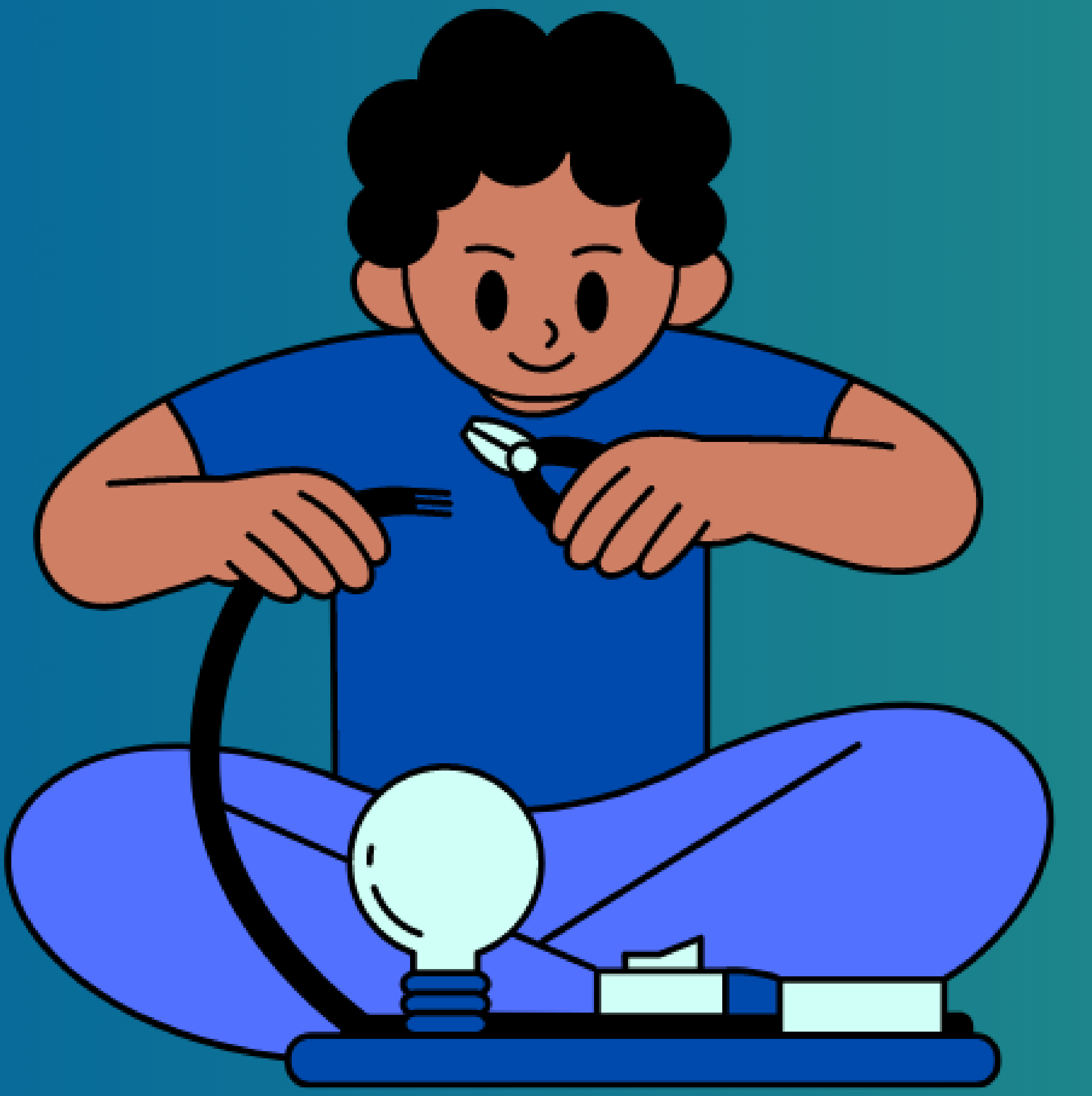
UNLOCKING MICROCONTROLLERS



Shashwat Jaguri & Sebastian Inestroza

from Coduriosity
www.coduriosity.com

Preface

Coduriosity was founded with the goal of bringing education to children who don't have the resources or support to pursue STEM. This book aims to not only helps bridge that gap by not only helping the children in those underprivileged schools but helping anyone who picks this book up to start their journey into robotics.

Through this book "Unlocking Microcontrollers" I hope to demystify and make complex concepts more accessible to young learners - such as those that I taught my classrooms in different rural schools - with a fun and engaging introduction to this field.

I would like to express my deepest gratitude to my students, whose boundless curiosity and enthusiasm have fueled my passion for teaching. I am also indebted to my fellow teammates who have through the experiences with me, helping me refine the ideas presented in this book.

I invite you, young inventors, to dive into the pages of this book with an open mind and a heart full of curiosity.

- Shashwat Jaguri (Founder)

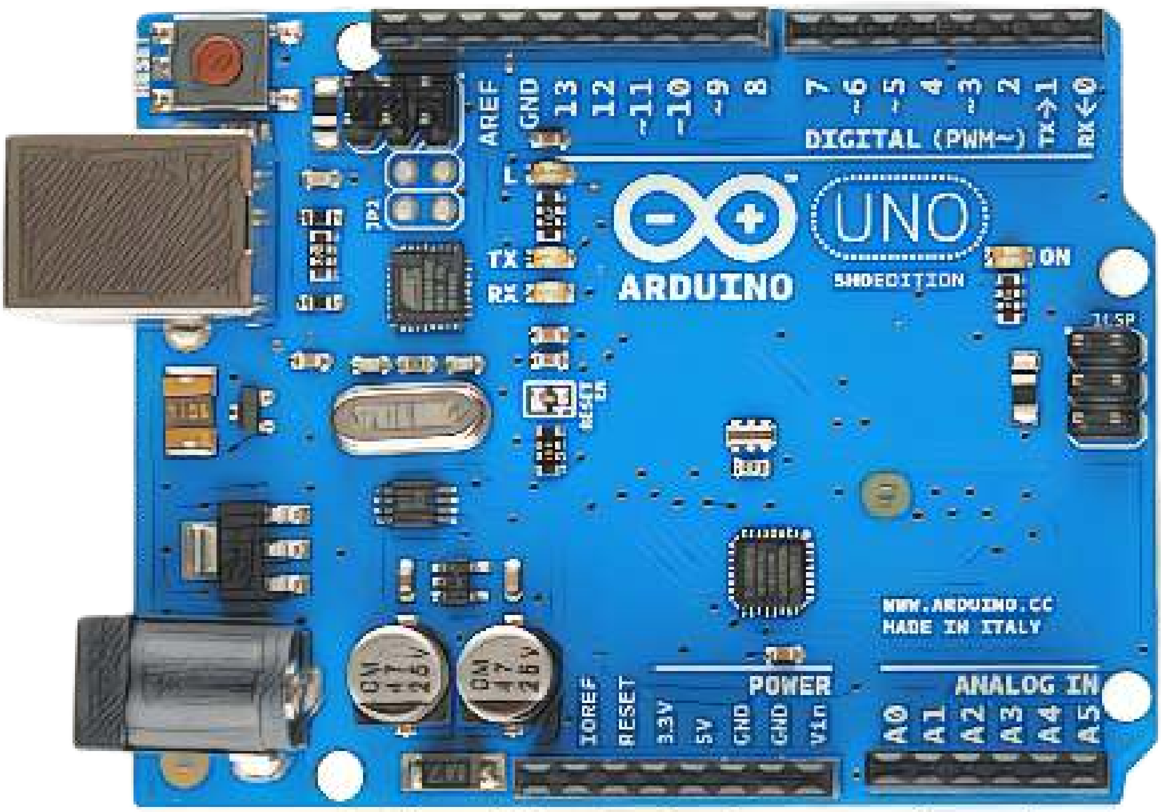
You have a **lightbulb**

You want it to **turn on by itself** during night and off during the day

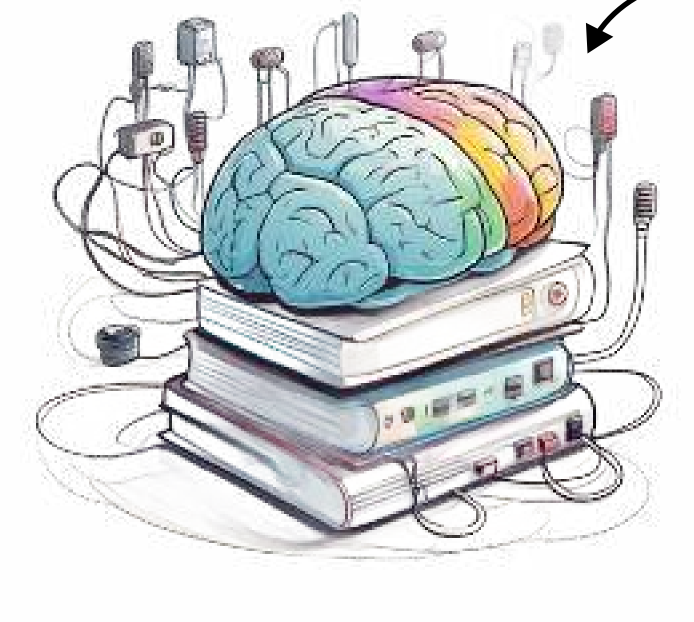
But how do you do that?

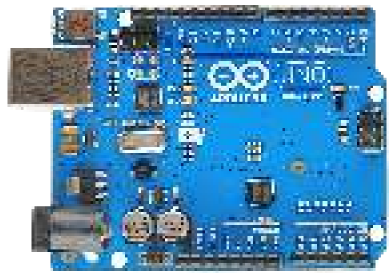
You need a..

MICROCONTROLLER!



A microcontroller is like a **tiny brain for electronics.**





Its a smart chip that can follow **instructions** to do cool things.

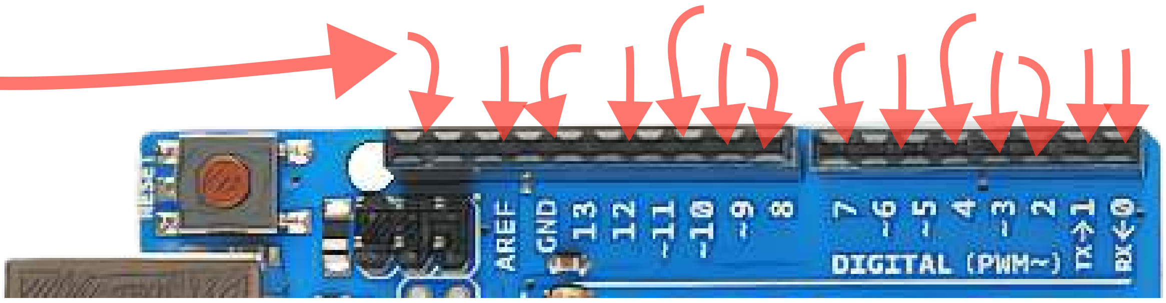
A microcontroller in a RC (Remote Controlled) car **tells the car how to move.**

Microcontrollers are used **everywhere**, for example, in a washing machine or a TV and its remote.

