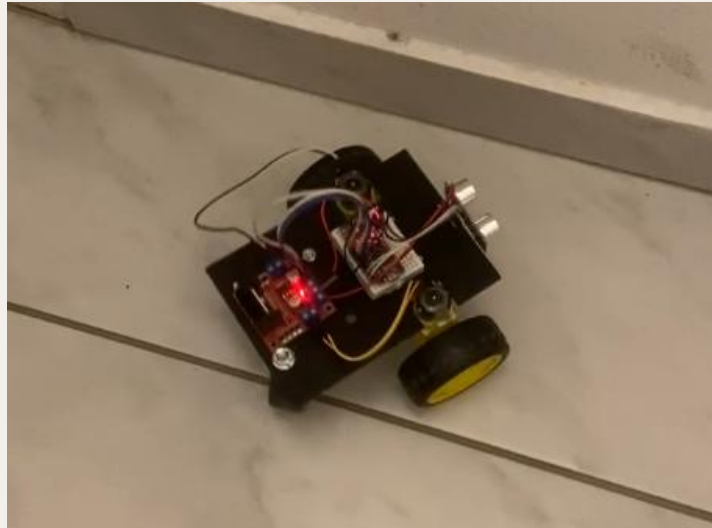


# Robot car workshop



by Domokos Kertész



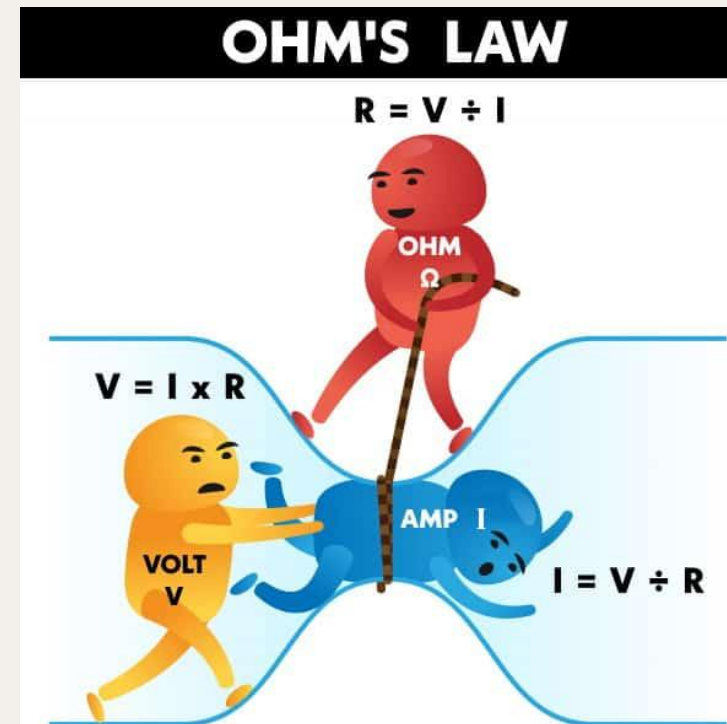
# Goal

- Understand the basics
- Move
- Avoid crash
- Create custom stuff with the car
- Tools for more building



