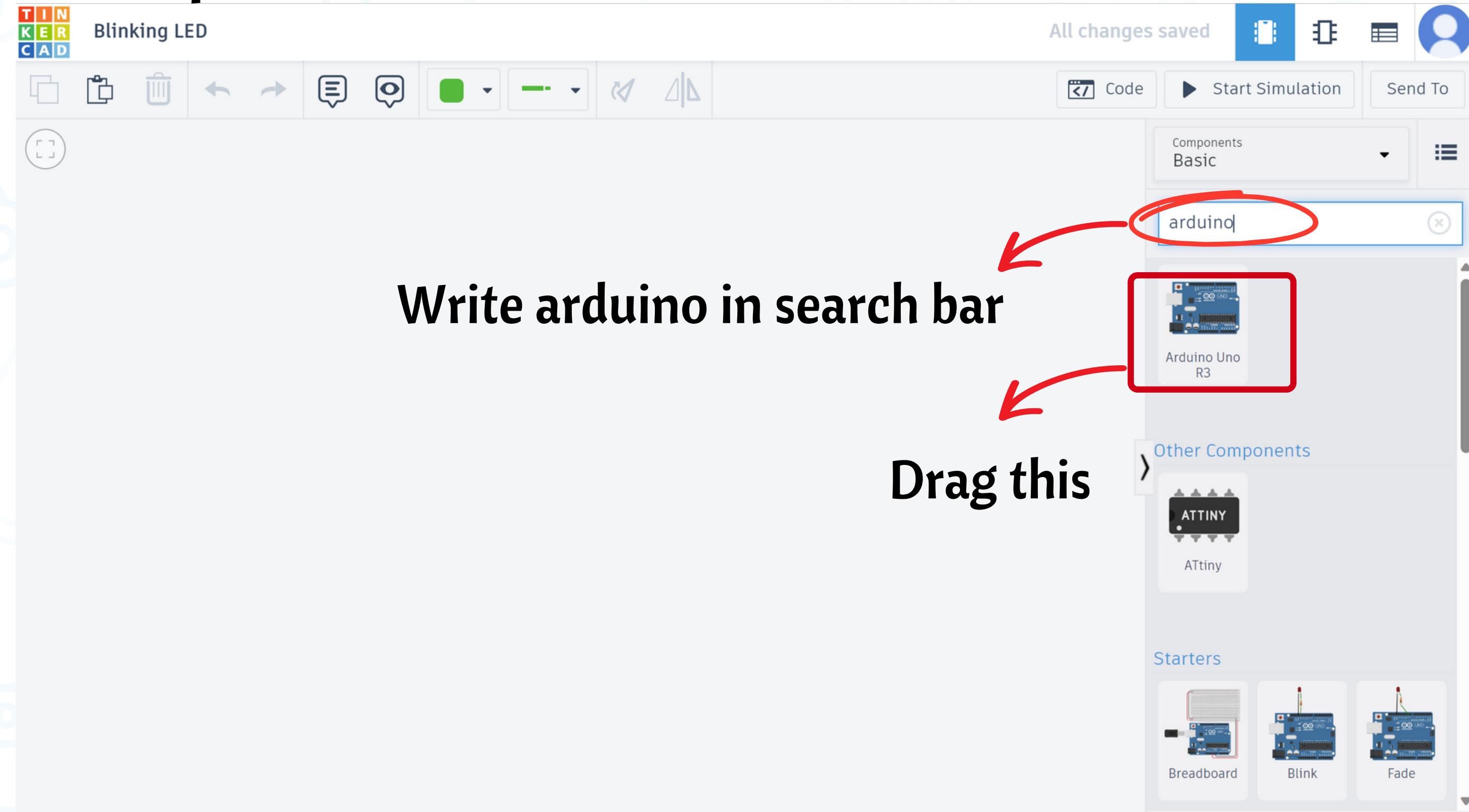
The diagram illustrates a breadboard setup connected to an Arduino Uno. The Arduino is shown on the left, with its pins labeled. On the right, a breadboard is populated with 8 red LEDs. Each LED is connected in series with a resistor. The breadboard has columns labeled a through j and rows labeled 1 through 20. Green lines connect digital pins 2 through 9 on the Arduino to the breadboard. A red line connects pin 10 to ground. The Arduino is connected to a computer via USB.

```
47 digitalWrite(2, HIGH);
48 digitalWrite(2, LOW);
49 digitalWrite(3, HIGH);
50 digitalWrite(3, LOW);
51 digitalWrite(4, HIGH);
52 digitalWrite(4, LOW);
53 delay(500);
54 digitalWrite(5, HIGH);
55 digitalWrite(5, LOW);
56 digitalWrite(6, HIGH);
57 digitalWrite(6, LOW);
58 digitalWrite(7, HIGH);
59 digitalWrite(7, LOW);
60 delay(500);
61 digitalWrite(8, HIGH);
62 digitalWrite(8, LOW);
63 digitalWrite(9, HIGH);
64 digitalWrite(9, LOW);
65 digitalWrite(10, HIGH);
66 digitalWrite(10, LOW);
67 digitalWrite(11, HIGH);
68 digitalWrite(11, LOW);
69 digitalWrite(12, HIGH);
70 }
```