But not just that, **microcontrollers can make circuits do whatever you want!**

Arduino is a board that uses microcontrollers to bring your ideas to life.

It has **pins** that allow you to easily connect wires, buttons, sensors, lights and motors which with a little bit of **coding** you can control.

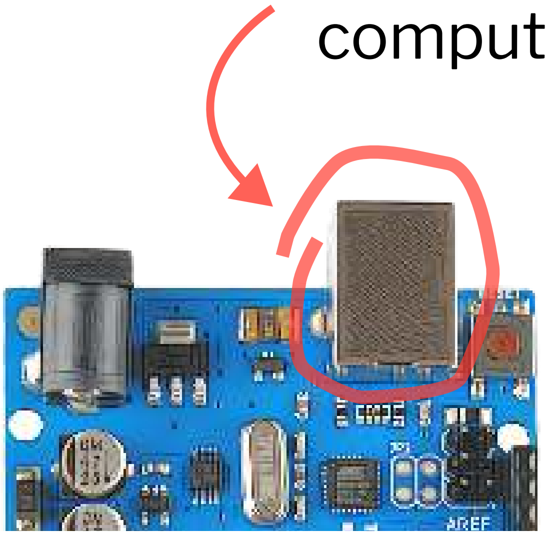


so many pins!!!

Coding is how we give **instructions** to the board in a way it can understand

this is how to connect the microcontroller to a laptop

The **USB port** is what we use to **talk to the board** using another computer



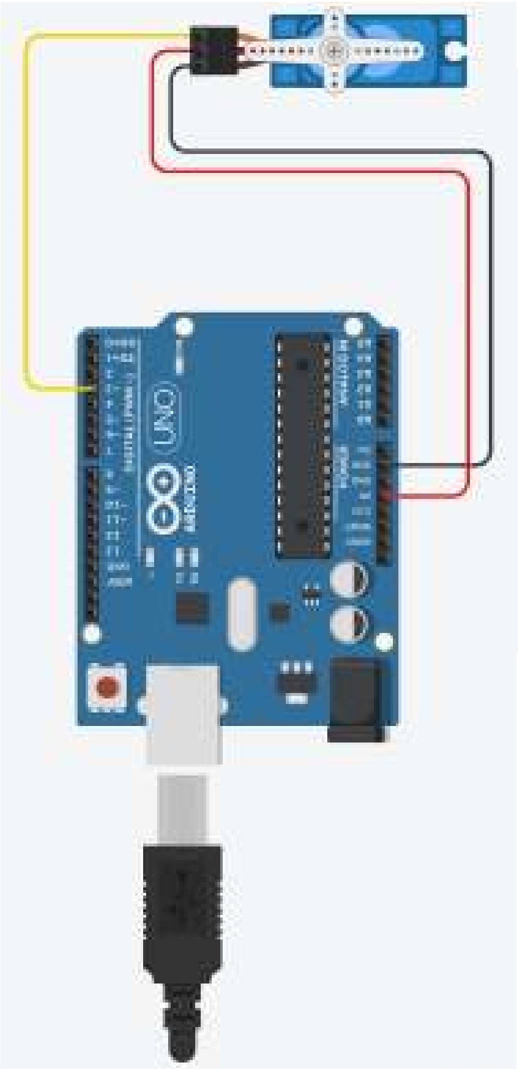
We can tell it **what do** with **each pin** using our **code** by sending it through the **usb**

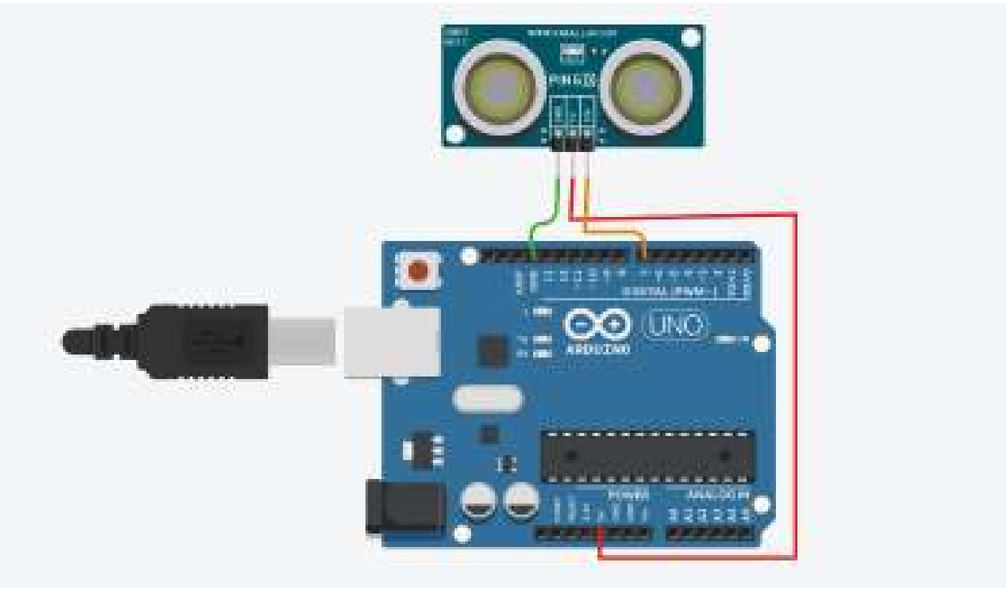
The microcontroller **knows where each pin is** and can **control** the circuit to turn it **on** and **off**

Just how you remember things, the **microcontroller can remember its code**

How does this work? Let’s think about the **circuit** as a team trying to do a task. !

The microcontroller is the **boss** and it talks to its team members - **sensors** and **actuators** - to make things happen



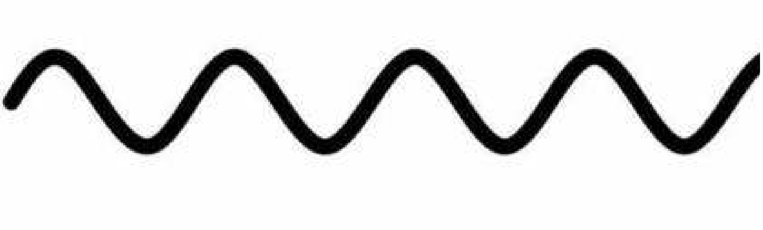


These **sensors** and **actuators** can be **connected** to the microcontroller **using wires**

Sensors are like the **eyes** and **ears** of the microcontroller. Depending on the sensor the microcontroller is able to **sense** things like **light, temperature and movement.**

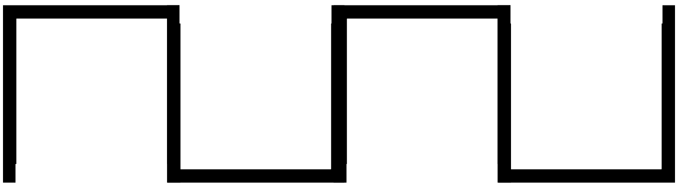
SENSORS

There are two types of sensors, **Analog** and **Digital**. The difference between them is that analog measures a flow of information that changes, while digital measures a set of steps or levels.

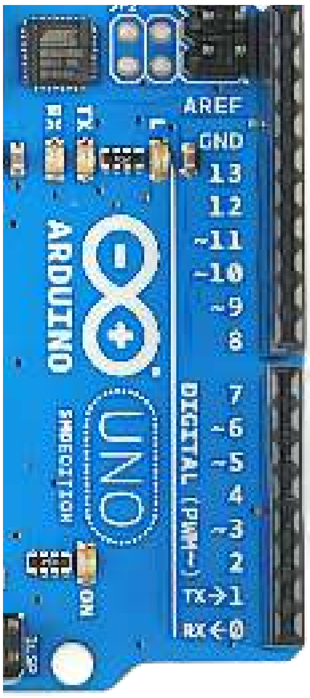


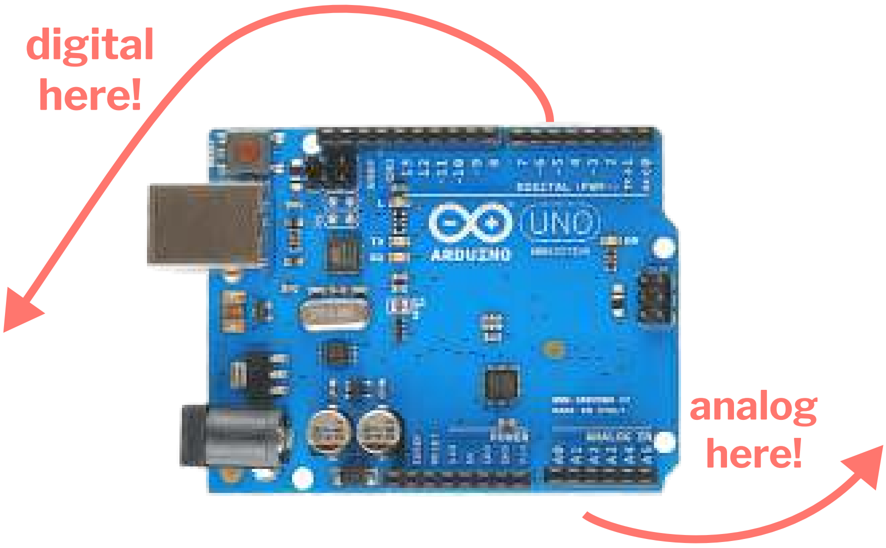
Analog Signals are **smooth** and chan show how a value changes over time

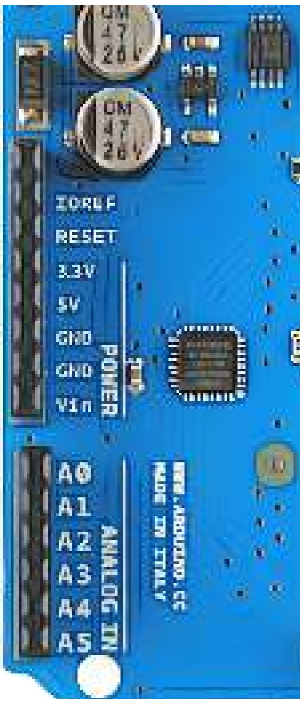
Digital signals can **only** be one state at a time either **high or low**



Since these **sensors** use **different languages** to communicate they need to be put in **different pins** to let the microcontroller know **how** to talk to them







ACTUATORS

Actuators allow our code to interact with the real world. Sensors let us know what is happenig, the microcontroller does the thinking, but the actuators do the action.

