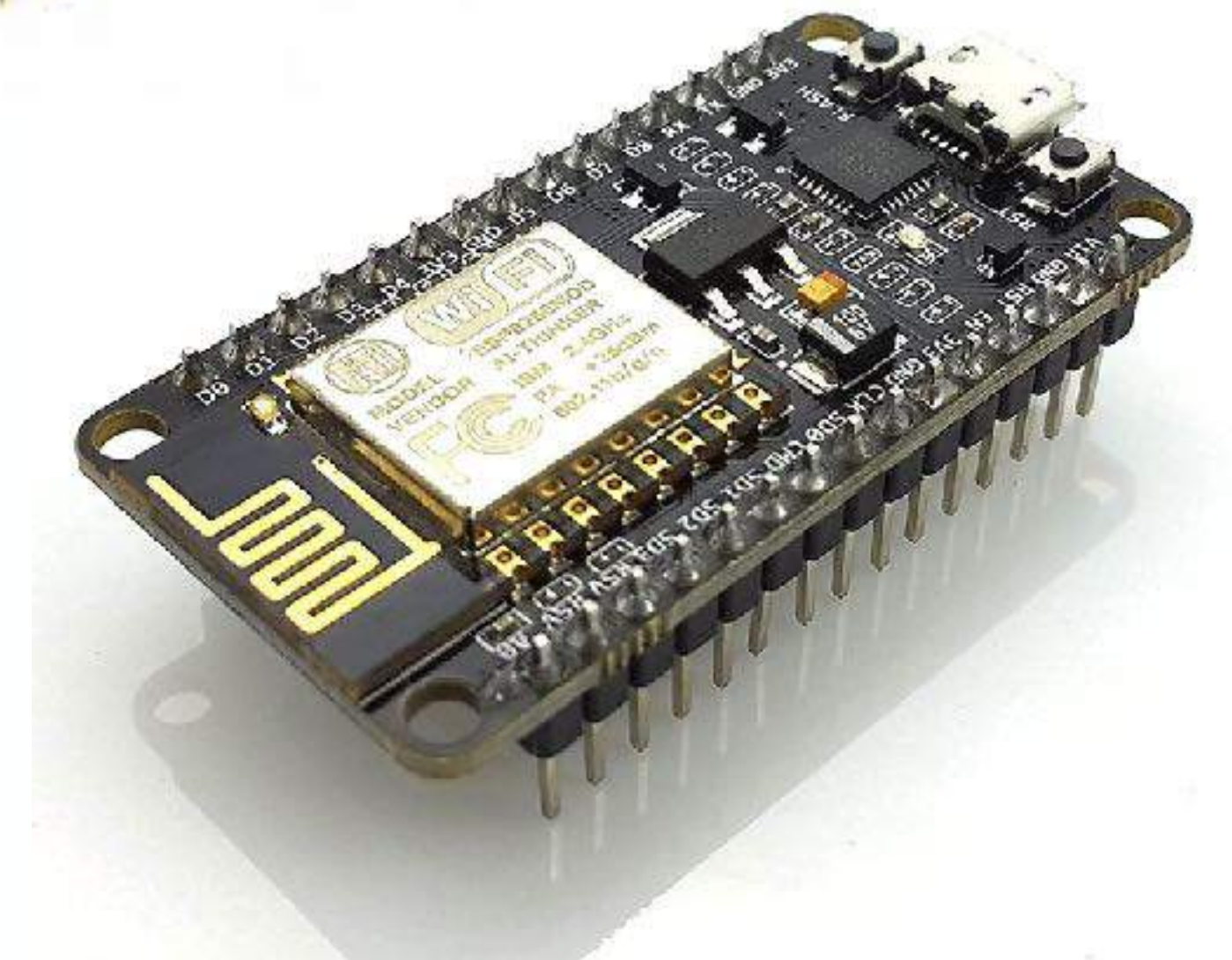