Serial Monitor

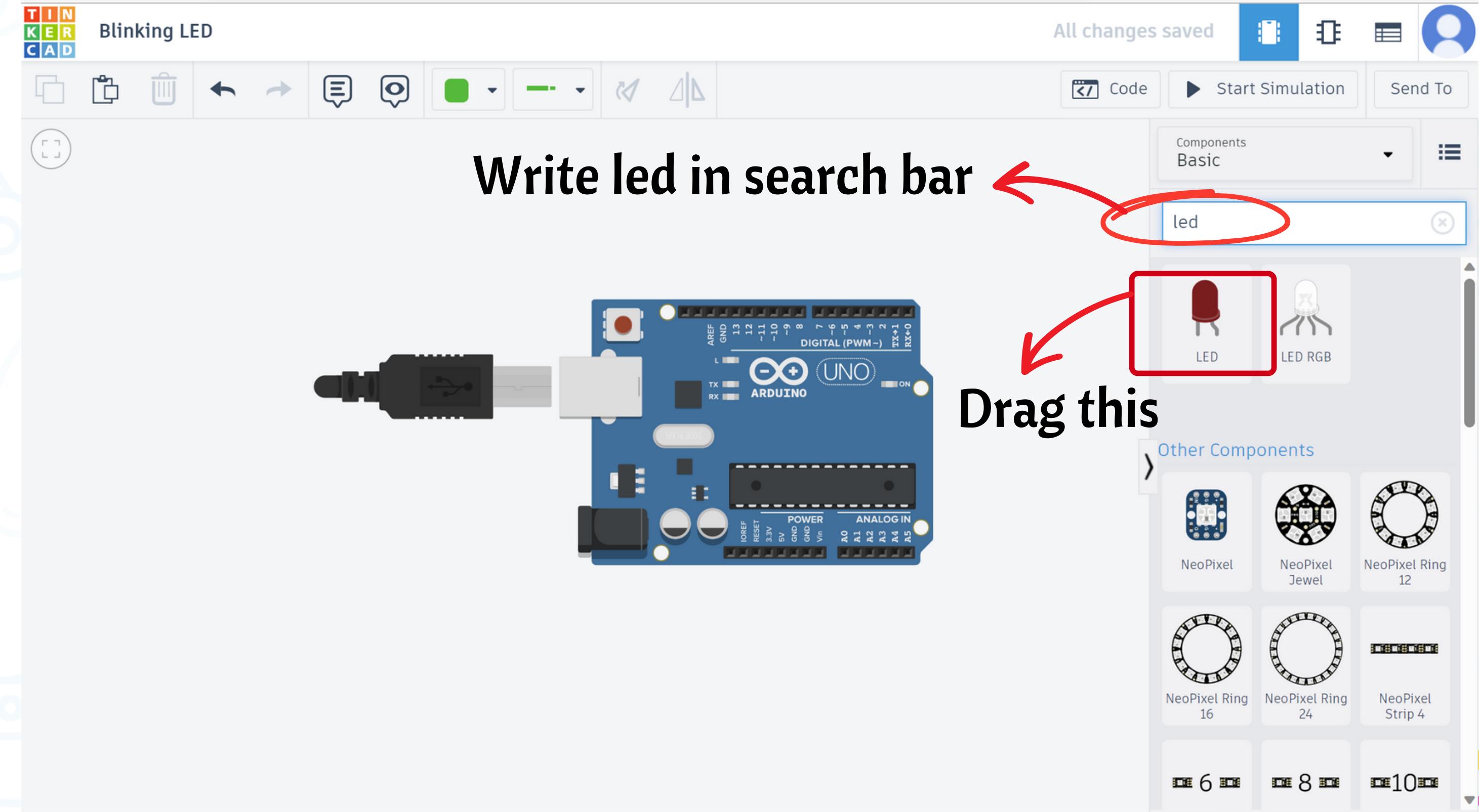
Blinking LEDs

Step 1: components



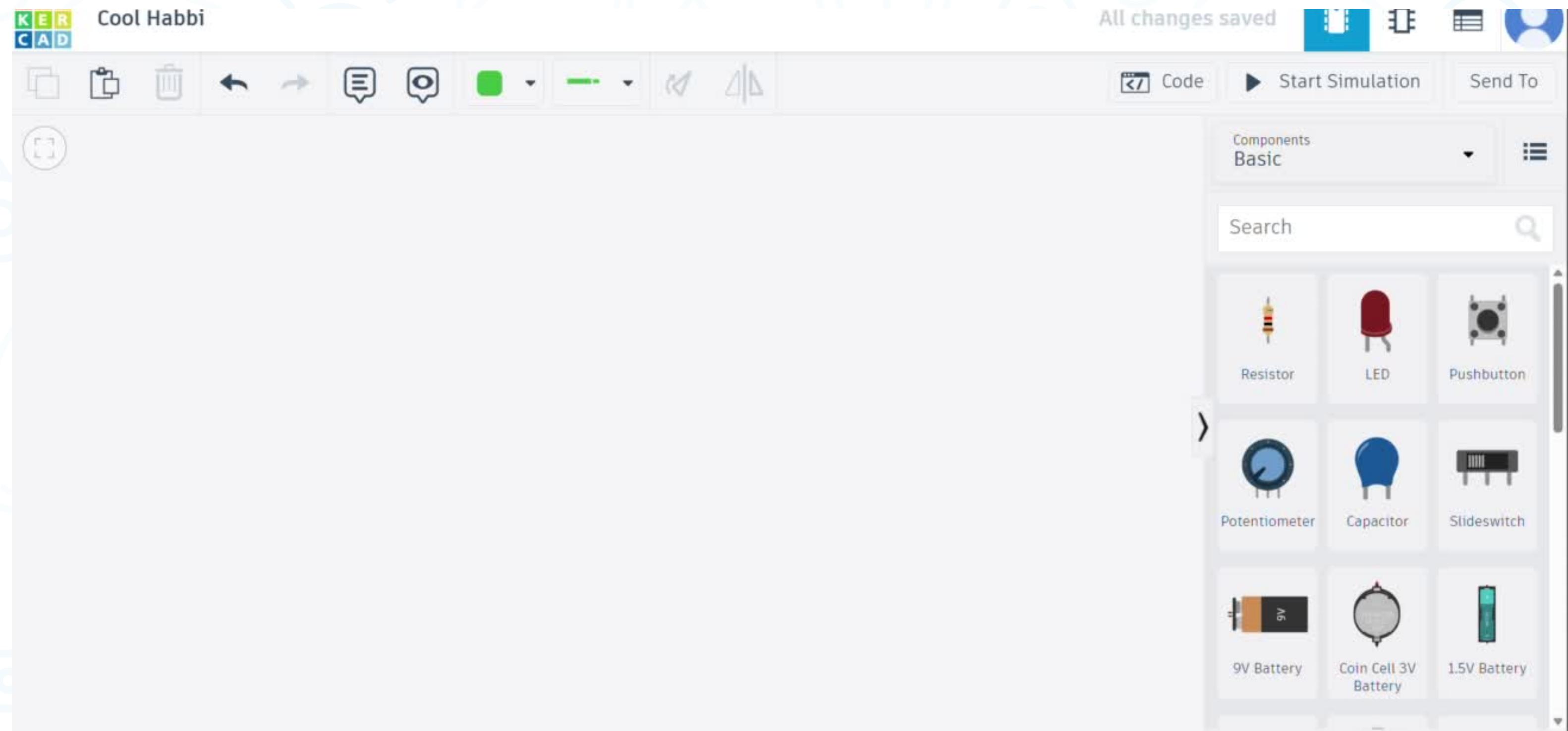
