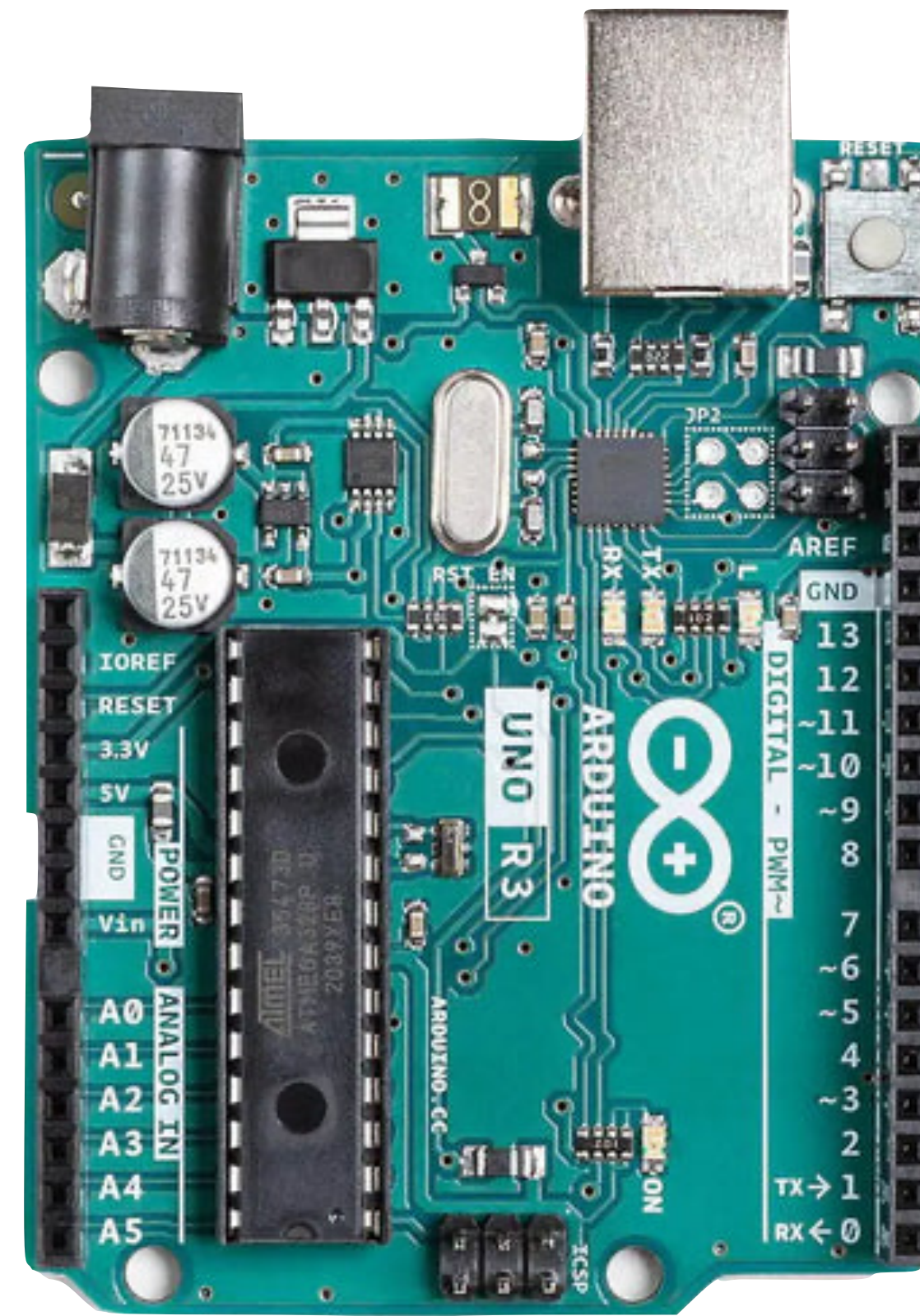


# Electronics Engineering Eca

Block C Lesson 2

# Microcontrollers

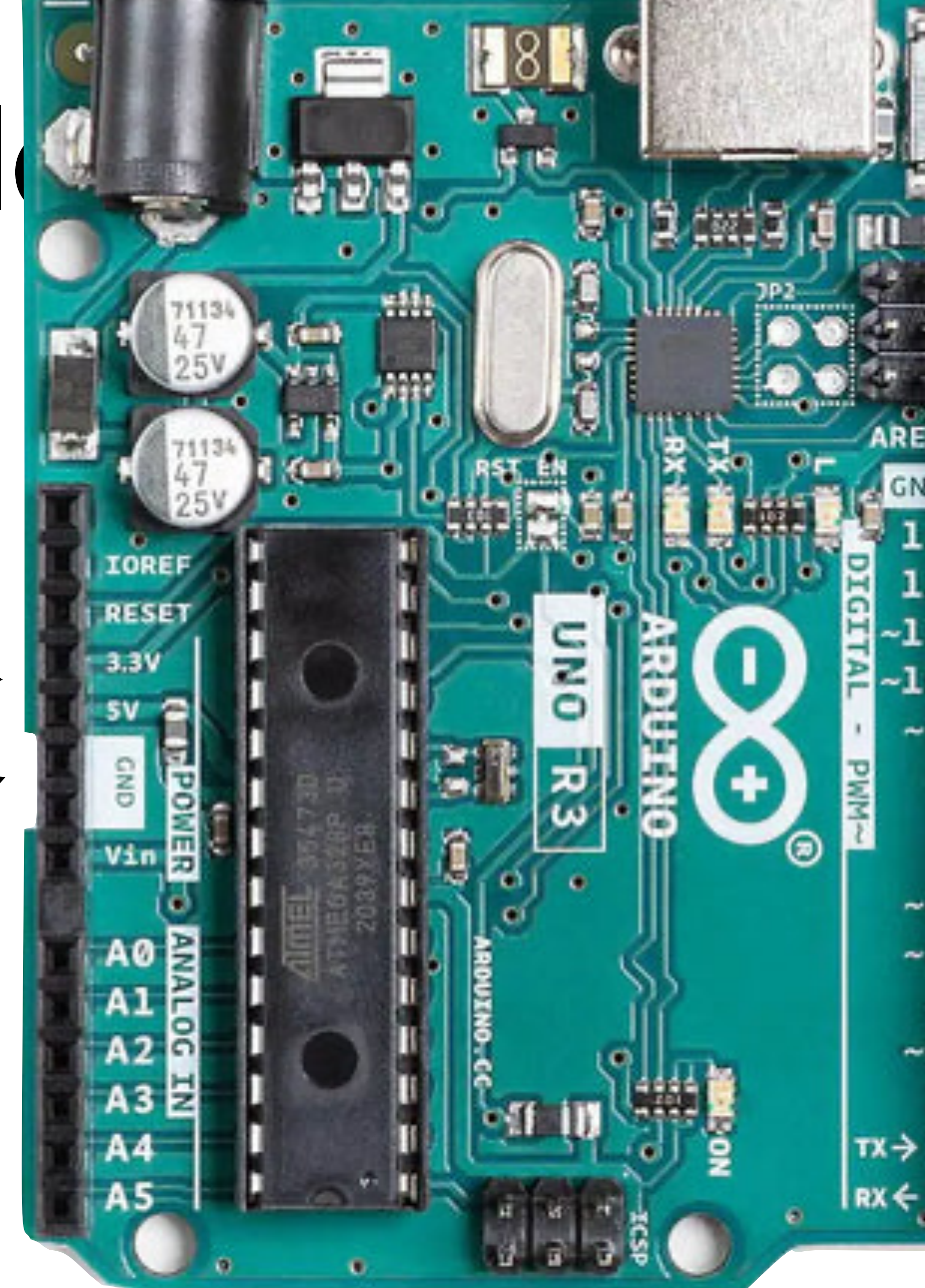