Some examples of actuators are

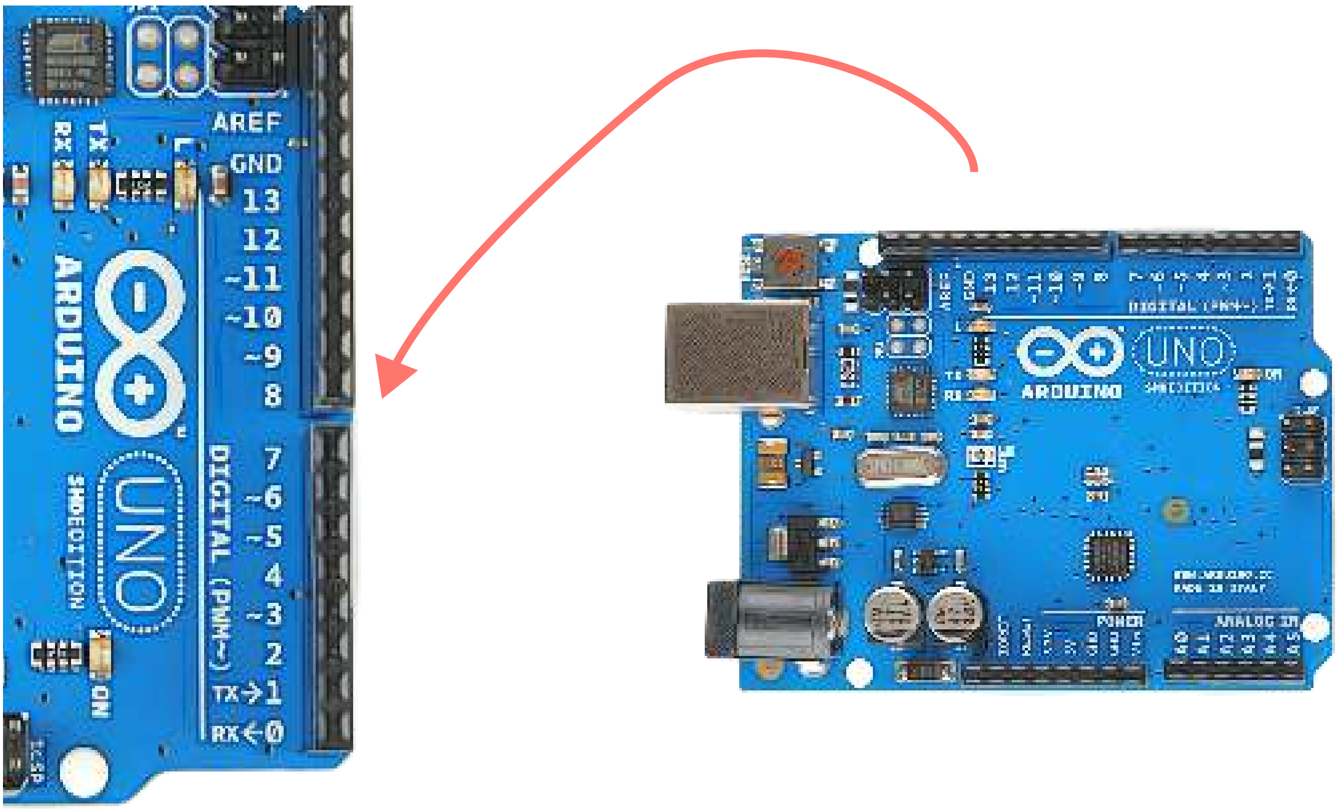
servos

leds

motors

These devices all convert electrical energy into something physicals like light or movement

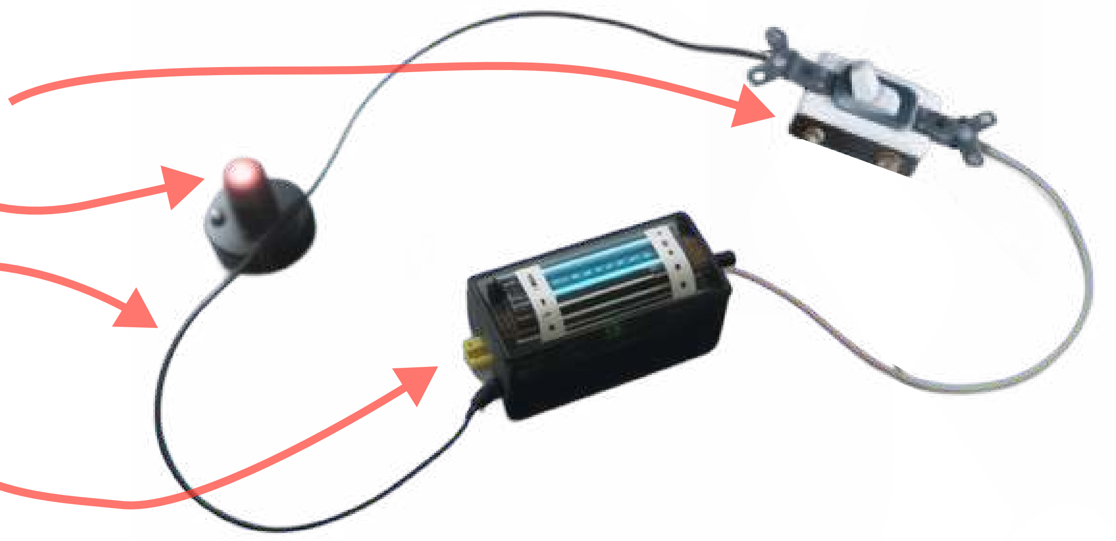
All actuators are controlled digitally through these pins



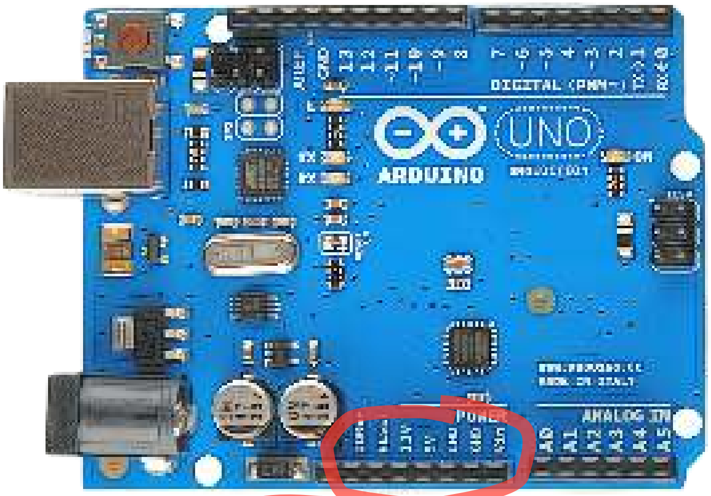
Do you remember how a circuit is made?

What are the components we need?

- switch
- bulb
- wires
- battery



Let’s see how to make a similar circuit using aurdino

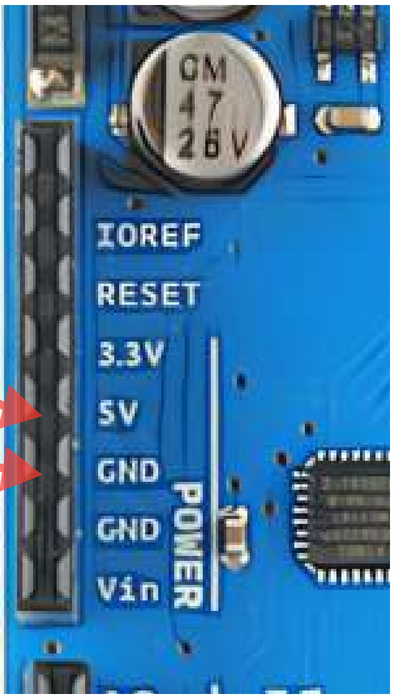


The the **-** of your battery is what is labled as **GND**

The **+** is the same as the **3.3v**, **5v** or **Vin** pins.

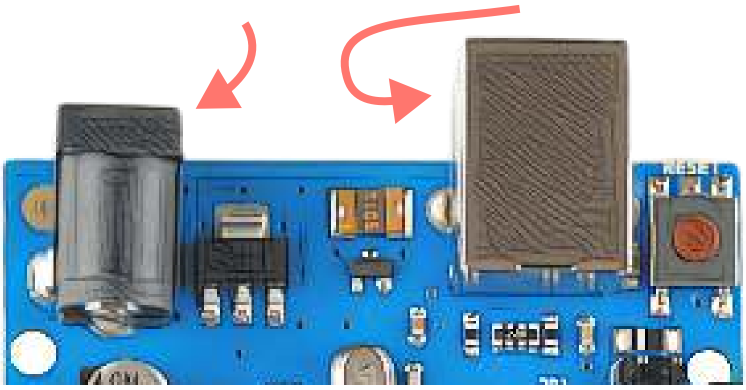
Different batteries have different voltages, the symbol **V** is used to represent voltage.