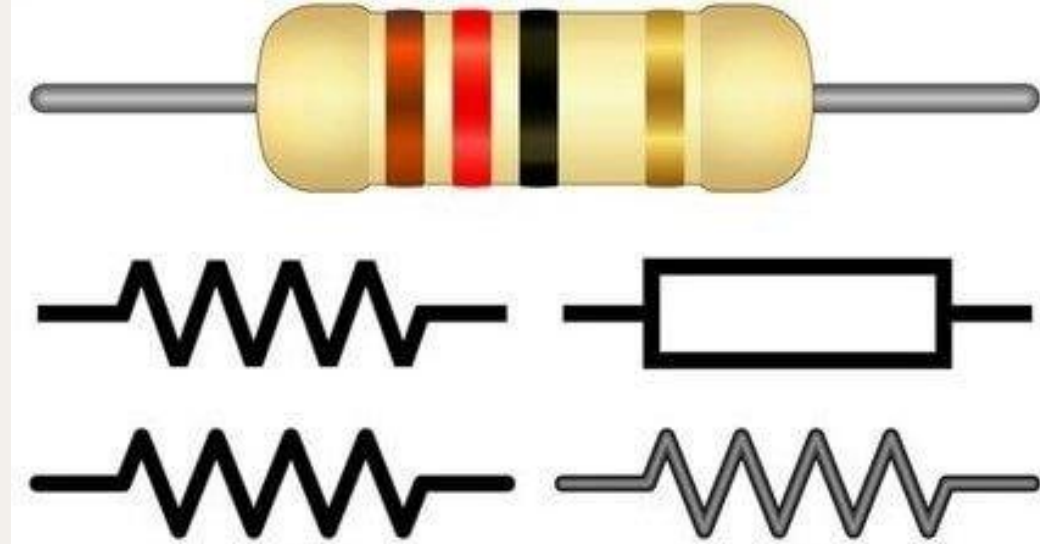
# Electricity

- A – current (amps) – how many flows
  - V – voltage (potential difference) – that's why it flows
  - R – resistance (ohm  $\Omega$ ) – that's why it slows
- 
- Flow of electrons.

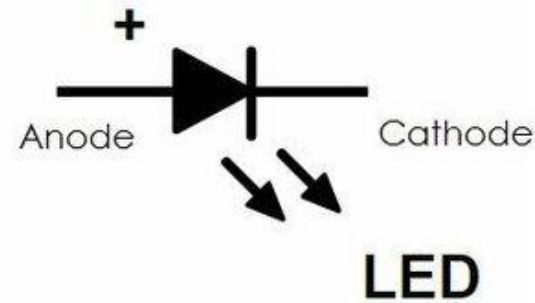
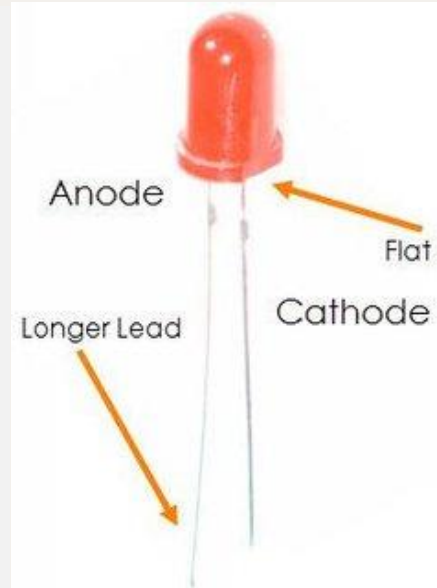
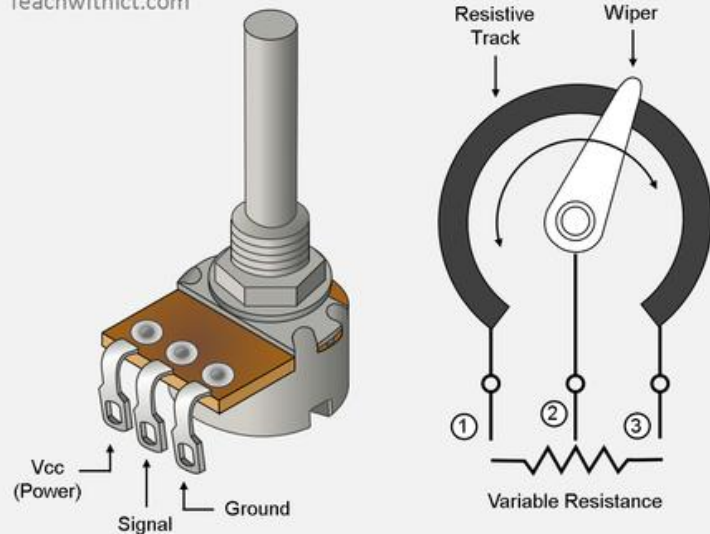