- The brain of an electronics project





# Microcontroller

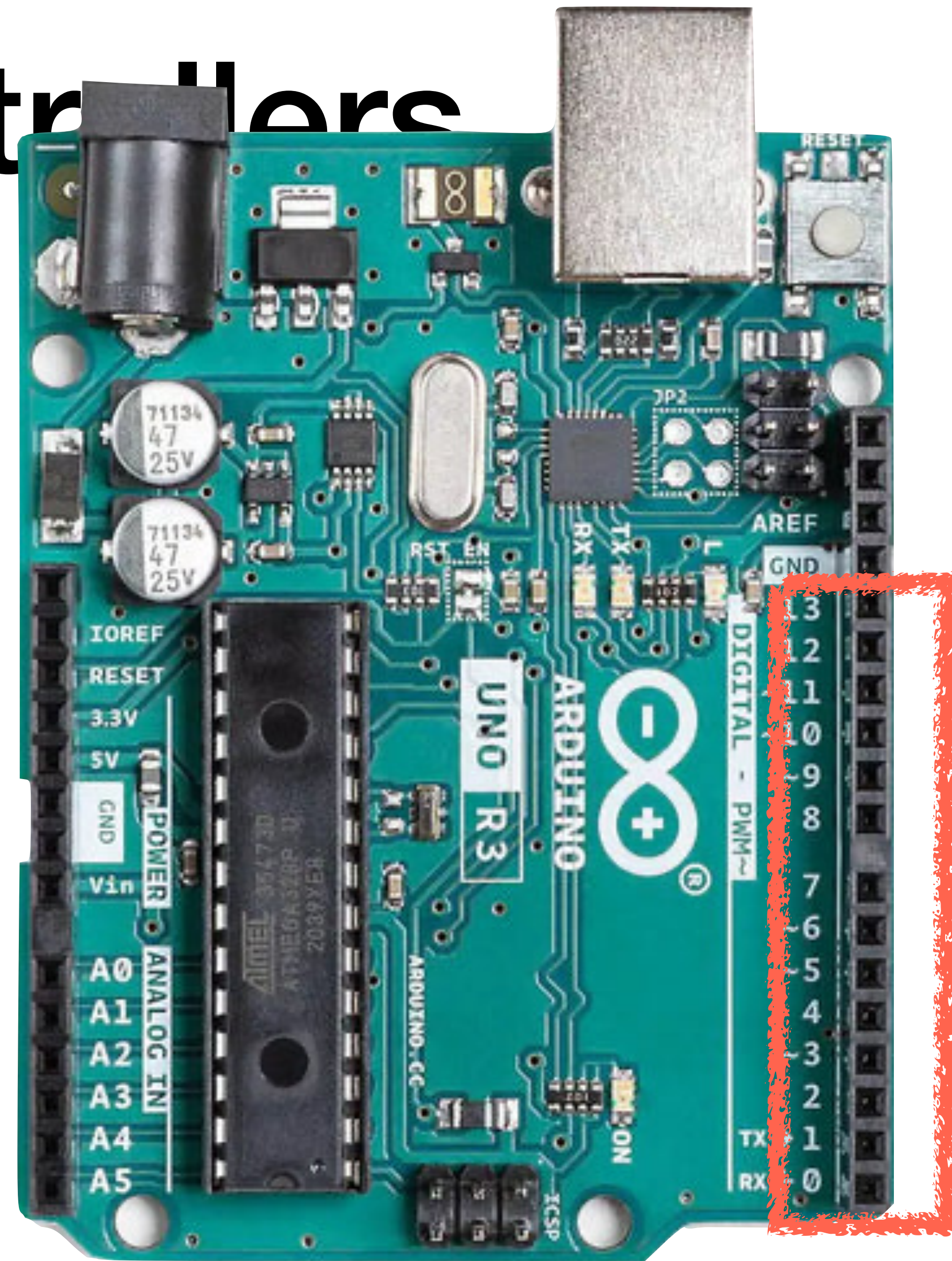
- Power output through 3.3V and 5V pin
  - Power connect to 3.3v or 5V pin on sensors to power them
- GND - Ground pin, connect ground to ground to complete a circuit
- Stab a jumper wire into the pin and connect to bread board