This large battery has a voltage of **9v**

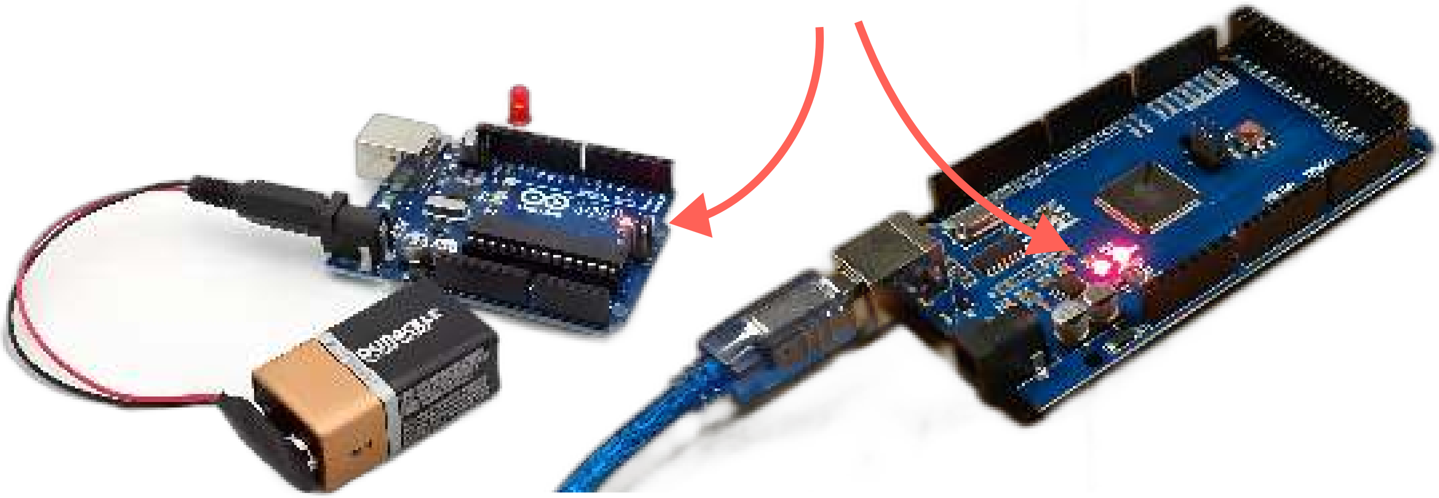


This battery has a voltage of **1.5v**

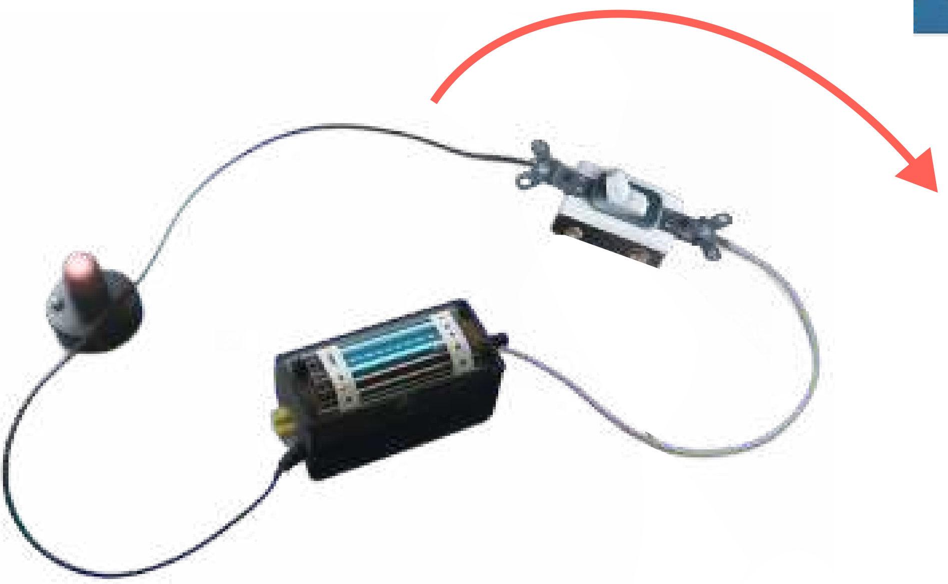
How does the microcontroller get it's power? We can either use the **Barrell** or **USB** connector.

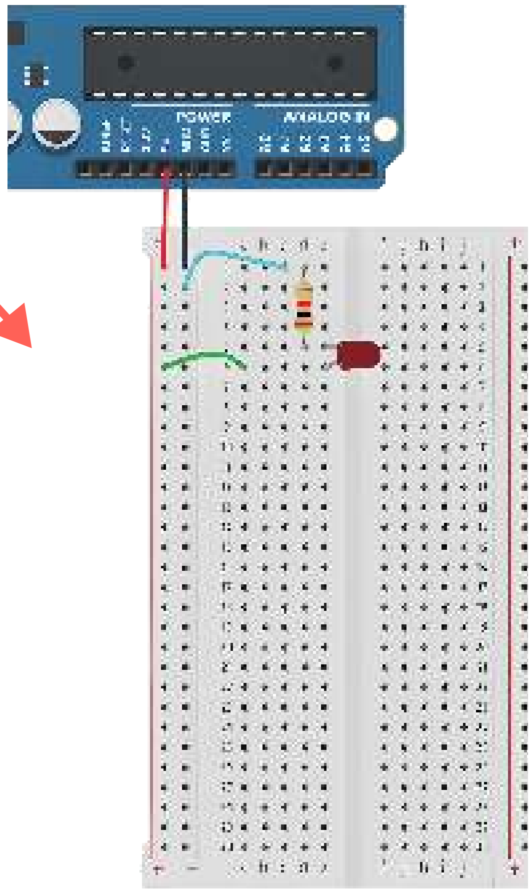


Once the board has power, some of it's LEDs will light up and start to flash



This is what a this circuit would look like with an microcontroller

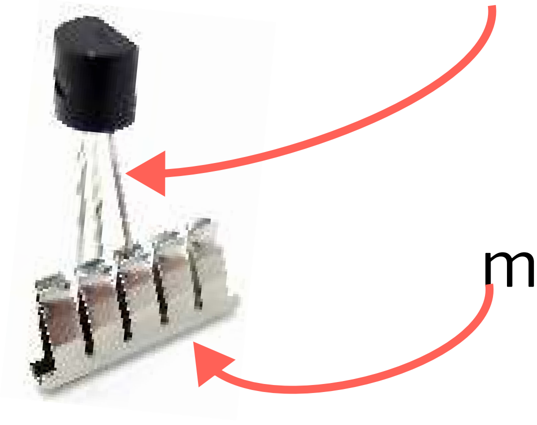




BREADBOARDS

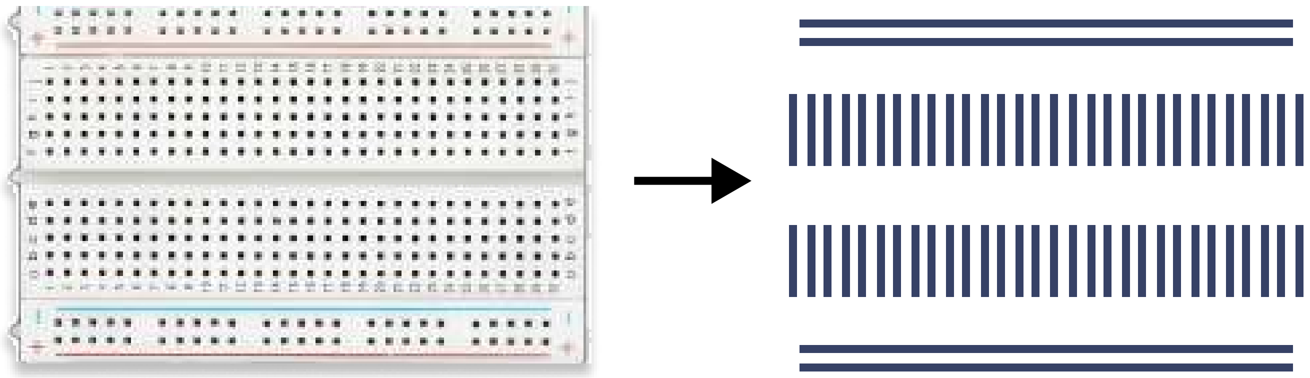
A breadboard **helps** us in **making circuits**. It has lots of holes which allow us to connect and rearrange wires **fast** as **easily**.

Most electronic components has **long metal legs** for connection **called leads** which can fit into a breadboard easily



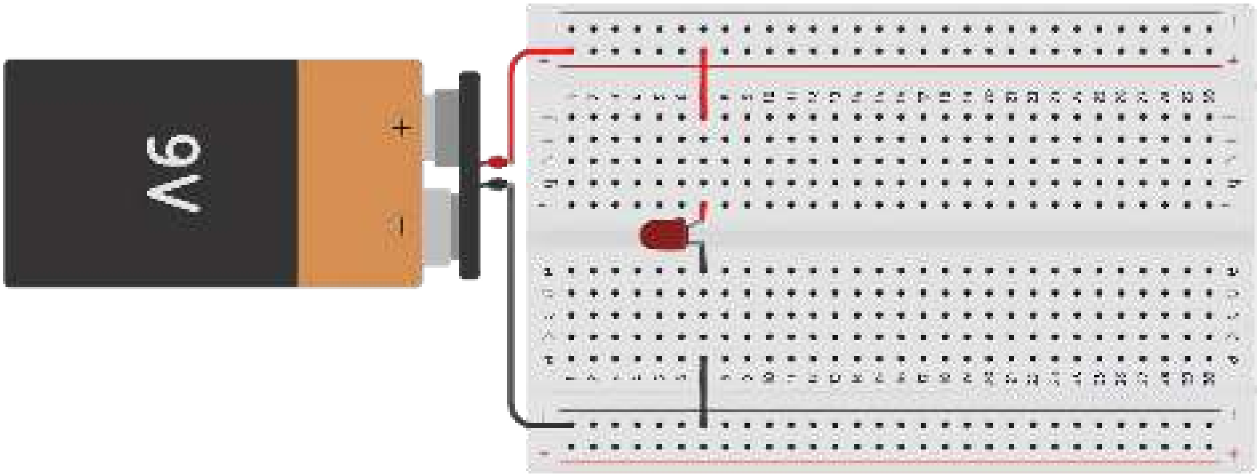
breadboards have tiny metal clips which can hold onto the components

Lets first look at the layout of a breadboard



This is how the pins are laid out. Usually a battery is connected to the - and + pins and then wires are connected on the same length to power components on the breadboard

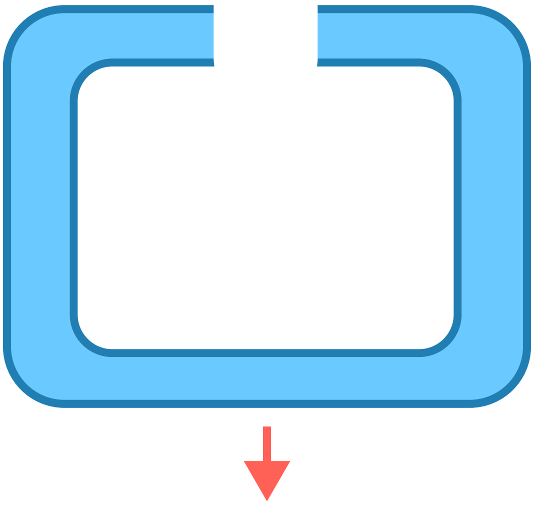
This is how we would create a simple bulb circuit using the breadboard



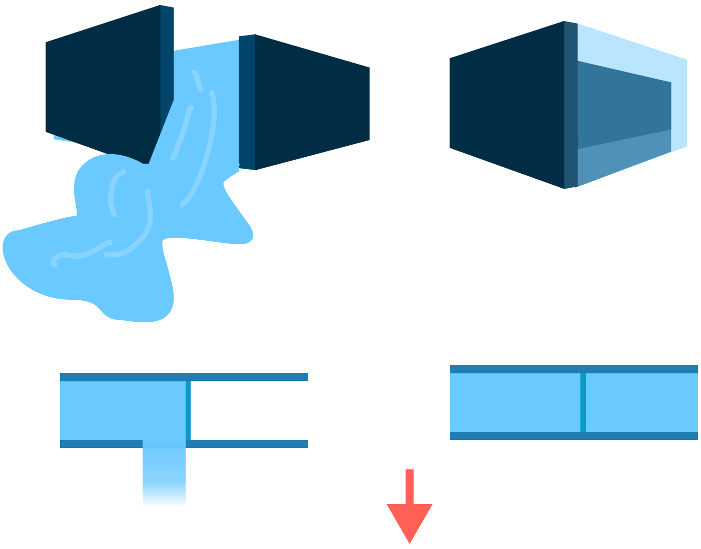
The most important thing is to make sure that the circuit is connected properly.