# Electrical components

- Resistor
- Capacitor
- LED
- Potentiometer



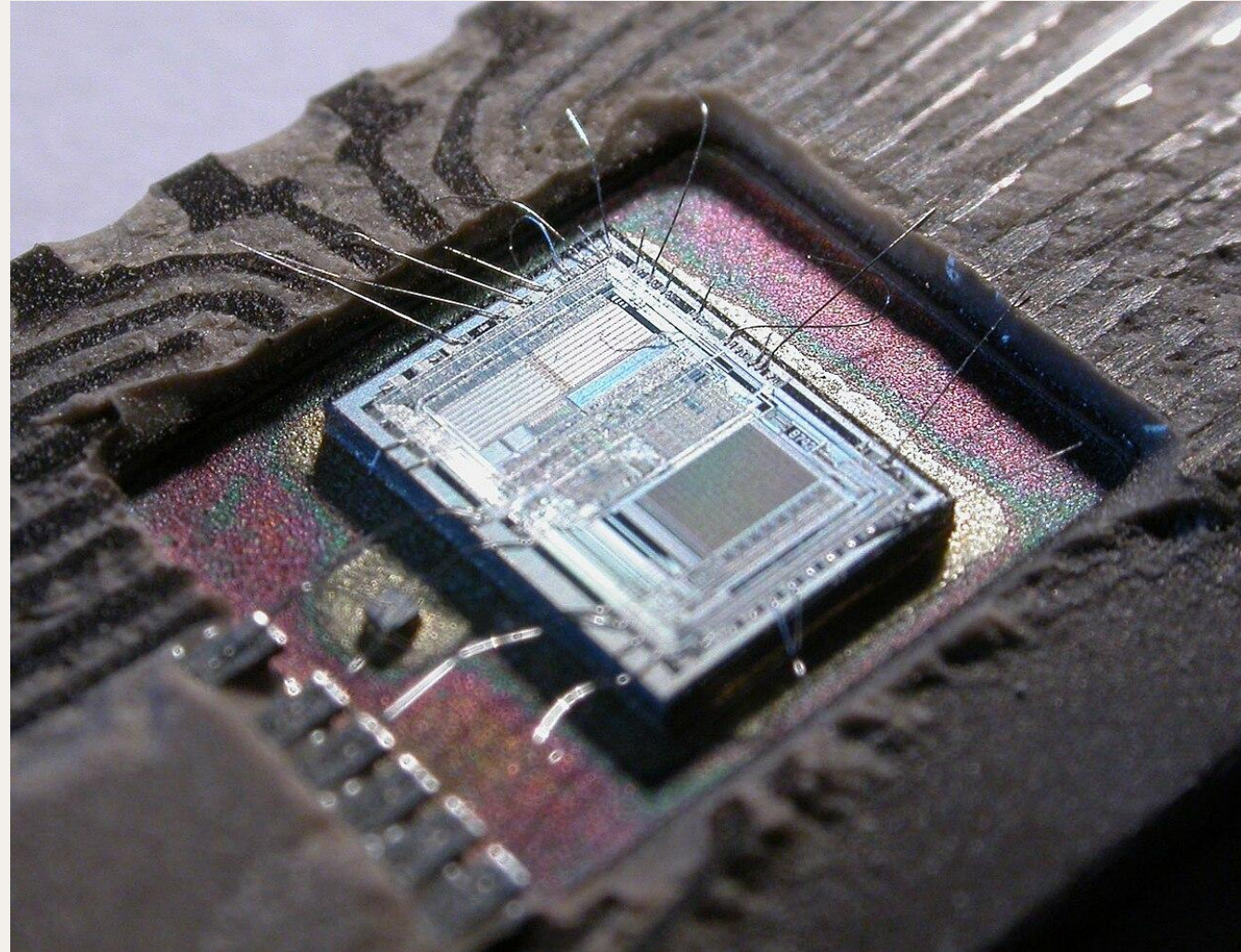
Teachwithict.com