# Microcontrollers

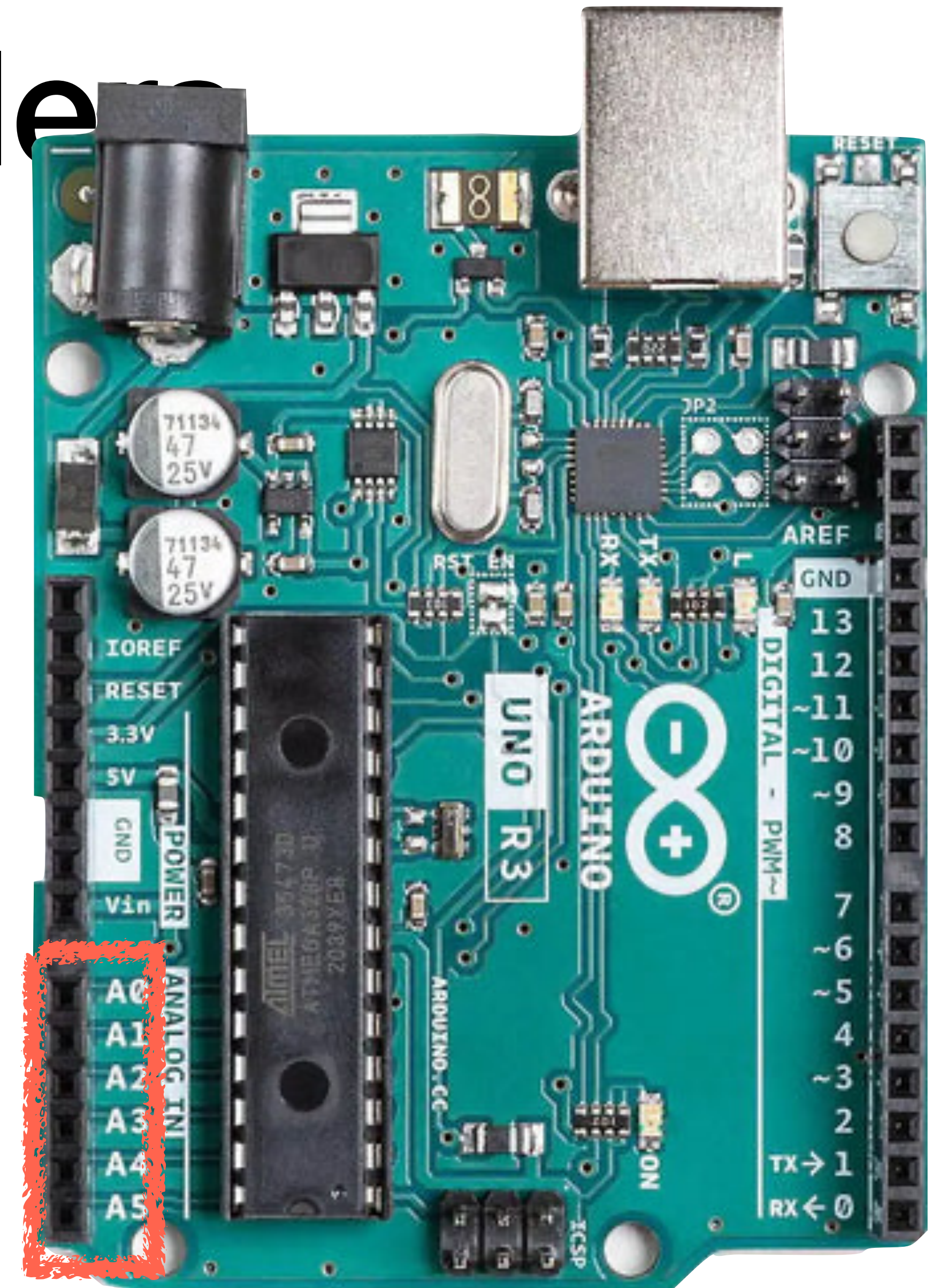
- Digital Pins: 2 States
  - HIGH (1) – Typically 3.3V or 5V
  - LOW (0) – Typically 0V (GND)
- The Microcontroller can either read the voltage level of a pin or write a voltage to the digital pin
- Can be controlled through code