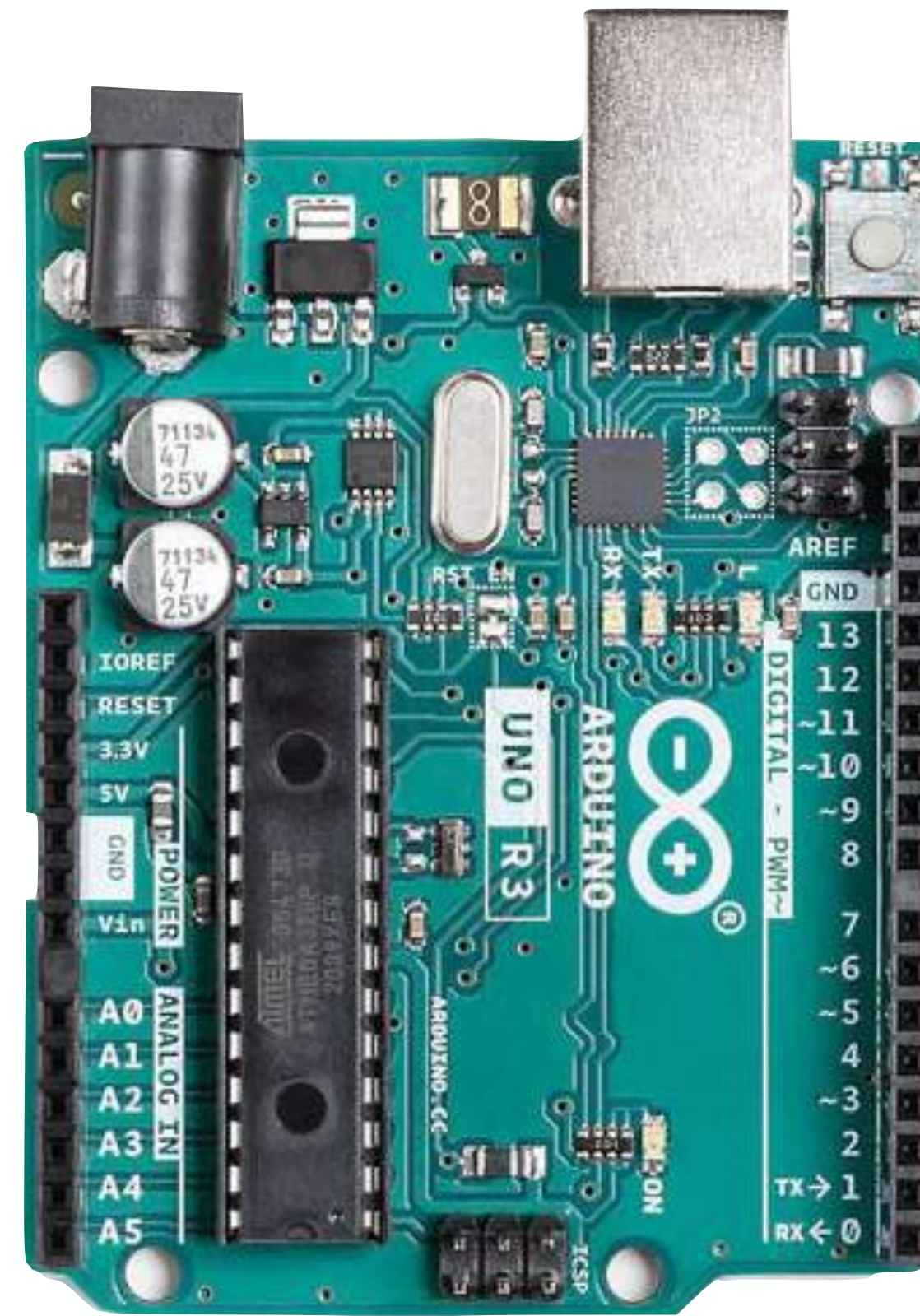# Electronics Engineering Eca