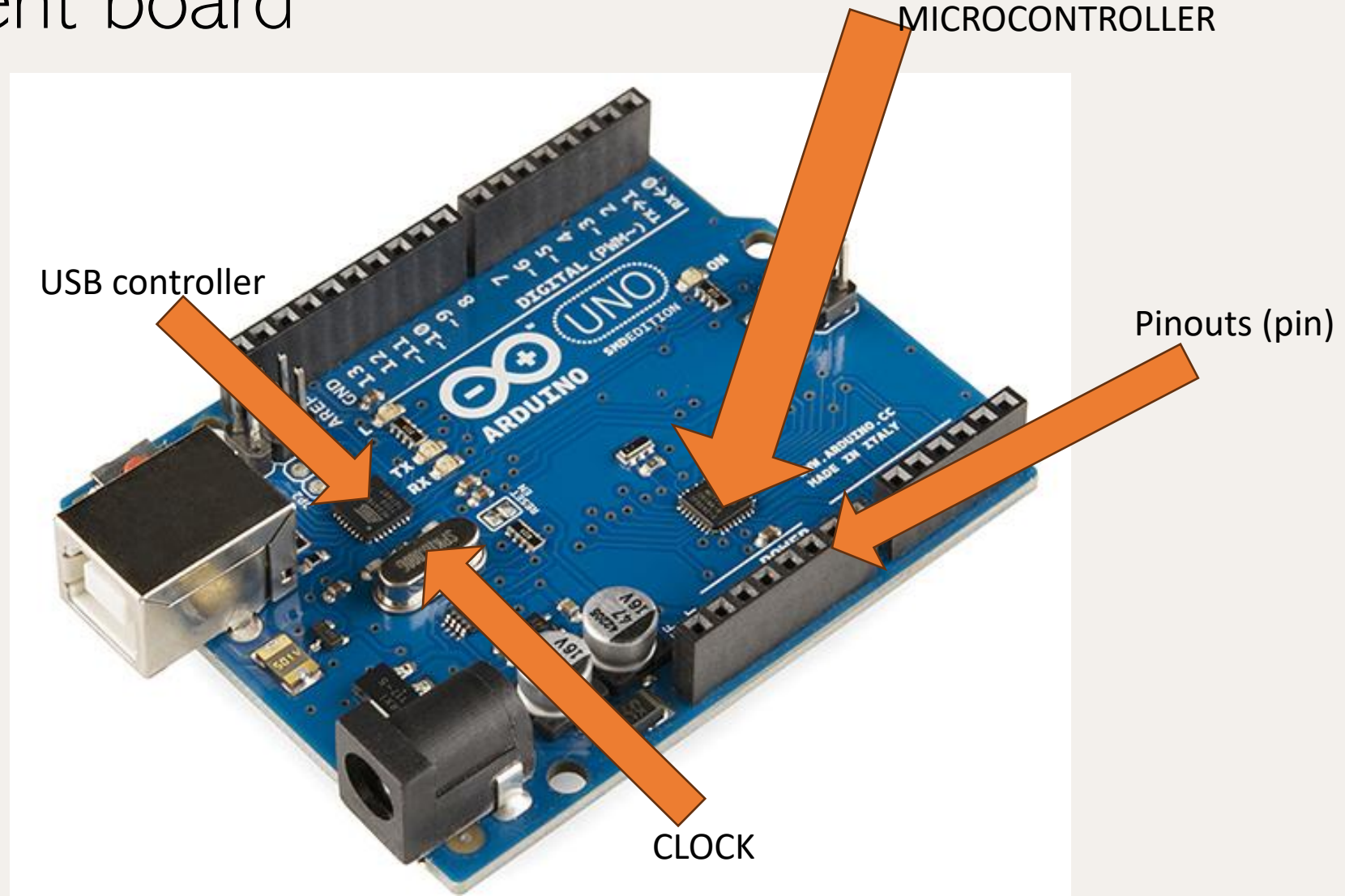
# What is a microcontroller

- MCU
  - Processor cores
  - Memory
  - I/O
  - RAM