# Microcontroller

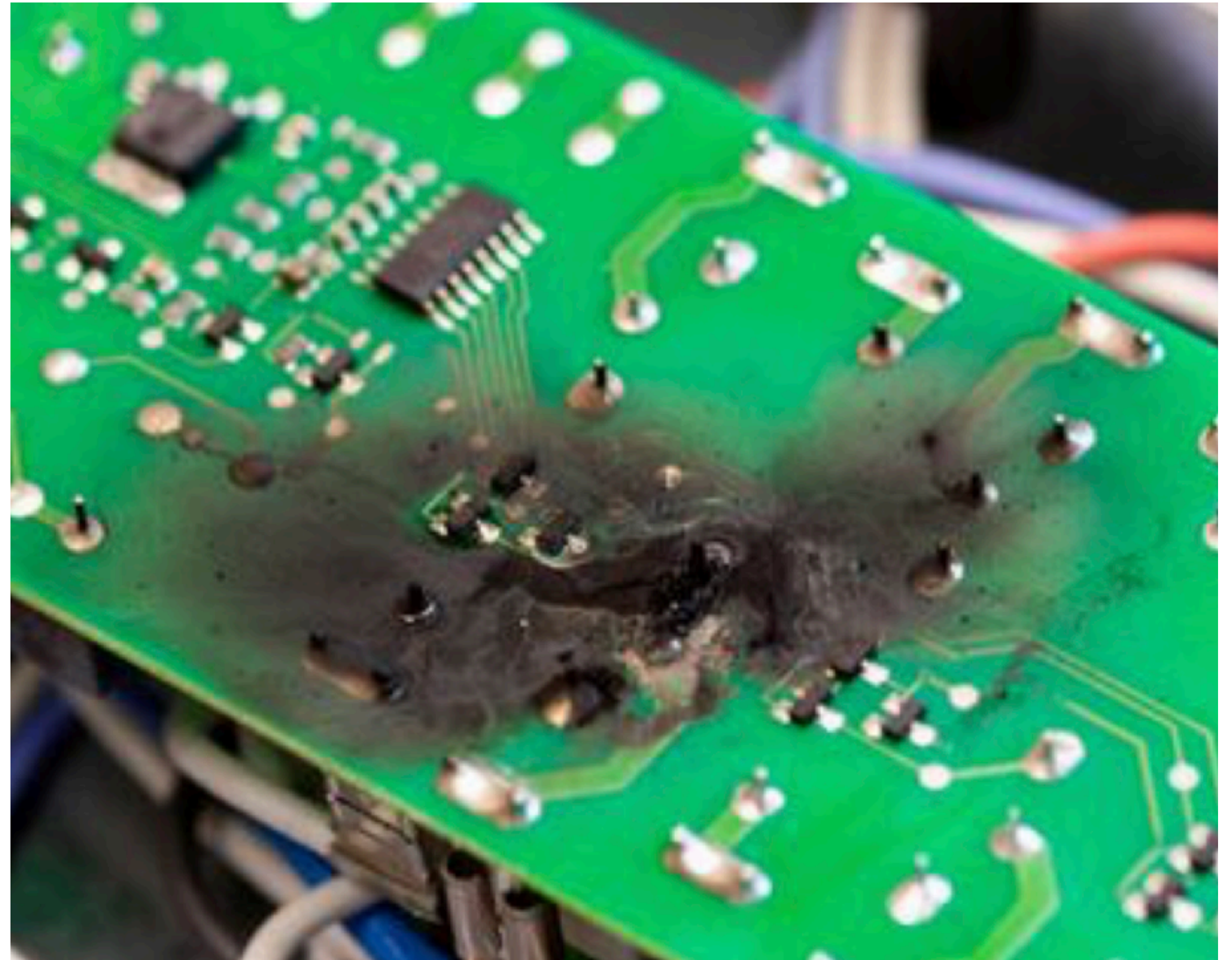
- Analog Pins: Read voltage values of a pin
- Used to read analog signals eg. potentiometers, where values change gradually, rather than switching between just two states
- Will discuss in a future class