Blinking LEDs

Step 1: components



Blinking LEDs

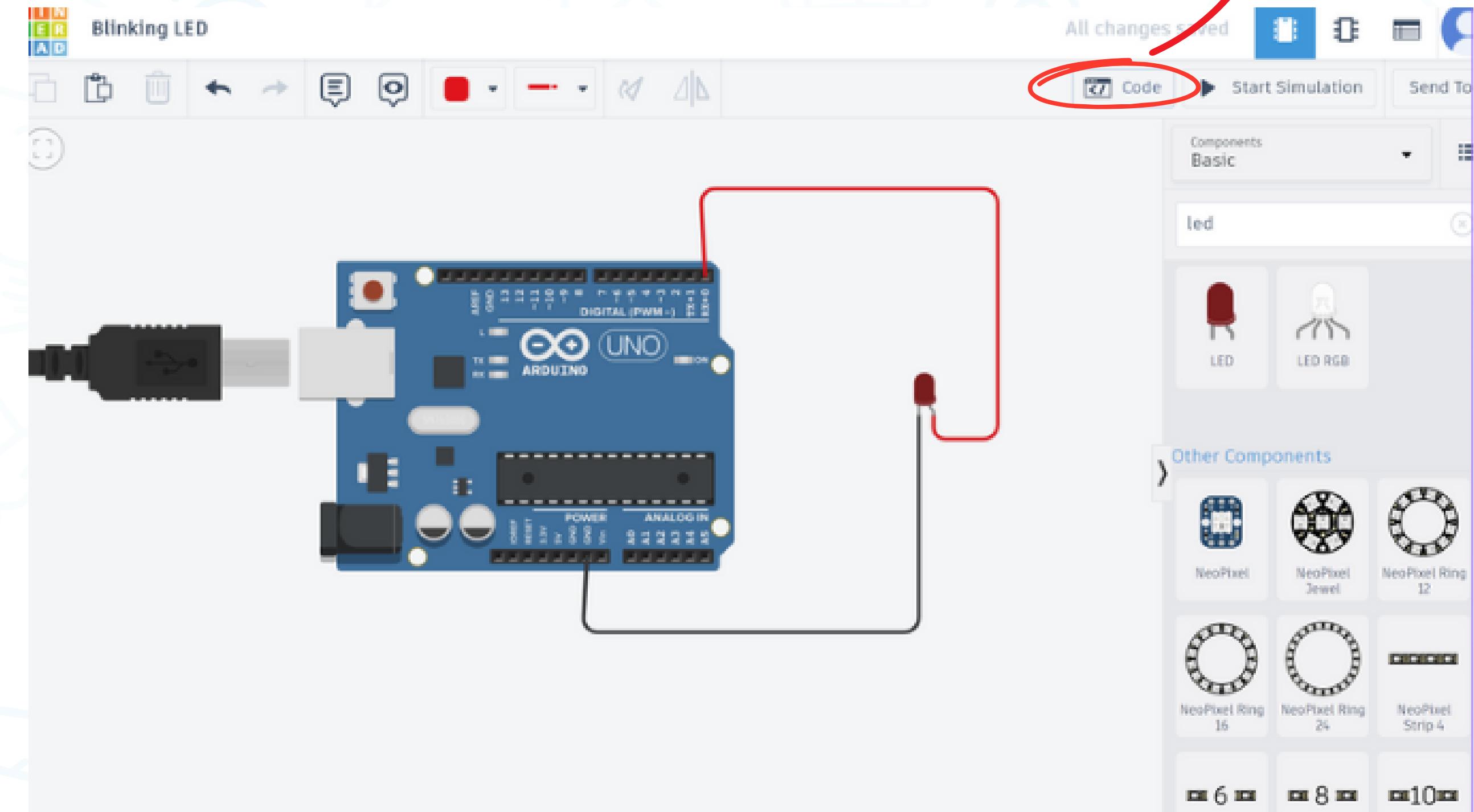
Step 2: Connect components