- Embedded applications



# What will we use?

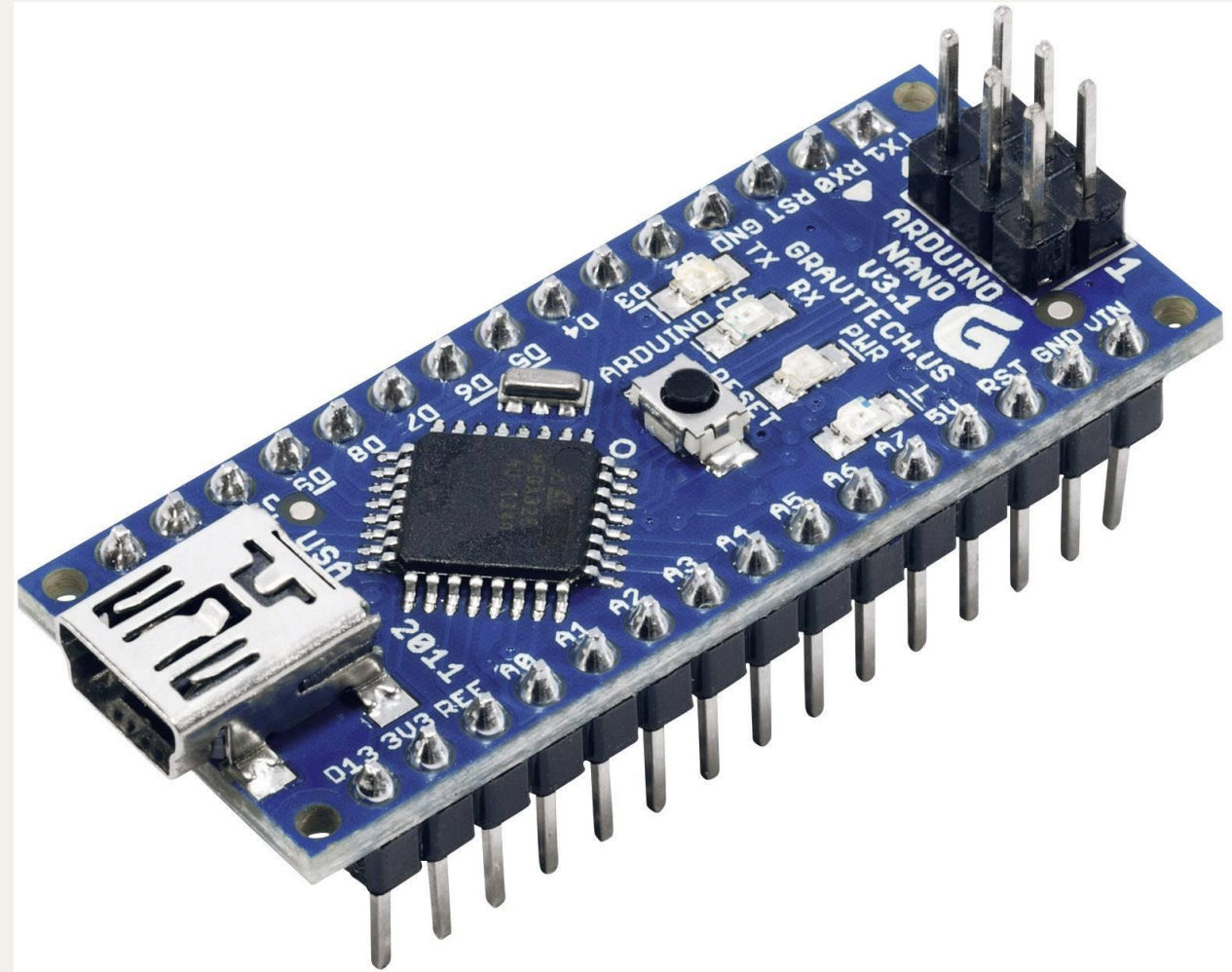
- Development board





# Arduino

