

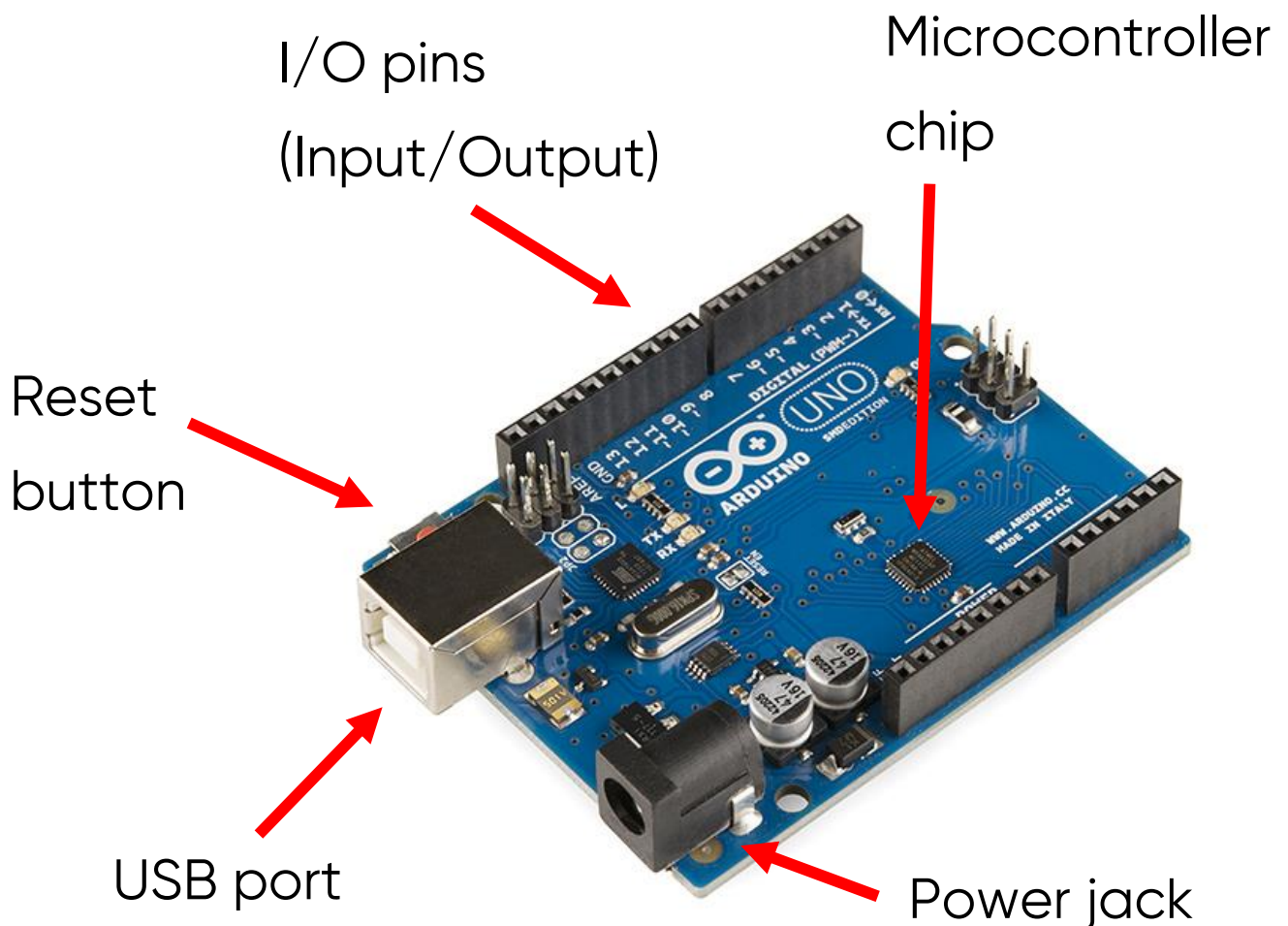
ARDUINO INTRO

Arduino is a microcontroller board – a small computer on an electronic chip. Arduino is an open-source project, which means people are free to copy the Arduino design, use it to make new products and share them. It was started to give people a cheap, simple way to build projects using sensors and actuators.

A sensor is something which reads information from the environment, and outputs an electronic signal that the Arduino can read.

An actuator is something that moves or does something, like a motor.

This means Arduino can be used to build things like robots (e.g. robocar) or devices that interact with the environment, like thermostats, self-watering plants, motion detectors, or lots of other things!



PROGRAMMING AN ARDUINO AND BUILDING CIRCUITS

We write some code in C++ and upload it to the Arduino using the "Arduino IDE" program (IDE = integrated development environment). Using this code we can READ the pins on the Arduino, which means take a measurement of what signal or VOLTAGE is being applied to that pin, and we can WRITE the pins, which means supply the pins with a certain voltage or signal.