# 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>





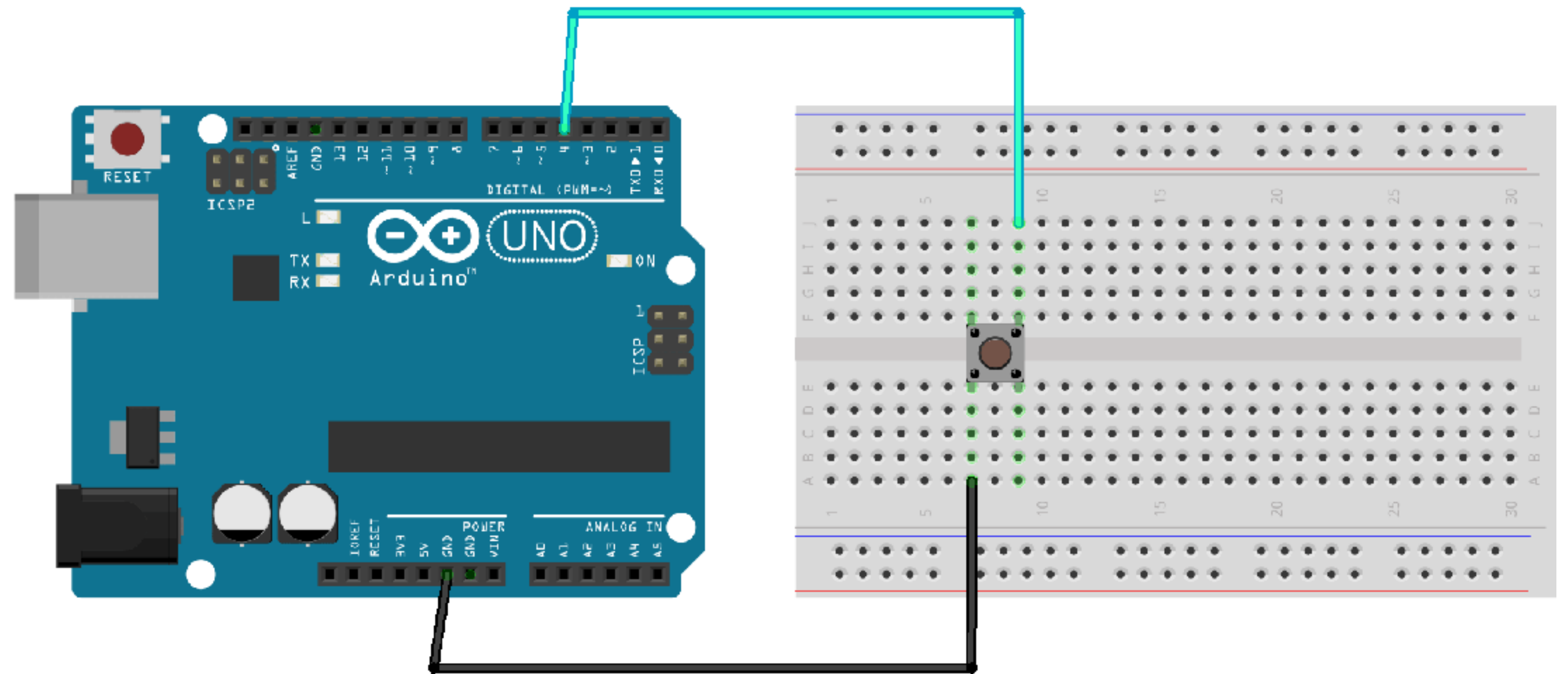
# Reading push button

- 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



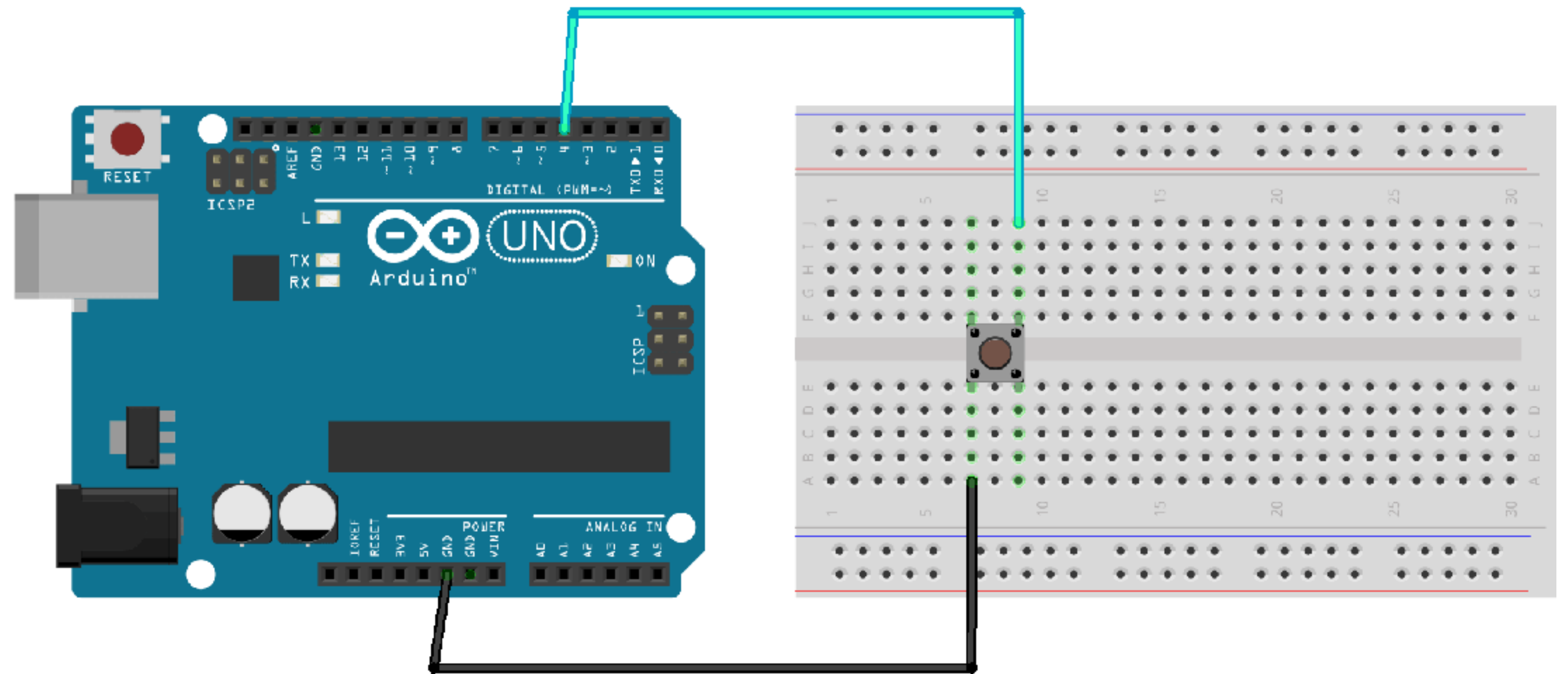
# Reading push button

- Connect Circuit as shown:
- Try out the following code:

```
#define BUTTON_PIN 4

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

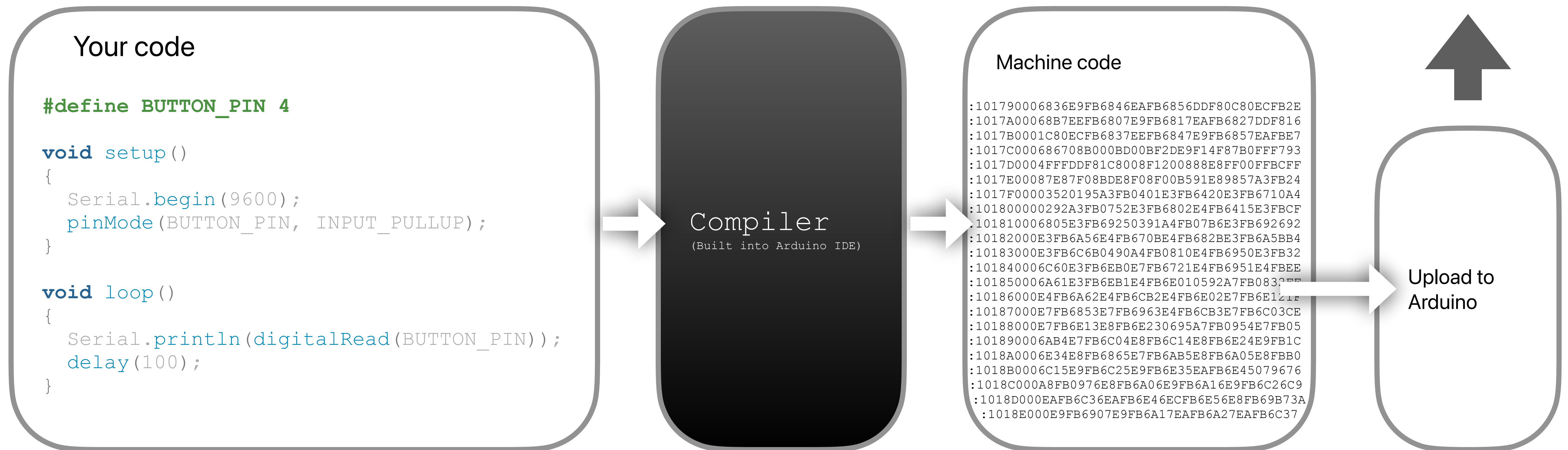
void loop()
{
  Serial.println(digitalRead(BUTTON_PIN));
  delay(100);
}
```



fritzing



# What is a compiler??





# 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