Blinking LEDs

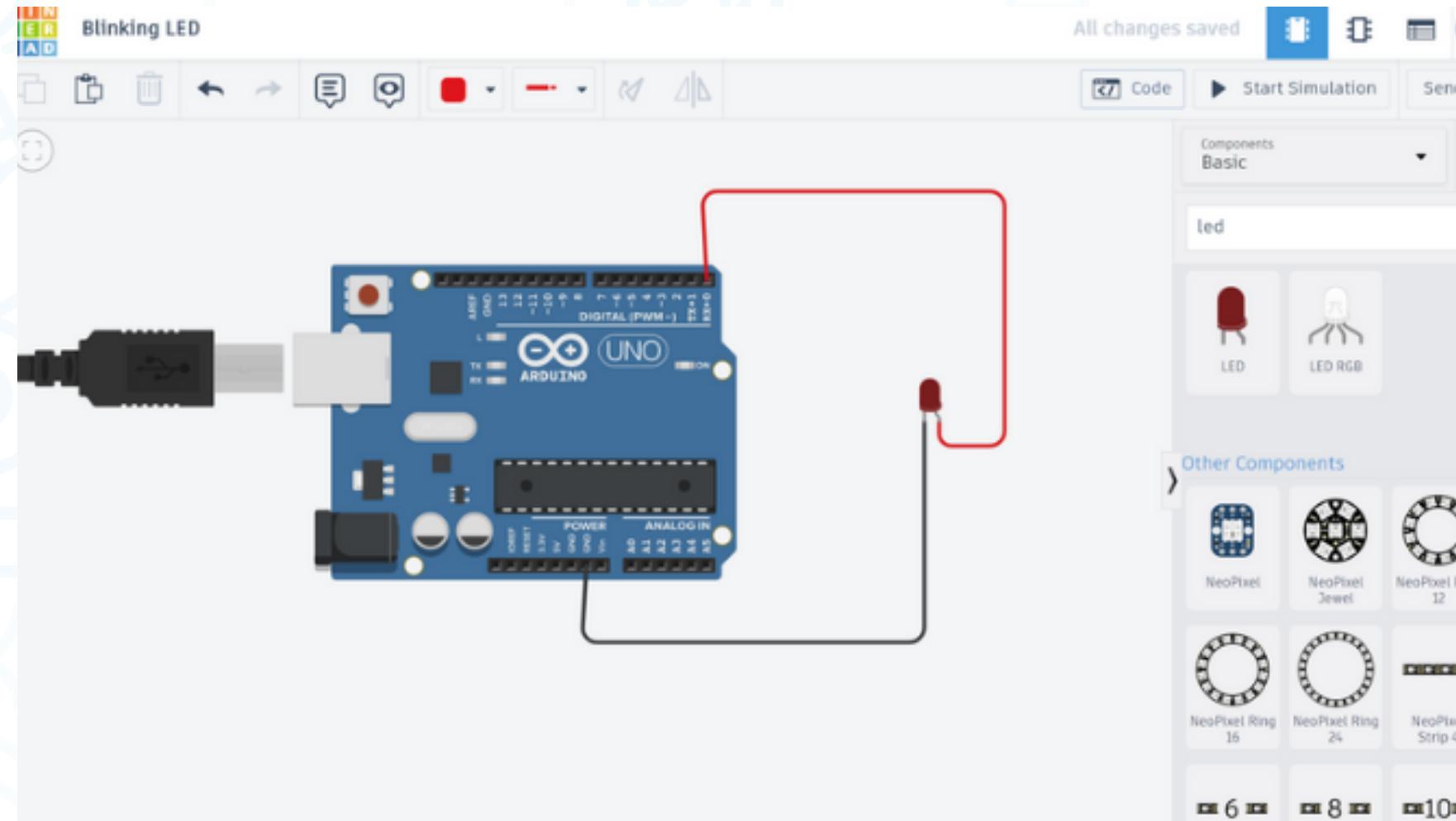
Step 3: Write code to blink one LED

Click here to write code --> text