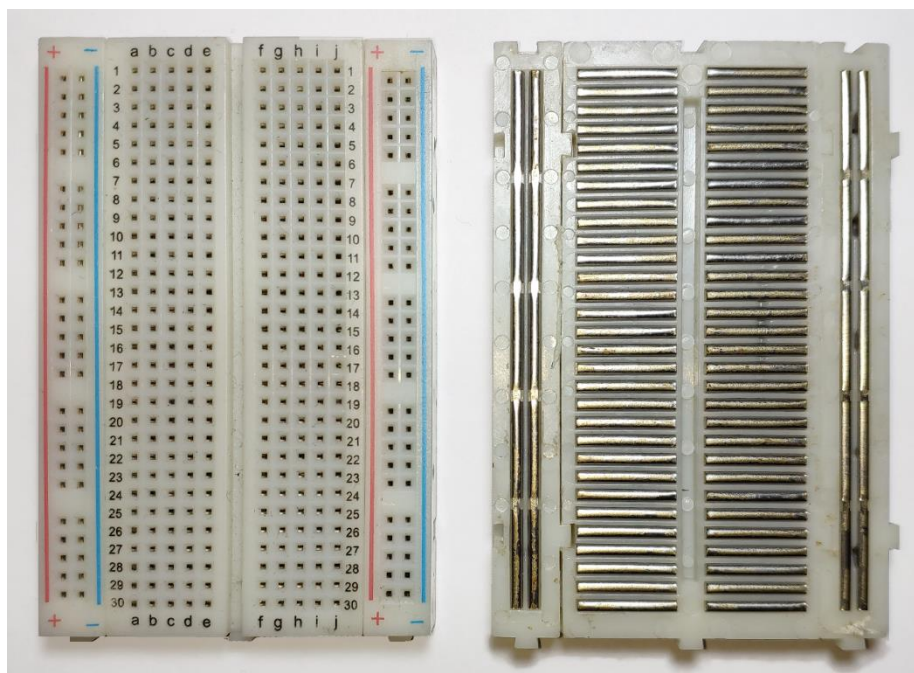
All Arduino code has the following structure:

Everything here will run
once, when the power
is turned on

then the code here will
repeat indefinitely

```
1  void setup() {  
2      // put your setup code here, to run once:  
3  
4  }  
5  
6  void loop() {  
7      // put your main code here, to run repeatedly:  
8  
9  }
```

We'll build circuits on the breadboard, see how the breadboard pins are connected below.



INTRO TASKS

1.) BLINK AN LED

<https://docs.arduino.cc/built-in-examples/basics/Blink>

2.) READ A POTENTIOMETER

<https://docs.arduino.cc/built-in-examples/basics/AnalogReadSerial>

3.) DIM AN LED

<https://docs.arduino.cc/built-in-examples/analog/AnalogInOutSerial>

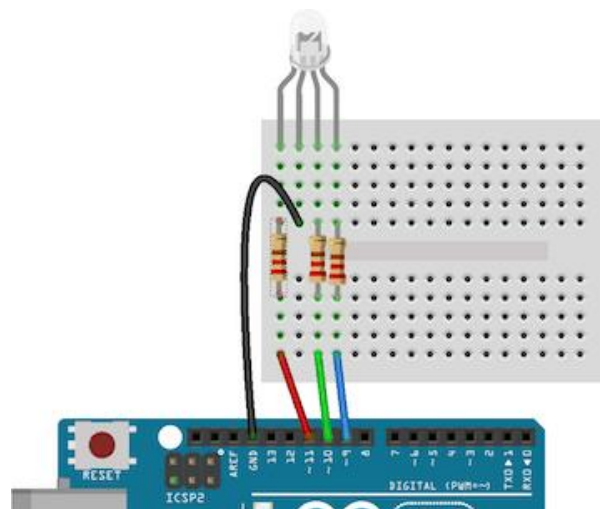
4.) USE A BUTTON

<https://docs.arduino.cc/built-in-examples/digital/Button>

5.) CONTROL AN RGB LED

Try making different colours

```
int red_light_pin= 11;
int green_light_pin = 10;
int blue_light_pin = 9;
void setup() {
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}
void loop() {
  RGB_color(255, 0, 0); // Red
  delay(1000);
  RGB_color(0, 255, 0); // Green
  delay(1000);
  RGB_color(0, 0, 255); // Blue
  delay(1000);
}
void RGB_color(int red_light_value, int green_light_value, int
blue_light_value) {
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
  analogWrite(blue_light_pin, blue_light_value);
}
```



CHALLENGE TASKS

- 1.) Change the RGB colour with a potentiometer or push button
- 2.) Follow the example
<https://docs.arduino.cc/built-in-examples/digital/toneMelody>
 - a. Write your own tune
 - b. Control which note is played with a potentiometer or other input
- 3.) Choose a sensor from the kit, search online how to use it and take a reading
- 4.) Use your sensor to control the brightness of an LED, the colour of the RGB, or the note played by the speaker