- if there is a break then electricity won’t flow
- if there is a path with less components (resistance) then the electricity will go there

You can think of the electricity as water.



If there is a break in the pipe then the water wont go where you want it to

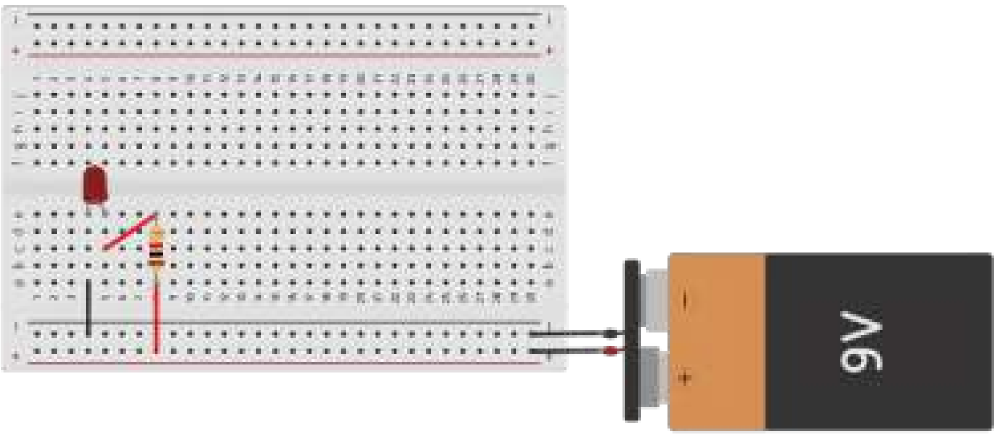


If there is somewhere for the water to leak too, it wont go where we want it to

Circuits do not need to be exactly the same as in the diagram.

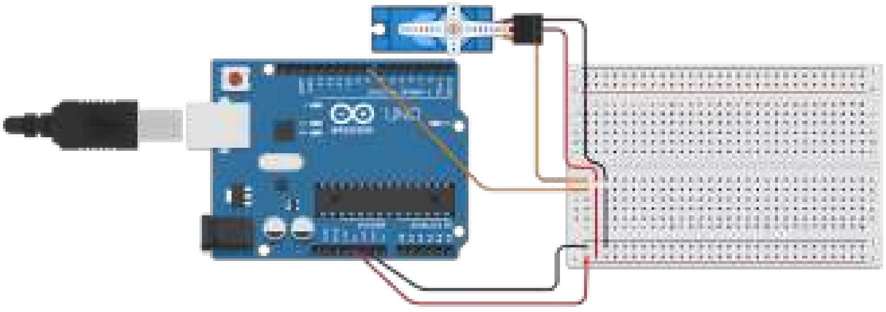
As long as there and no breaks, no alternate paths for the electricity to go through and all pins are connected to the right port

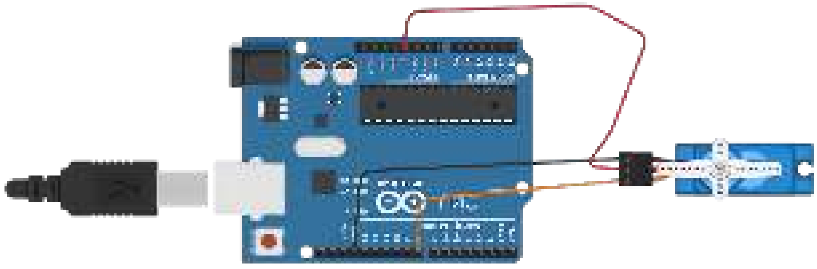
This means that the wire connected to a digital port should go to the same place. Similarly a wire connected to plus should go to the same place



This circuit is same as the one in the previous diagram

These both are also ways that you can connect the servo, see how the **black wire always goes to GND,** the **red always to 5v** and **yellow always to D8**



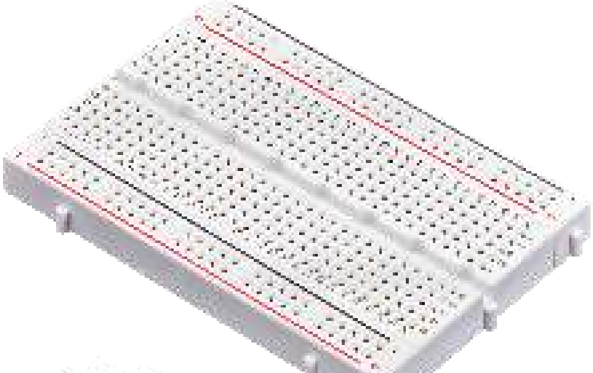


How to make a light flash

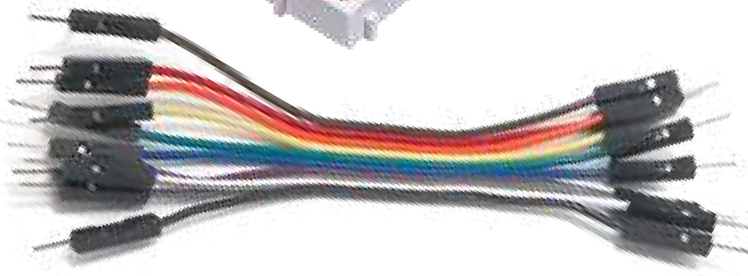
Step 1:

Gather your tools,

- Breadboard
  - makes connecting wires easier

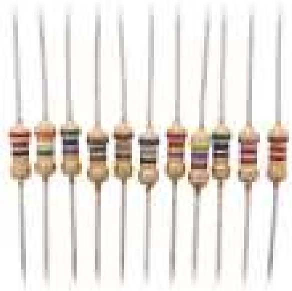


- Jumper Wires
  - wires with metal pins at the end to make connecting easier



- Arduino (Microcontroller)
  - to give the circuit instructions

- Resistor
  - to limit the current that is reaching our bulb

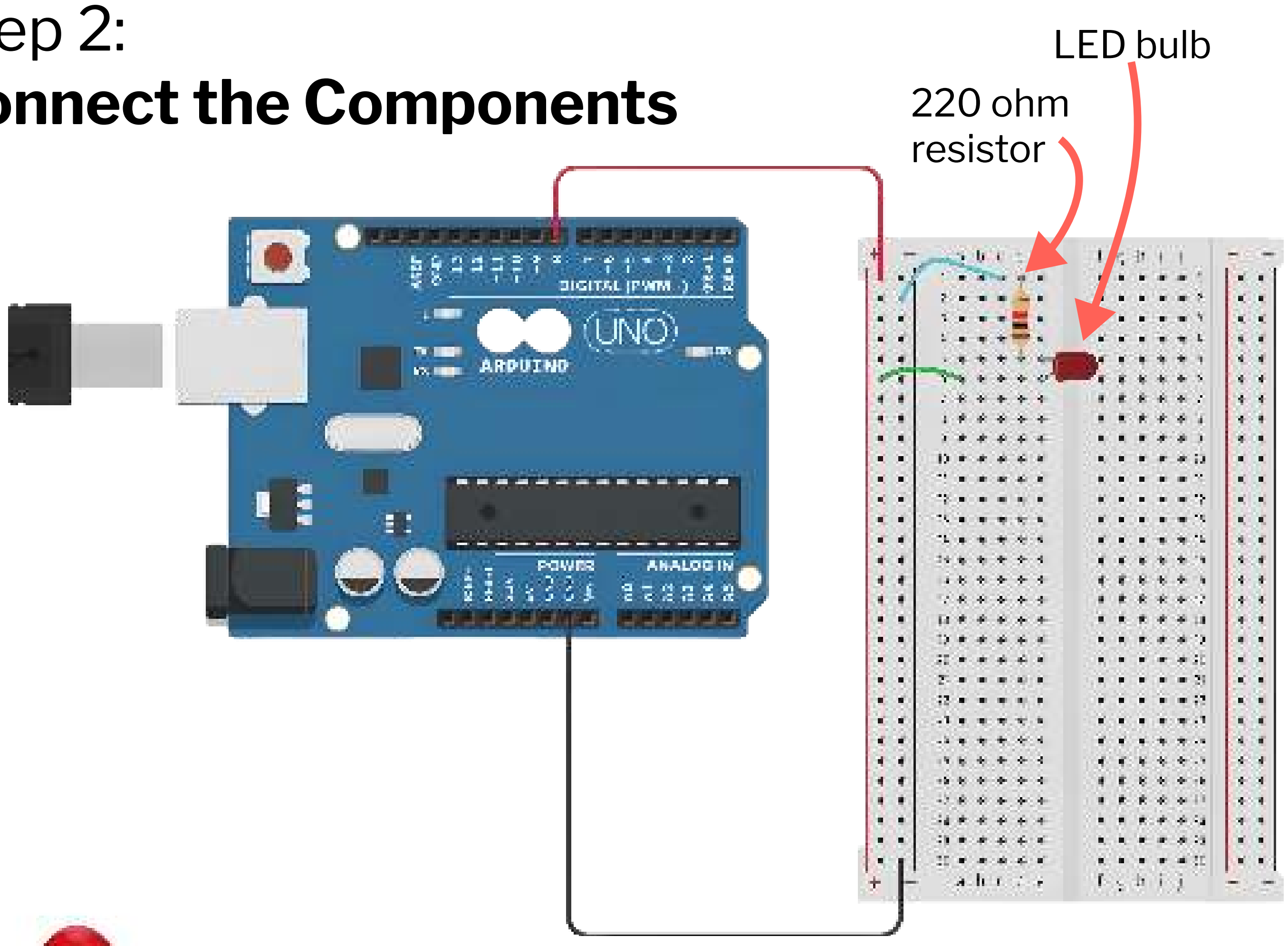


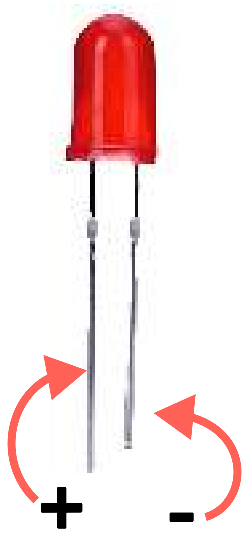
- LED blub
  - Our Light Source

- USB type B cable
  - to program the Arduino and provide power

Step 2:

Connect the Components





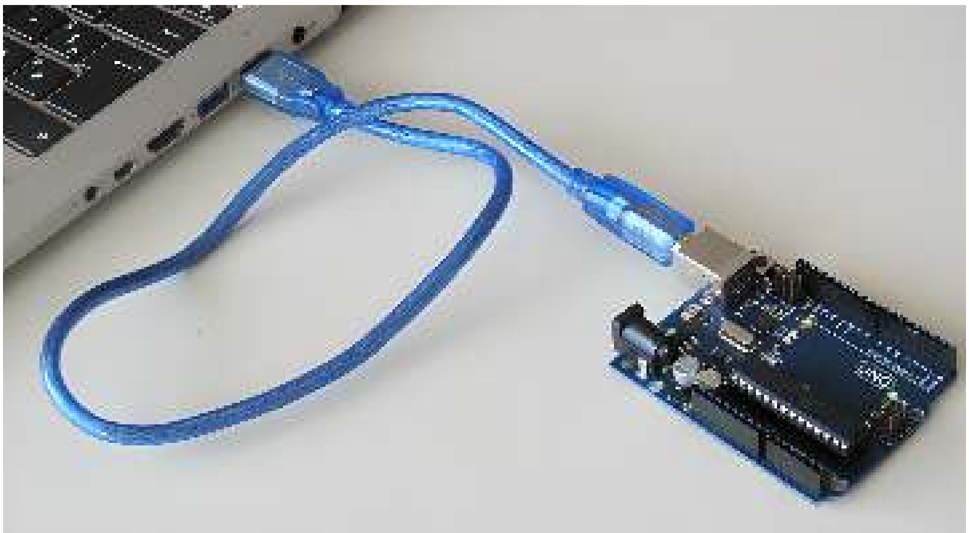
You will notice that the LED has one side which is longer than the other. LEDs only let **electricity flow** through in one direction. This means that the **long side always has to go to the +** and the **short side always to the -**

We connect the + side of the pin to the Digital Pins. Over here **the LED is connected to pin 8.**

The microcontroller can be told which pin the LED is in and what to do with it.

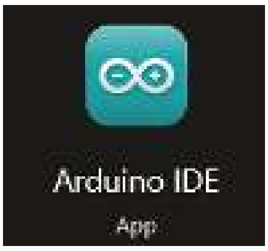
As you remember GND means -, so the short side of the LED is connected to the - pin

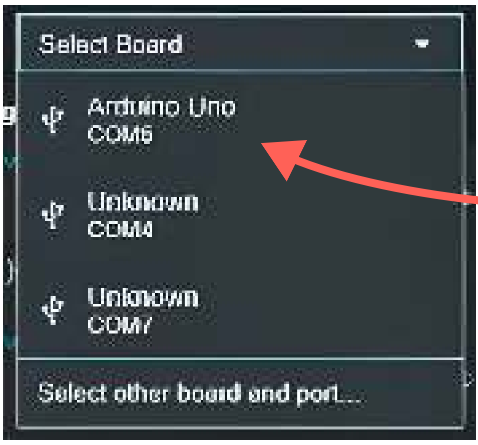
Step 3:
Coding/Programming



Make sure the microcontroller is **connected to** the **computer** and then open up the Aurdino IDE

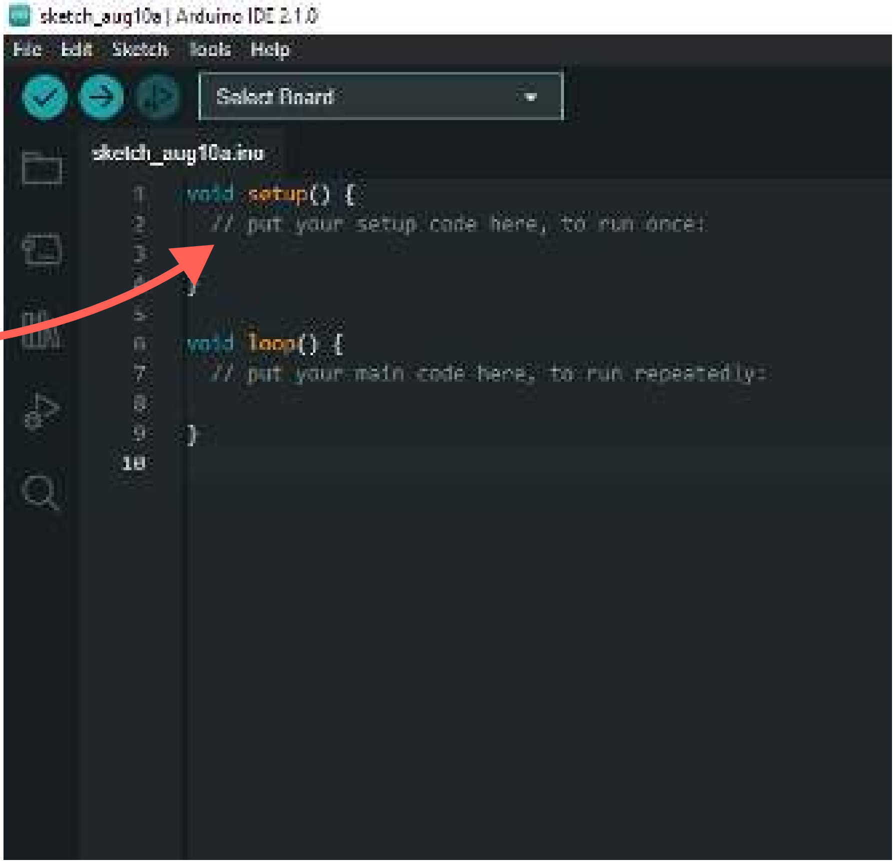
Arduino IDE is the app that we will use to **send our code**. It’s icon looks like this





Select the microcontroller

And write your code here!!


sketch_aug10a | Arduino IDE 2.1.0
File Edit Sketch Tools Help
Select Board
sketch_aug10a.ino