Blinking LEDs

Step 3: Write code to blink one LED

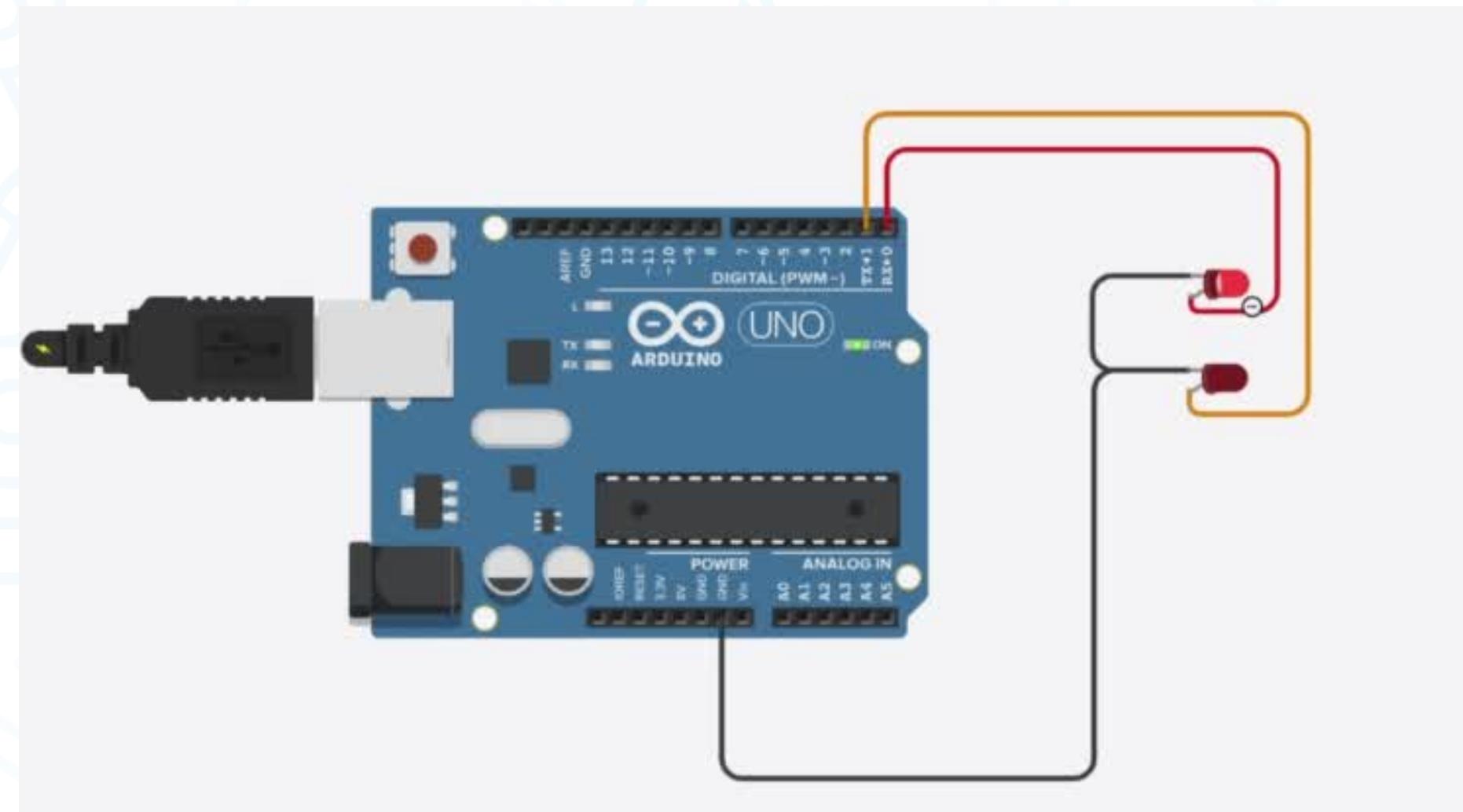


```
void setup()
{
    pinMode(D0,OUTPUT);
}

void loop()
{
    digitalWrite(D0,HIGH);
    delay(500);
    digitalWrite(D0,LOW);
    delay(500);
}
```