Block C Lesson 2
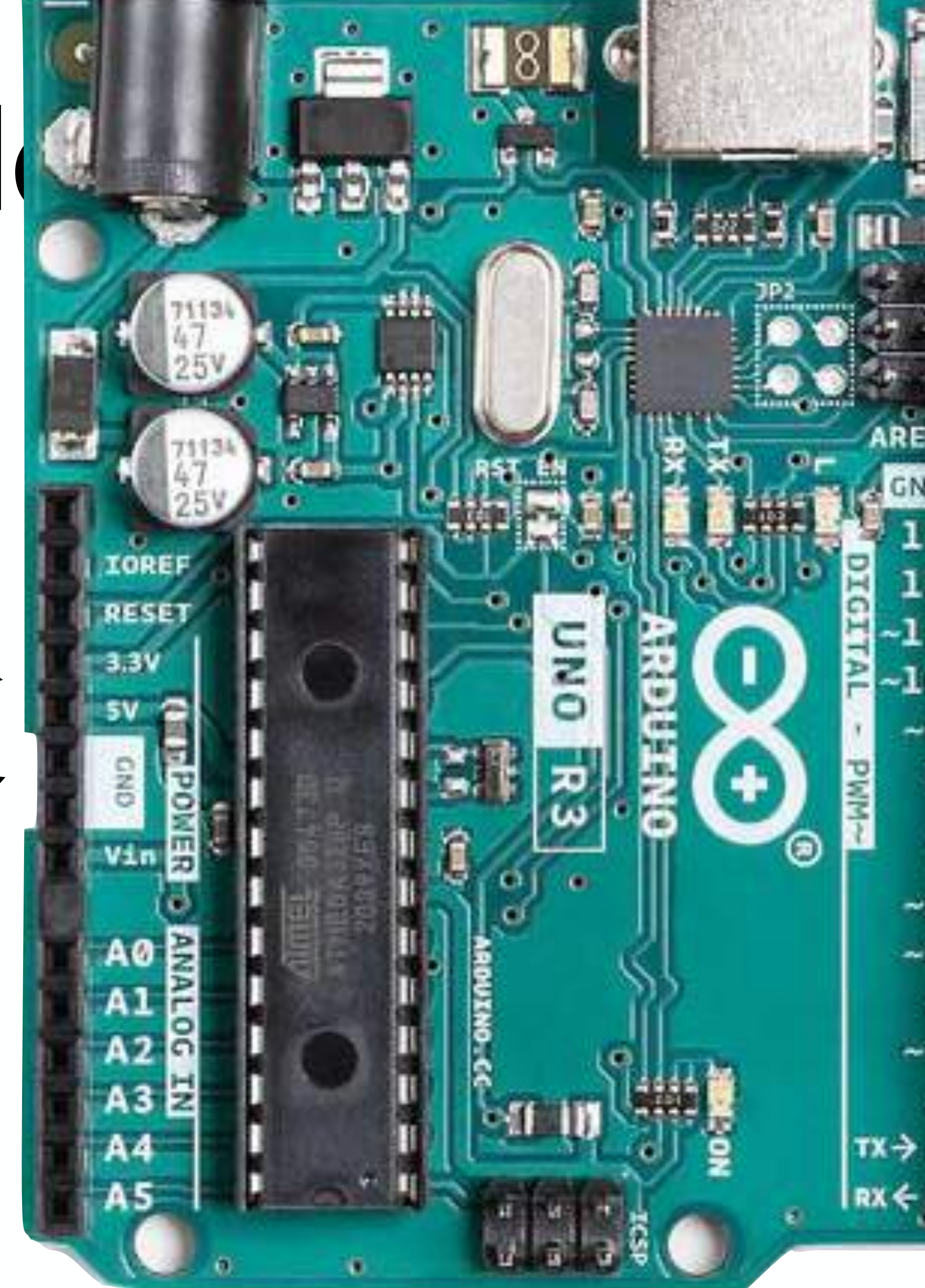