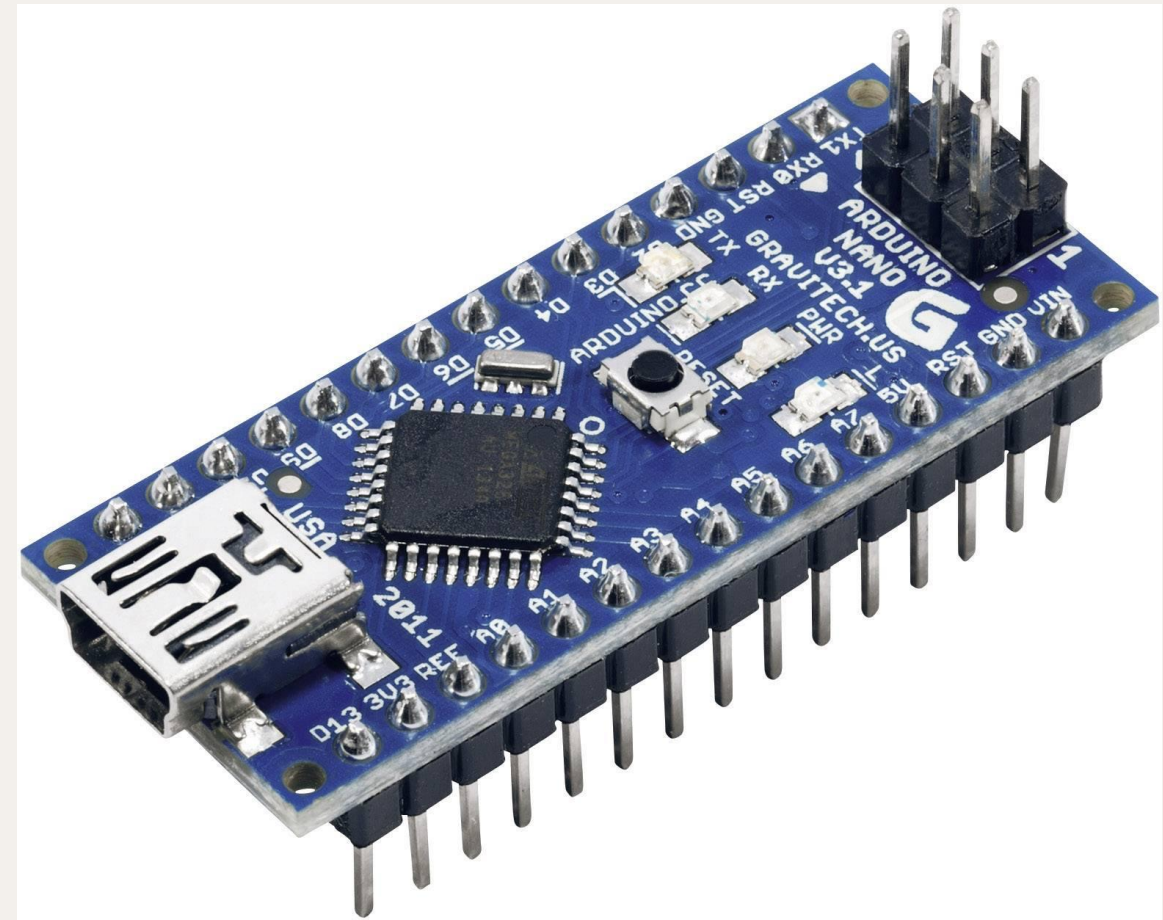
- Open-source hardware company
- Qualcomm exit
- ATMEL based boards
  - UNO
  - Leonardo
  - Nano