Blinking LEDs

Mission 1: Blink 2 LEDs



```
void setup()
{
    pinMode(D0,OUTPUT);
    pinMode(D3,OUTPUT);
}

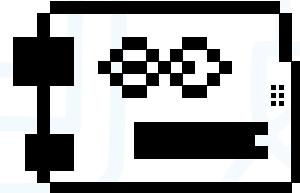
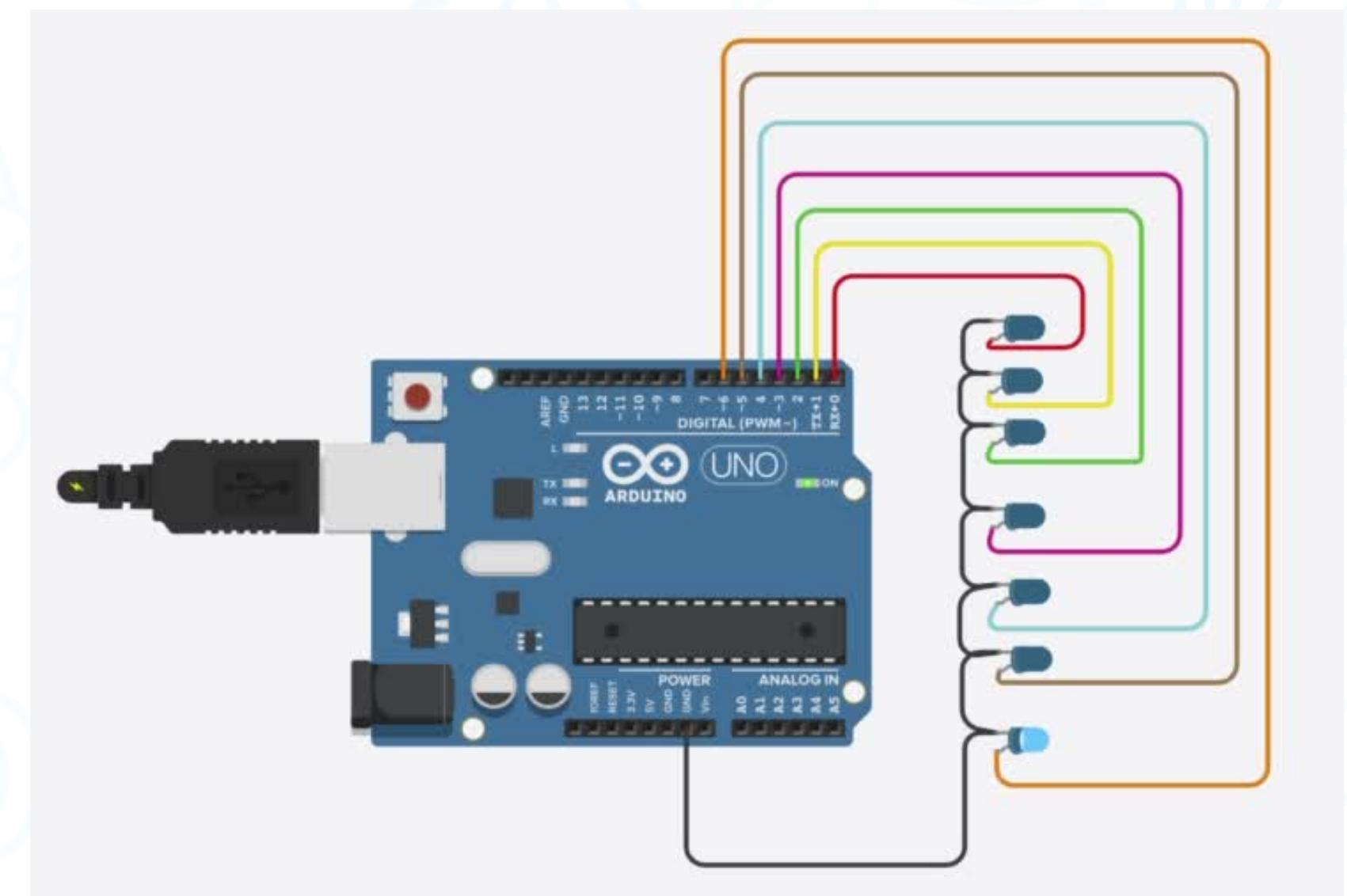
void loop()
{
    digitalWrite(D0,HIGH);
    digitalWrite(D3,LOW);
    delay(500);

    digitalWrite(D2,LOW);
    digitalWrite(D3,HIGH);
    delay(500);
}
```