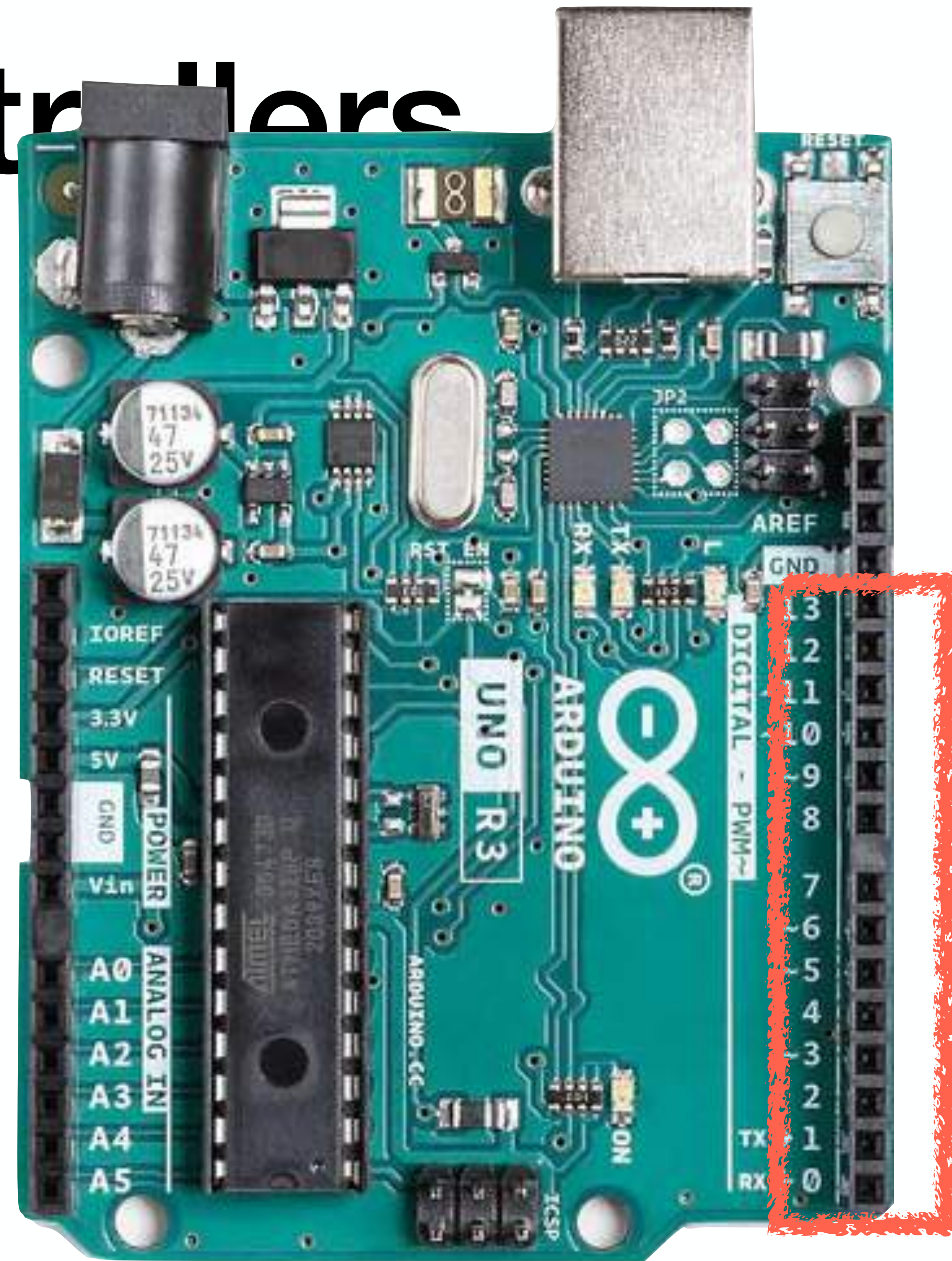# Microcontrollers