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

}

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

}
```

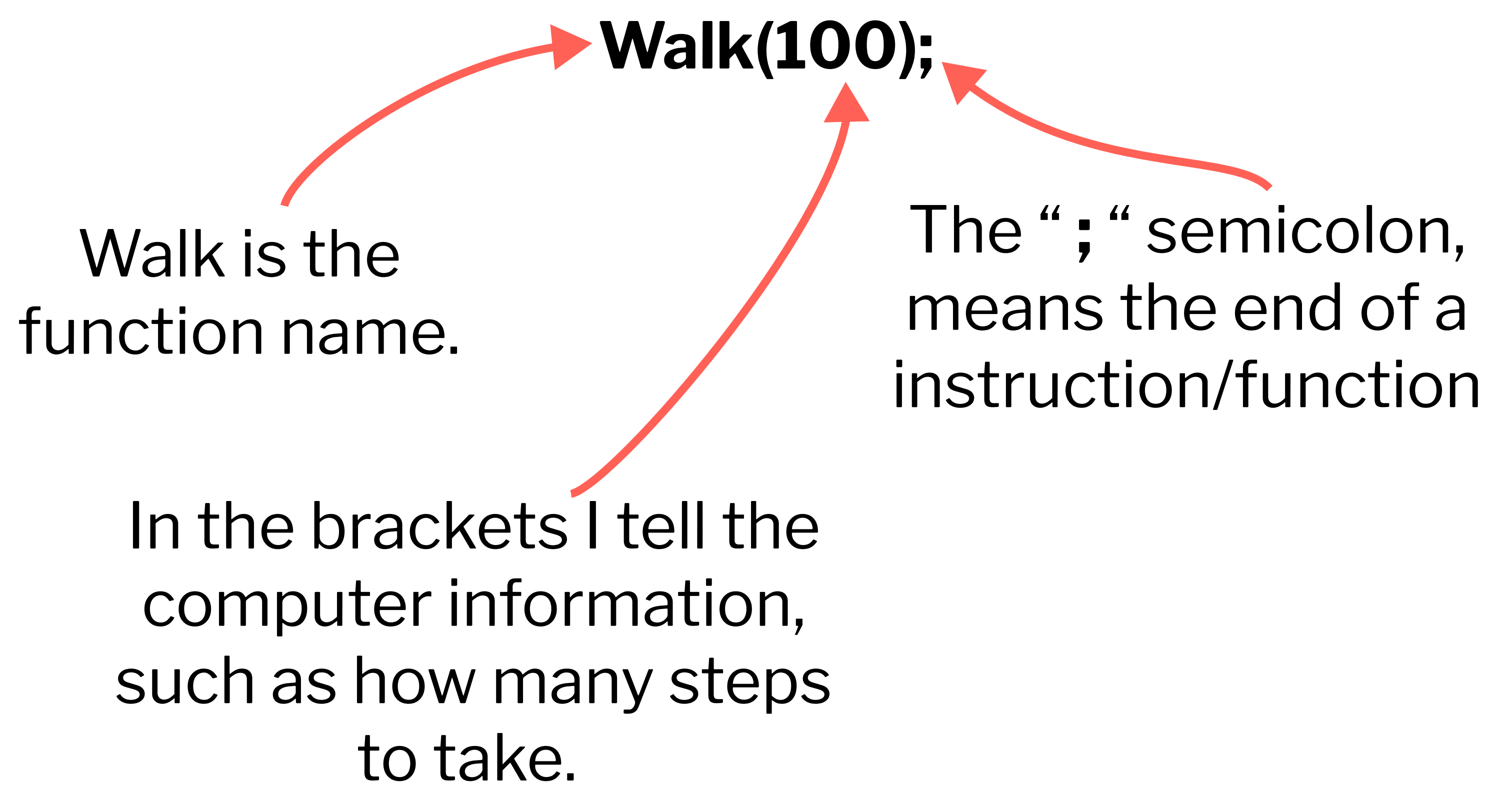
Make sure the code is written exactly like the example or there might be errors

Step 3:
Coding/Programming

Functions are how we tell computers what to do. It is a set of instructions.

If I want you to the other end of the room, I can give an instruction like. Walk 100 steps in that direction

Similarly I would tell a computer

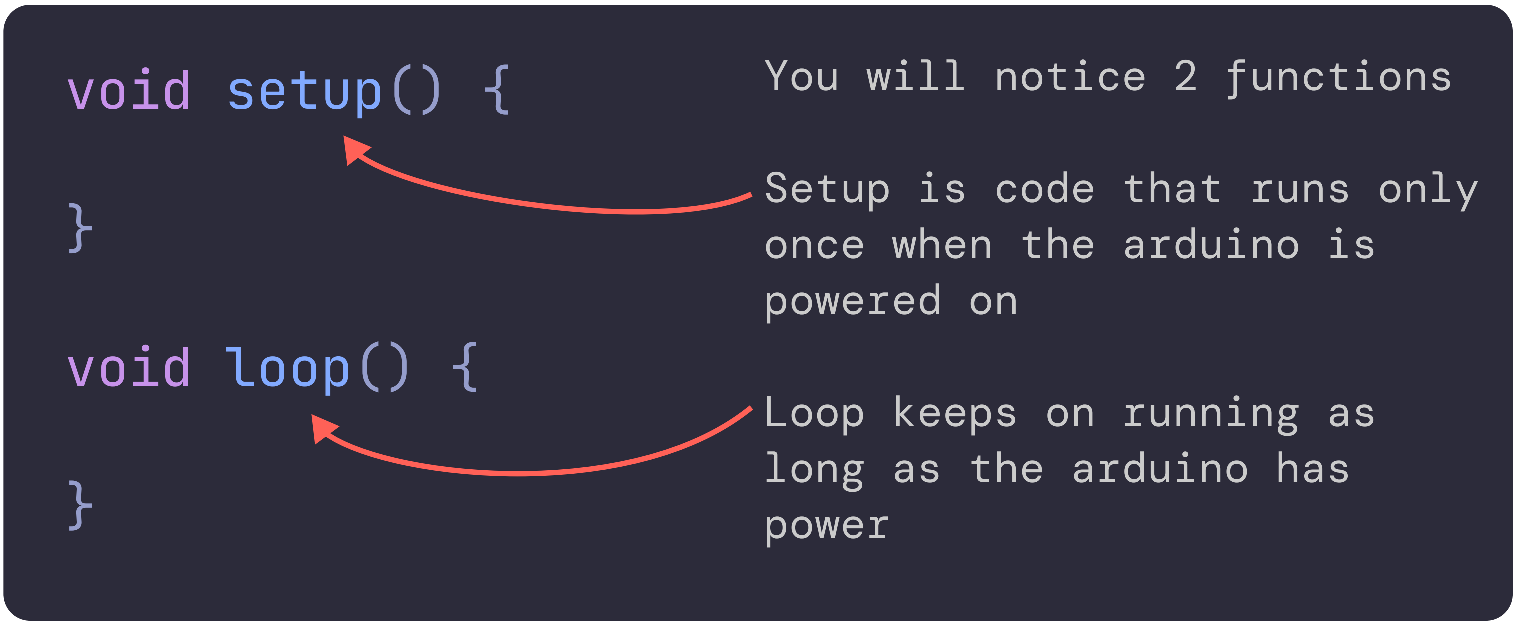


Walk(100);

Walk is the function name.

The " **;** " semicolon, means the end of a instruction/function

In the brackets I tell the computer information, such as how many steps to take.



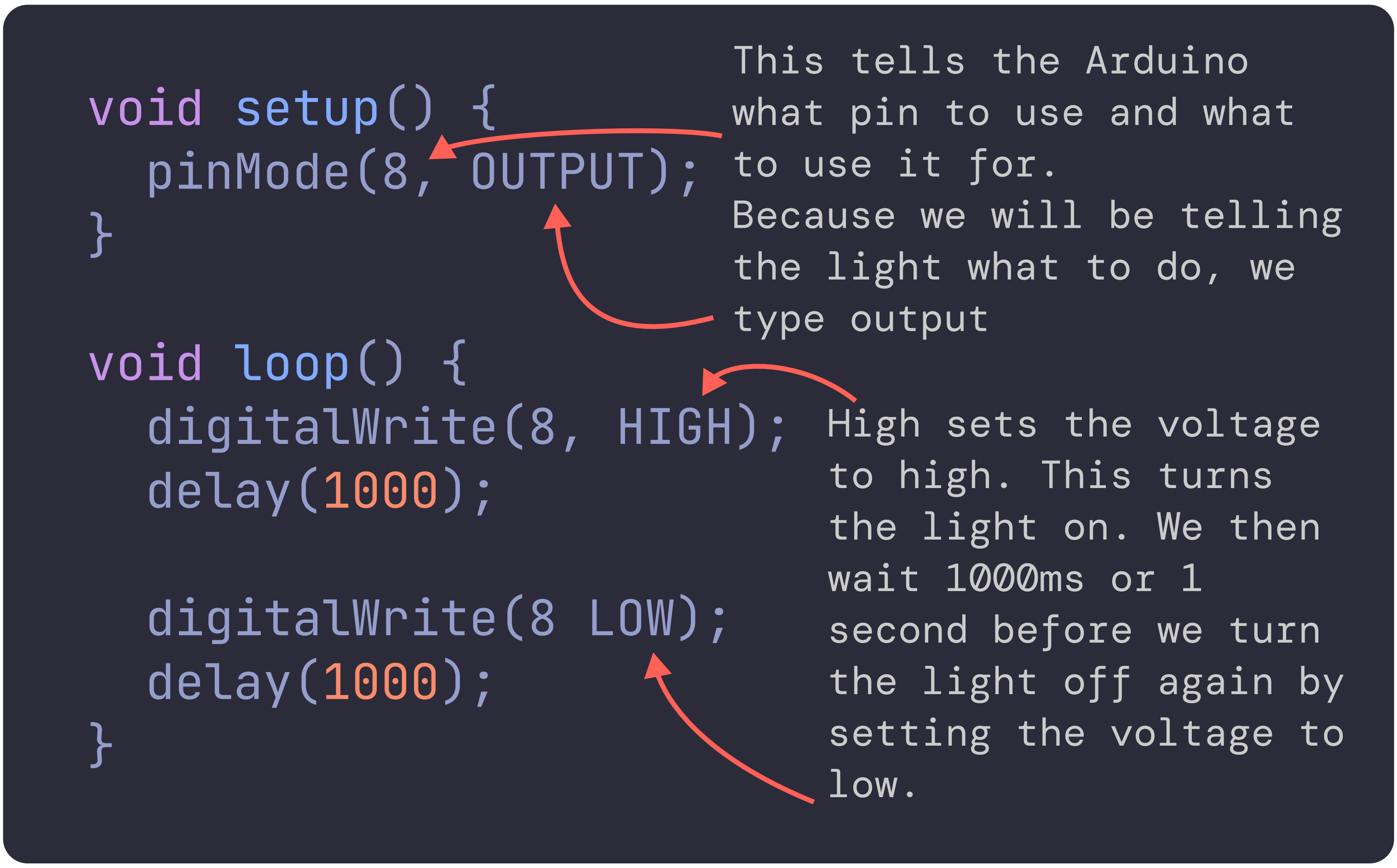
```
void setup() {

}

void loop() {

}
```

You will notice 2 functions

Setup is code that runs only once when the arduino is powered on

Loop keeps on running as long as the arduino has power

Step 3:
Coding/Programming

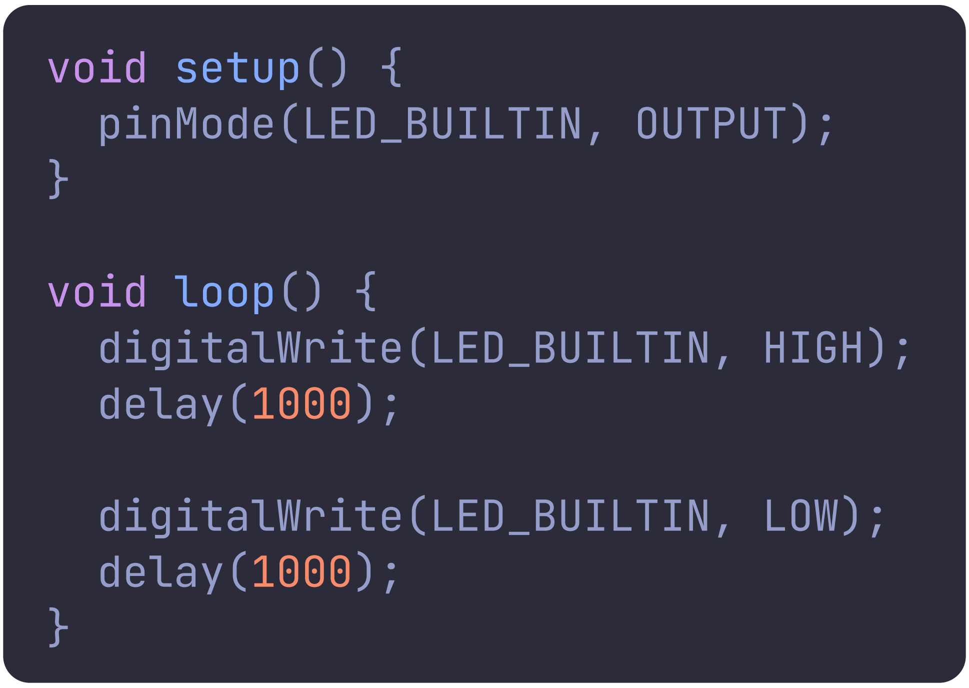


```
void setup() {
  pinMode(8, OUTPUT);
}

void loop() {
  digitalWrite(8, HIGH);
  delay(1000);