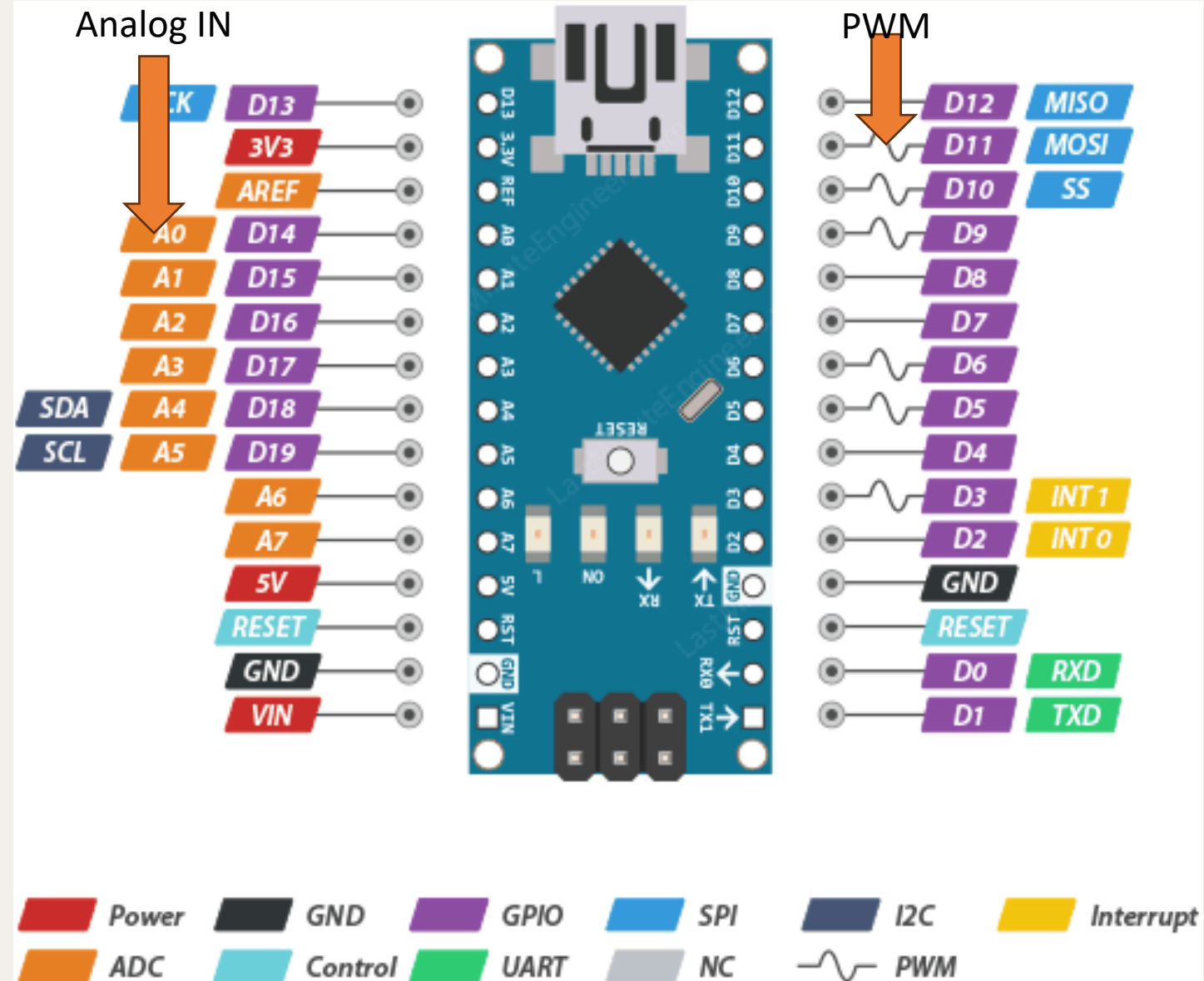
# Arduino nano

- ATmega328/ATmega168
- Program memory size: 16kB
- Data RAM: 1kB
- Clock frequency: 20MHz
- ROM size: 512B
- Chinese clone