- The brain of an electronics project

Microcontroll

- Power output through 3.3V and 5V pin

  - Power connect to 3.3v or 5V pin on sensors to power them

- GND - Ground pin, connect ground to ground to complete a circuit

- Stab a jumper wire into the pin and connect to bread board

Microcontrollers

- Digital Pins: 2 States

  - HIGH (1) – Typically 3.3V or 5V

  - LOW (0) – Typically 0V (GND)

- The Microcontroller can either read the voltage level of a pin or write a voltage to the digital pin

- Can be controlled through code

Microcontrollers

- Analog Pins: Read voltage values of a pin

- Used to read analog signals eg. potentiometers, where values change gradually, rather than switching between just two states

- Will discuss in a future class
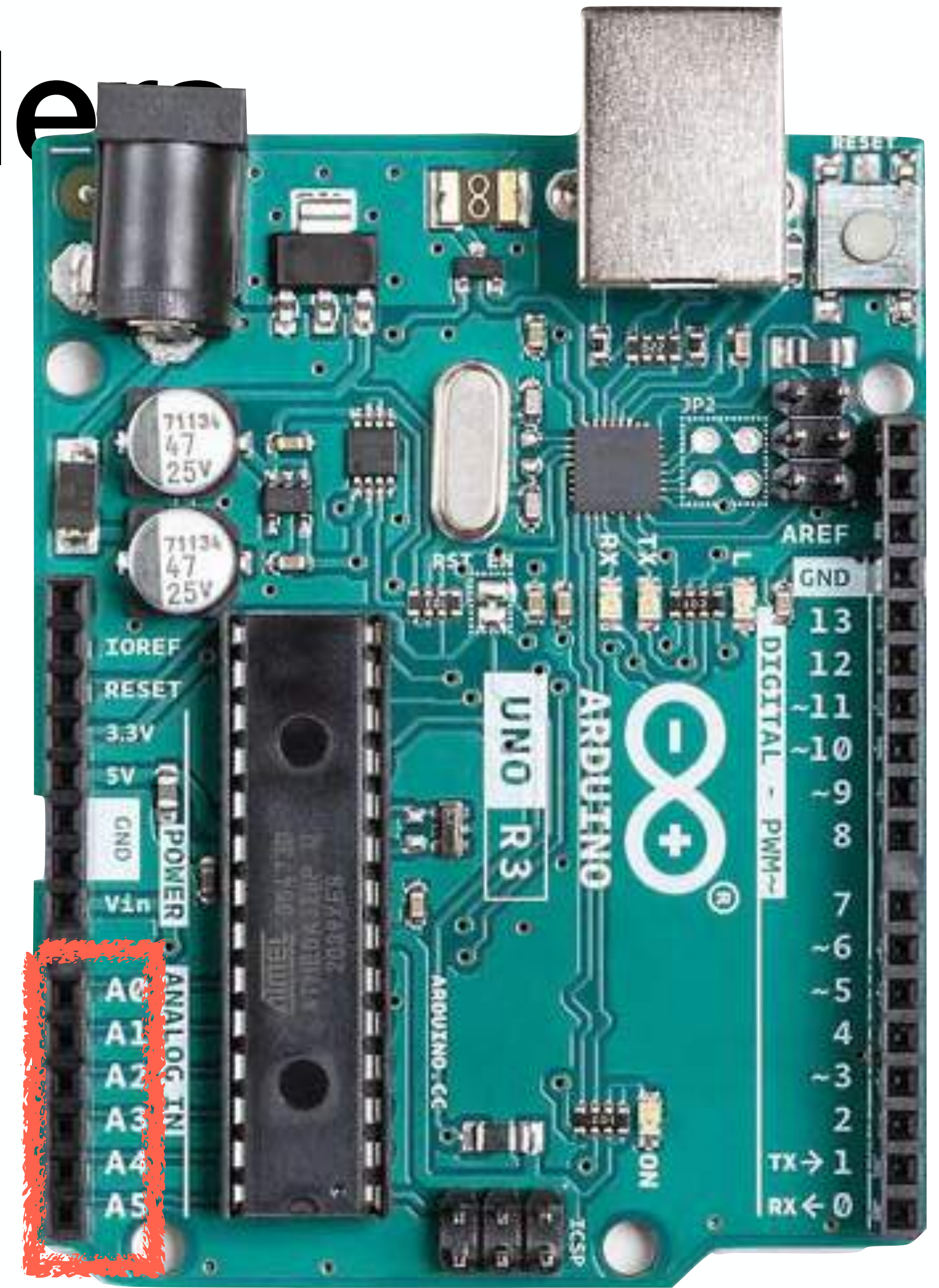
Safety

- Always Disconnect Power Before Working on a Circuit

- Never connect power directly to ground, this results in a short circuit and something will burn

- Double-check wiring to avoid direct connections between power and ground.

- Double check wiring to the microcontroller, ensure that you understand the function of the pin before connecting to it
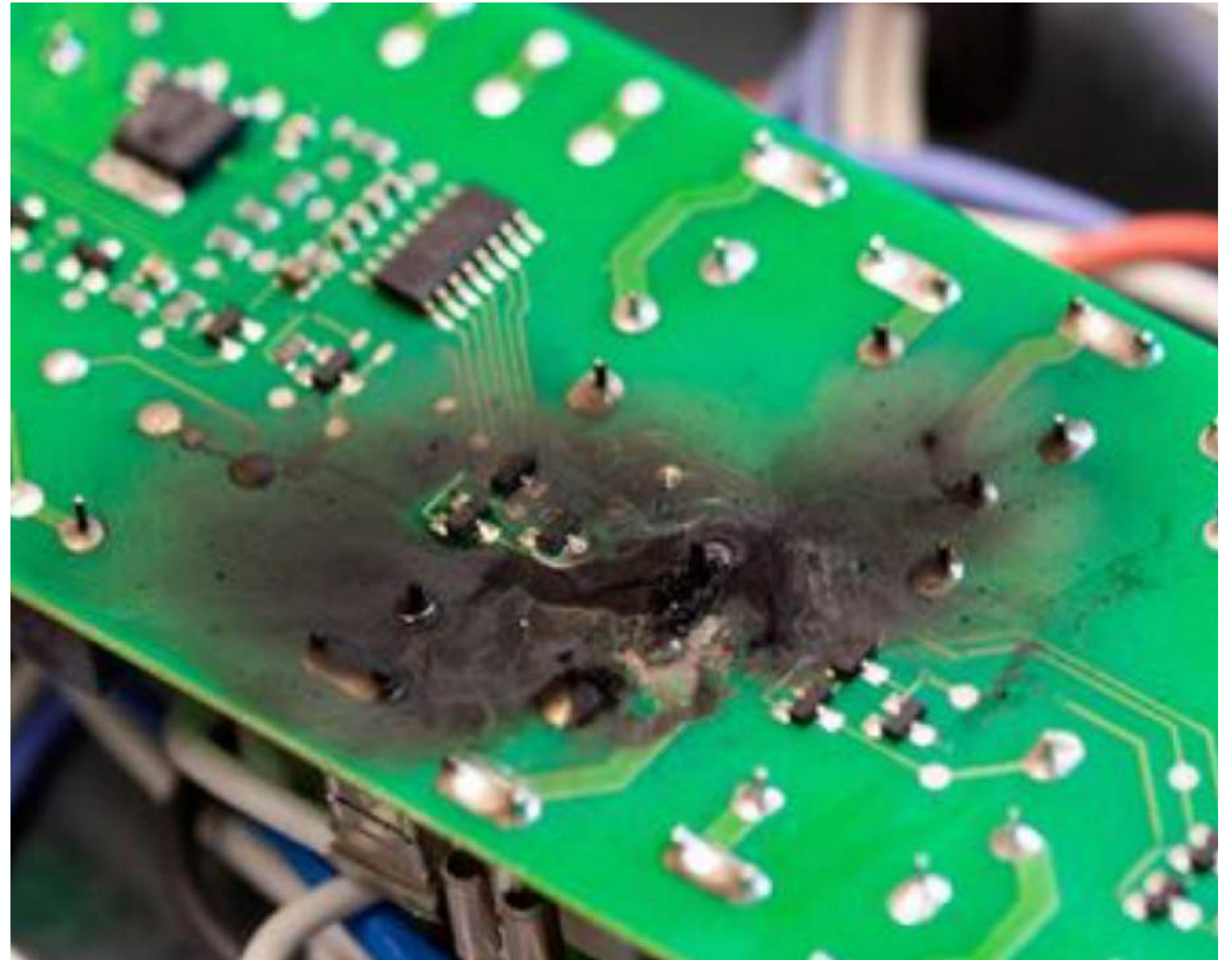
Coding a microcontroller