  digitalWrite(8 LOW);
  delay(1000);
}
```

This tells the Arduino what pin to use and what to use it for.

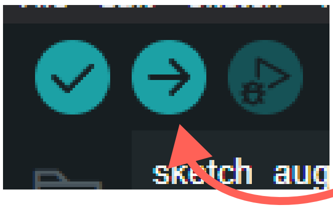
Because we will be telling the light what to do, we type output

High sets the voltage to high. This turns the light on. We then wait 1000ms or 1 second before we turn the light off again by setting the voltage to low.



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

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);

  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

This is the finished code



Pressing this will check the code for errors then send it to the Arduino

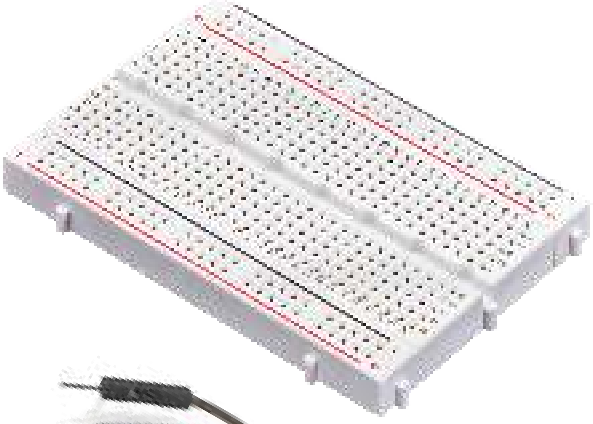
If there are no errors then the code will start running on the Arduino!

How to make control a servo

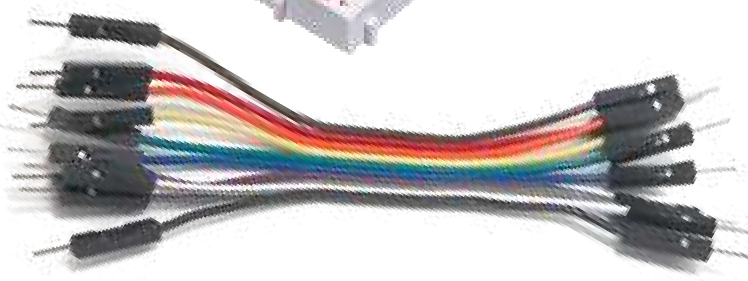
Step 1:

Gather your tools,

- Breadboard
  - makes connecting wires easier



- Jumper Wires
  - wires with metal pins at the end to make connecting easier



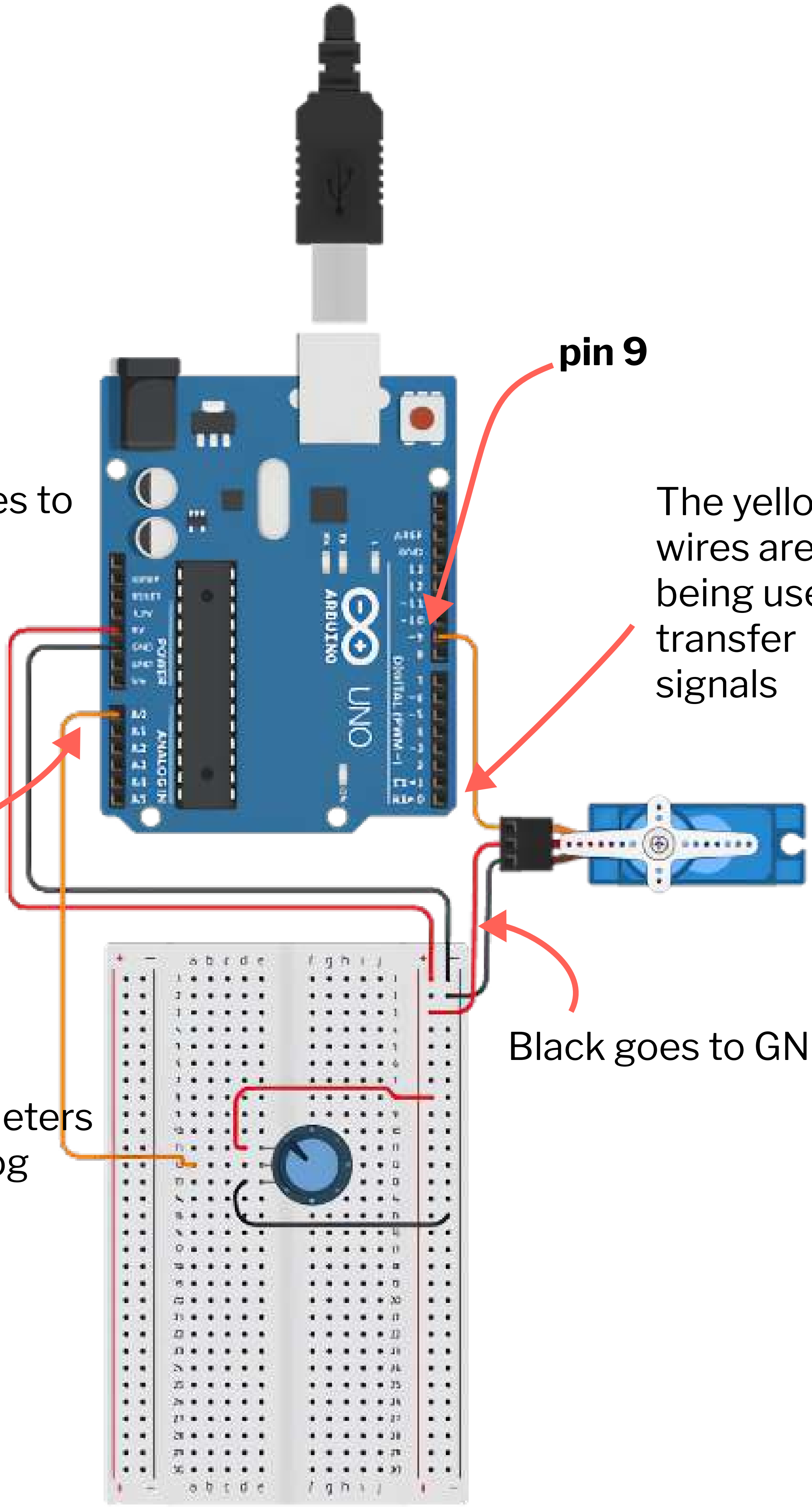
- Arduino (Microcontroller)
  - to give the circuit instructions

- USB type B cable
  - to program the Arduino and provide power

- Servo
  - a special motor that moves only how much you tell it to

- Potentiometer
  - a special dial that tells the microcontroller how much it has been turned

Servo Wiring Diagram



Red goes to 5v

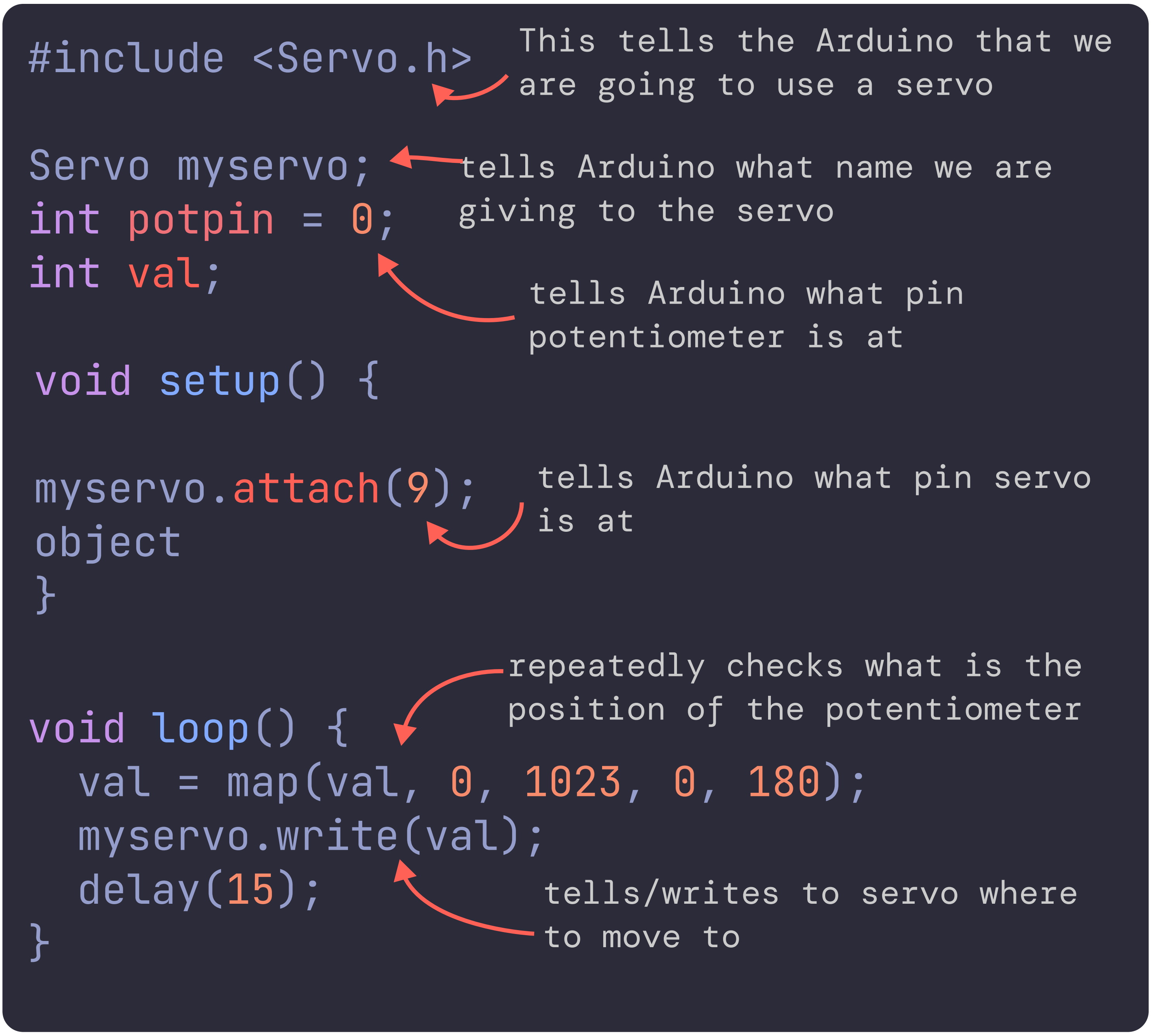
pin 9

The yellow wires are being used to transfer signals

pin A0
Because potentiometers have analog output

Black goes to GND

Coding



```
#include <Servo.h>

Servo myservo;
int potpin = 0;
int val;

void setup() {

myservo.attach(9);
object
}


void loop() {
  val = map(val, 0, 1023, 0, 180);
  myservo.write(val);
  delay(15);
}
```

This tells the Arduino that we are going to use a servo

tells Arduino what name we are giving to the servo

tells Arduino what pin potentiometer is at

tells Arduino what pin servo is at

repeatedly checks what is the position of the potentiometer

tells/writes to servo where to move to

You can now

- send the code to the Arduino
- make sure everything is connected properly

and **Enjoy!!**

THE END

About our Authors

Shashwat Jaguri

Hi, I’m the founder of the organization Coduriosity. My journey started with my own fascination with tech. Inspired by what I was able to create and my own drive too keep on learning, I’ve been working to bring opportunities to pursue stem to rural locations by running workshops and trying to bring the resources, getting funding for the electronics needed to support such programs and providing students and teachers with the knowledge needed to continue exploring on their own too!

Sebastian Inestroza

Hi! I am a passionate and dedicated member of Coduriosity, committed to making a positive impact in the engineering field. With a deep-rooted belief in the power of collective action, I contribute tirelessly to the organization's mission, striving for a better future for all. Driven by an unwavering passion for sophisticated and impactful education, I actively engage as the resource director for the organization to help enrich the minds of those we reach.

Salil Joshi

My commitment to Coduriosity cause is fueled by my belief that everyone has a role to play in creating positive change! I believe that through collaboration, education and awareness, we can address the issues facing our world and foster a more compassionate society. As the head of social media, I craft engaging content, dedicated to spreading awareness and driving positive change. I hope to spread awareness and amplify our mission for a more compassionate and just society

UNLOCKING MICROCONTROLLERS

Unlock the Marvels of Technology with "Microcontrollers Unlocked" : A Kid's Guide to Arduino Adventures! This captivating book is your child's key to a world of creativity, innovation, and hands-on learning.

In "Microcontrollers Unlocked," young explorers will journey into the exciting realm of microcontrollers. **Discover the secrets** behind these miniature wonders that power our gadgets and gizmos. Through clear and kid-friendly instructions, your child will grasp the art of wiring and connecting microcontrollers, **unleashing their potential for endless fun.**

from Coduriosity

www.coduriosity.com

designed by Salil Joshi