Let's try it

Try to do the same with the rest of the LEDs

Try it by yourself



Blinking LEDs

```
void setup() {  
    pinMode(D0, OUTPUT);  
    pinMode(D3, OUTPUT);  
    pinMode(D4, OUTPUT);  
    pinMode(D5, OUTPUT);  
    pinMode(D6, OUTPUT);  
    pinMode(D7, OUTPUT);  
    pinMode(D8, OUTPUT);  
  
}
```

Blinking LEDs

Pin Number

```
void setup() {  
    pinMode(D0, OUTPUT);  
    pinMode(D3, OUTPUT);  
    pinMode(D4, OUTPUT);  
    pinMode(D5, OUTPUT);  
    pinMode(D6, OUTPUT);  
    pinMode(D7, OUTPUT);  
    pinMode(D8, OUTPUT);  
}
```

Pin Status

Blinking LEDs

```
void loop() {  
    digitalWrite(D0,HIGH);  
    digitalWrite(D3,LOW);  
    digitalWrite(D4,LOW);  
    digitalWrite(D5,LOW);  
    digitalWrite(D6,LOW);  
    digitalWrite(D7,LOW);  
    digitalWrite(D8,LOW);  
    delay(500);  
    digitalWrite(D0,LOW);  
    digitalWrite(D3,HIGH);  
    digitalWrite(D4,LOW);  
    digitalWrite(D5,LOW);  
    digitalWrite(D6,LOW);  
    digitalWrite(D7,LOW);  
    digitalWrite(D8,LOW);  
    delay(500);  
    digitalWrite(D0,LOW);  
    digitalWrite(D3,LOW);  
    digitalWrite(D4,HIGH);  
    digitalWrite(D5,LOW);  
    digitalWrite(D6,LOW);  
    digitalWrite(D7,LOW);  
    digitalWrite(D8,LOW);  
    delay(500);  
}
```

First LED

Second LED

Third LED

```
digitalWrite(D0,LOW);  
digitalWrite(D3,LOW);  
digitalWrite(D4,LOW);  
digitalWrite(D5,HIGH);  
digitalWrite(D6,LOW);  
digitalWrite(D7,LOW);  
digitalWrite(D8,LOW);  
delay(500);  
digitalWrite(D0,LOW);  
digitalWrite(D3,LOW);  
digitalWrite(D4,LOW);  
digitalWrite(D5,LOW);  
digitalWrite(D6,HIGH);  
digitalWrite(D7,LOW);  
digitalWrite(D8,LOW);  
delay(500);  
digitalWrite(D0,LOW);  
digitalWrite(D3,LOW);  
digitalWrite(D4,LOW);  
digitalWrite(D5,LOW);  
digitalWrite(D6,LOW);  
digitalWrite(D7,HIGH);  
digitalWrite(D8,LOW);  
delay(500);
```

Fourth LED

Fifth LED

Sixth LED

```
digitalWrite(D0,LOW);  
digitalWrite(D3,LOW);  
digitalWrite(D4,LOW);  
digitalWrite(D5,LOW);  
digitalWrite(D6,LOW);  
digitalWrite(D7,LOW);  
digitalWrite(D8,HIGH);  
delay(500);
```

Seventh LED