- We will be using Arduino IDE to program the microcontrollers

  https://www.arduino.cc/en/software

Reading push button

- Connect Circuit as shown:

- Try out the following code:

```
#define BUTTON_PIN 4

void setup()
{
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop()
{
  Serial.println(digitalRead(BUTTON_PIN));
  delay(100);
}
```
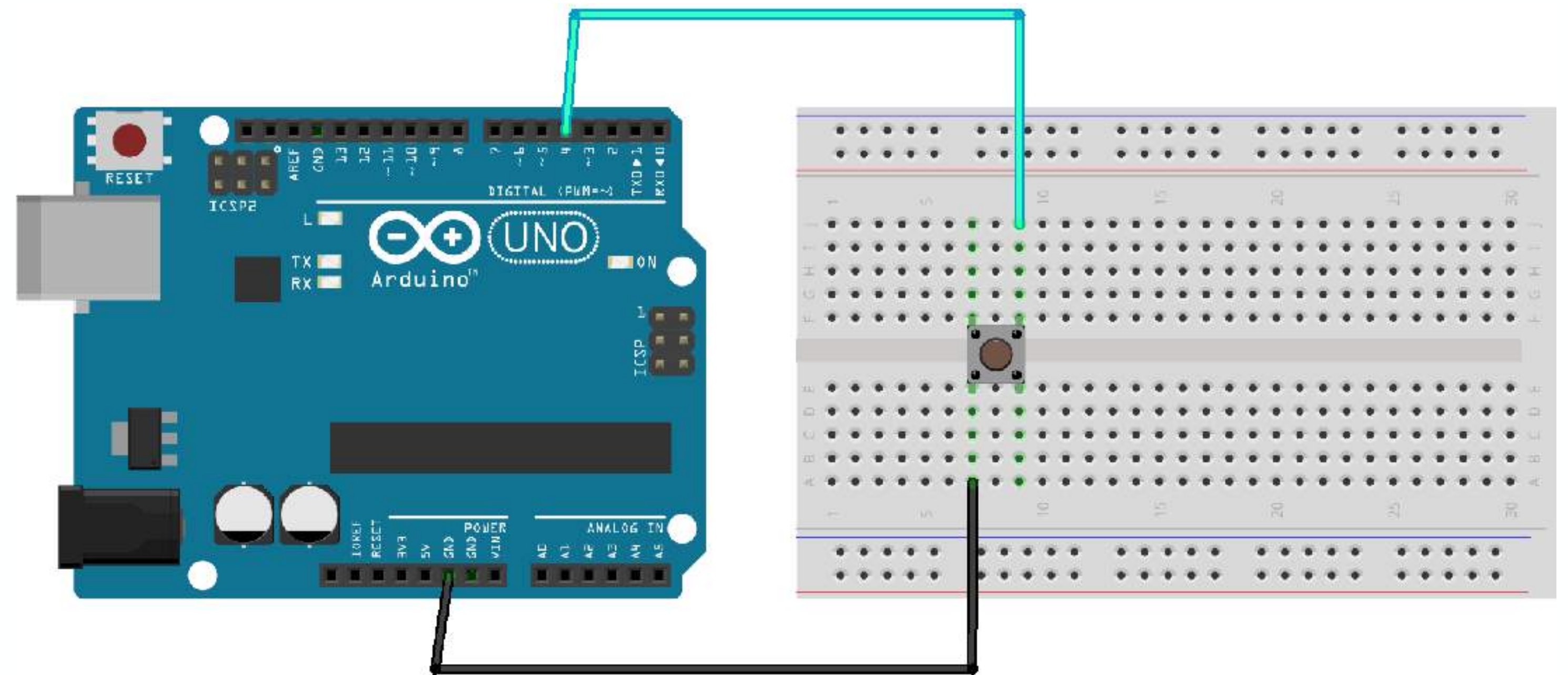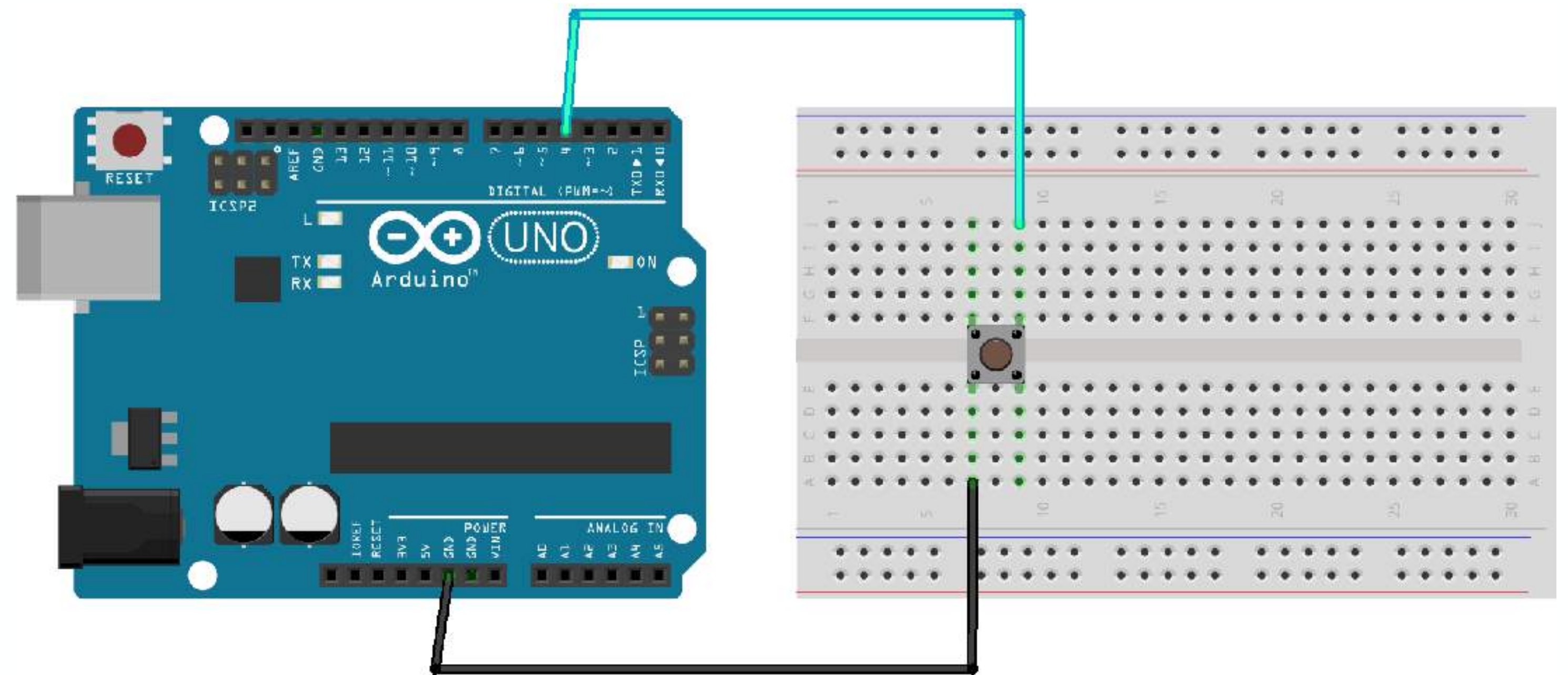

fritzing

Reading push button

- Connect Circuit as shown:

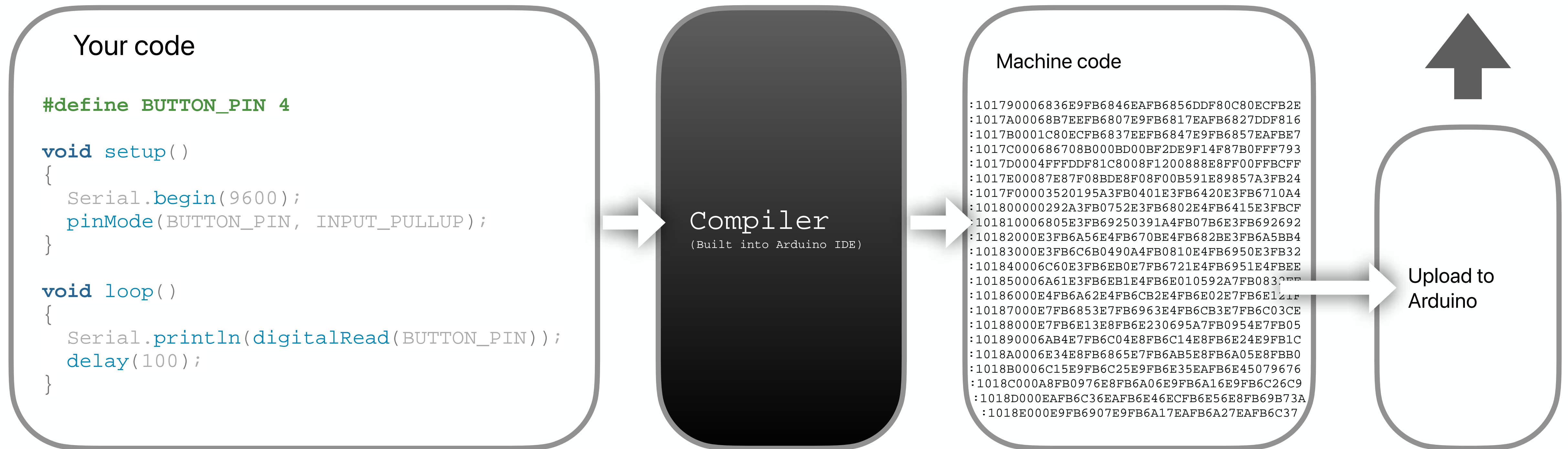- Try out the following code:

```
#define BUTTON_PIN 4

void setup()
{
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop()
{
  Serial.println(digitalRead(BUTTON_PIN));
  delay(100);
}
```


fritzing

What is a compiler??

Your code

```
#define BUTTON_PIN 4

void setup()
{
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop()
{
  Serial.println(digitalRead(BUTTON_PIN));
  delay(100);
}
```

Compiler
(Built into Arduino IDE)

Machine code

```
:101790006836E9FB6846EAFB6856DDF80C80ECFB2E
:1017A00068B7EEFB6807E9FB6817EAFB6827DDF816
:1017B0001C80ECFB6837EEFB6847E9FB6857EAFBE7
:1017C000686708B000BD00BF2DE9F14F87B0FFF793
:1017D0004FFFDDF81C8008F1200888E8FF00FFBCFF
:1017E00087E87F08BDE8F08F00B591E89857A3FB24
:1017F00003520195A3FB0401E3FB6420E3FB6710A4
:101800000292A3FB0752E3FB6802E4FB6415E3FBCF
:101810006805E3FB69250391A4FB07B6E3FB692692
:10182000E3FB6A56E4FB670BE4FB682BE3FB6A5BB4
:10183000E3FB6C6B0490A4FB0810E4FB6950E3FB32
:101840006C60E3FB6EB0E7FB6721E4FB6951E4FBEE
:101850006A61E3FB6EB1E4FB6E010592A7FB0830FF
:10186000E4FB6A62E4FB6CB2E4FB6E02E7FB6E121F
:10187000E7FB6853E7FB6963E4FB6CB3E7FB6C03CE
:10188000E7FB6E13E8FB6E230695A7FB0954E7FB05
:101890006AB4E7FB6C04E8FB6C14E8FB6E24E9FB1C
:1018A0006E34E8FB6865E7FB6AB5E8FB6A05E8FBB0
:1018B0006C15E9FB6C25E9FB6E35EAFB6E45079676
:1018C000A8FB0976E8FB6A06E9FB6A16E9FB6C26C9
:1018D000EAFB6C36EAFB6E46ECFB6E56E8FB69B73A
:1018E000E9FB6907E9FB6A17EAFB6A27EAFB6C37
```

Upload to Arduino

Format of code

```
void setup()
{
  //Everything in here is ran once
on startup
}

void loop()
{
  //Code here loops indefinitely
}
```

- Text that starts with // are comments, Ignored by compiler

Reading push button

```
#define BUTTON_PIN 4

pinMode(BUTTON_PIN, INPUT_PULLUP);
```

- #Define is a Macro, Compiler replaces the define at compile time

- pinMode defines how the code should handle the pin on the microcontroller

INPUT_PULLUP

INTERNAL PULL-UP RESISTOR CONFIGURATION

Push Button (4-Leg)

1   3

2   4

GND

Pin 2

Outside Microcontroller

Inside Microcontroller

20KΩ

5V

Software controlled
switch enabled by
INPUT_PULLUP

Pin 2