# Arduino nano

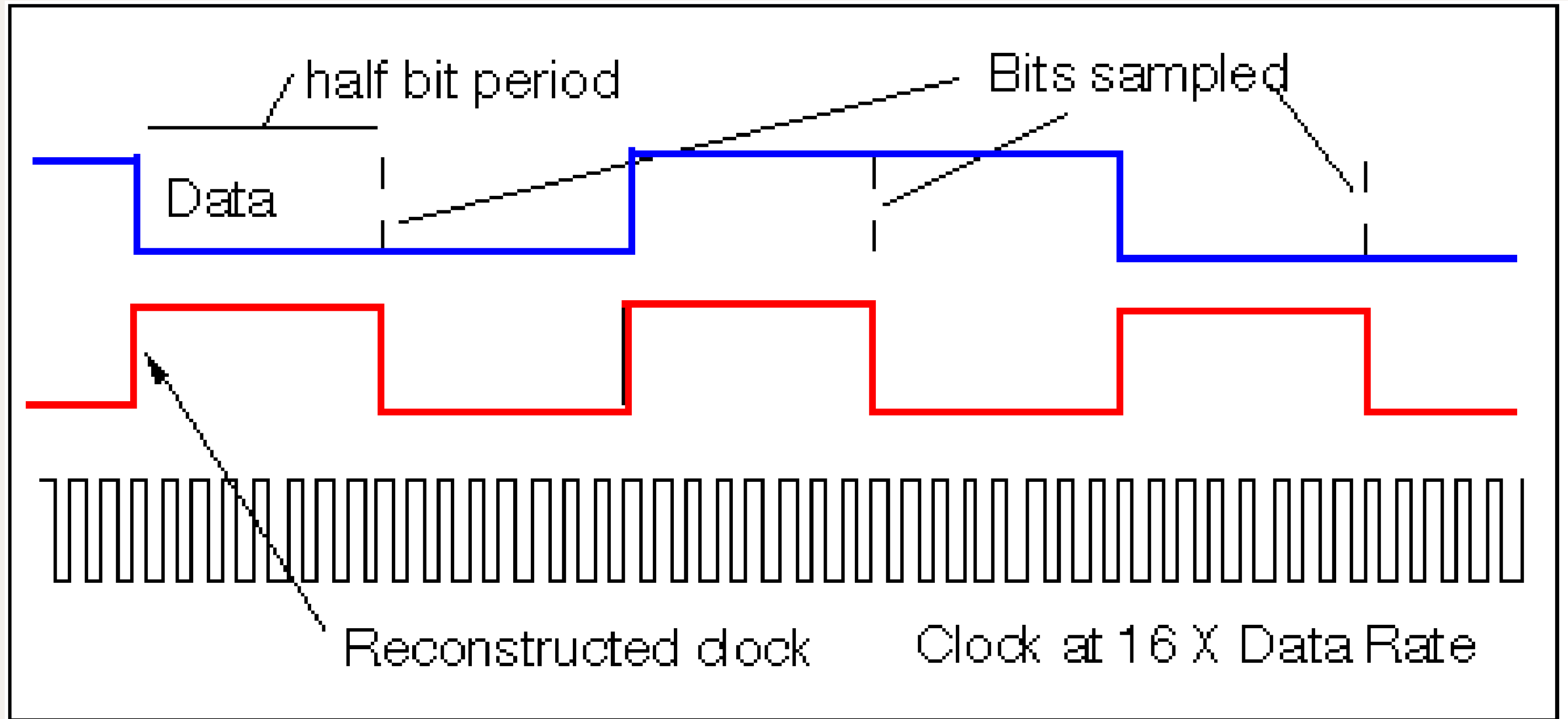
- Pinouts



# Interface – Arduino IDE

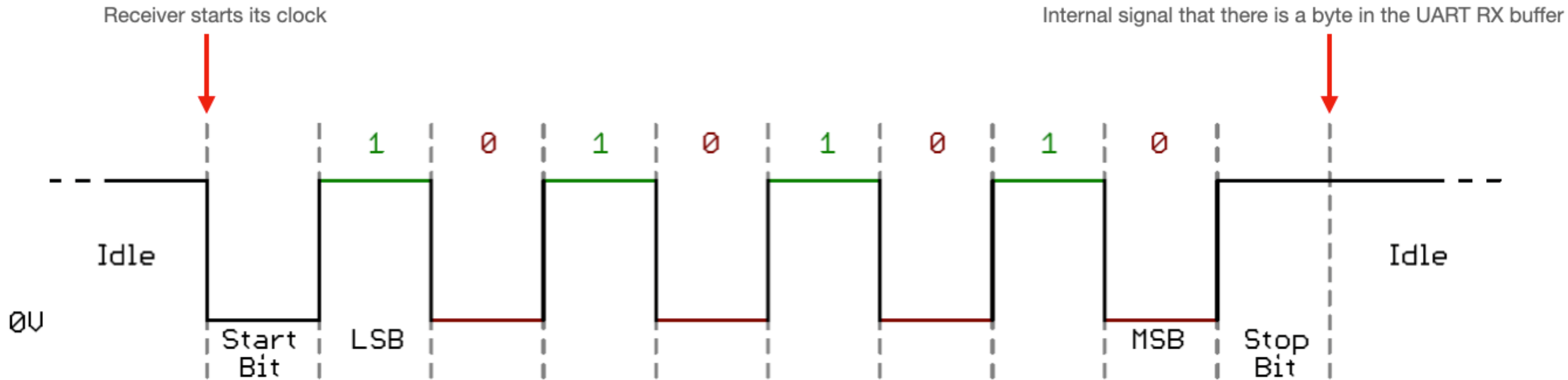
- Interface for programming
  - Lightweight IDE
  - Build system
  - Upload system
- Different hardware different architecture
  - arm-gcc, avr-gcc
- **`#include <Arduino.h>`**

# Async communication protocols



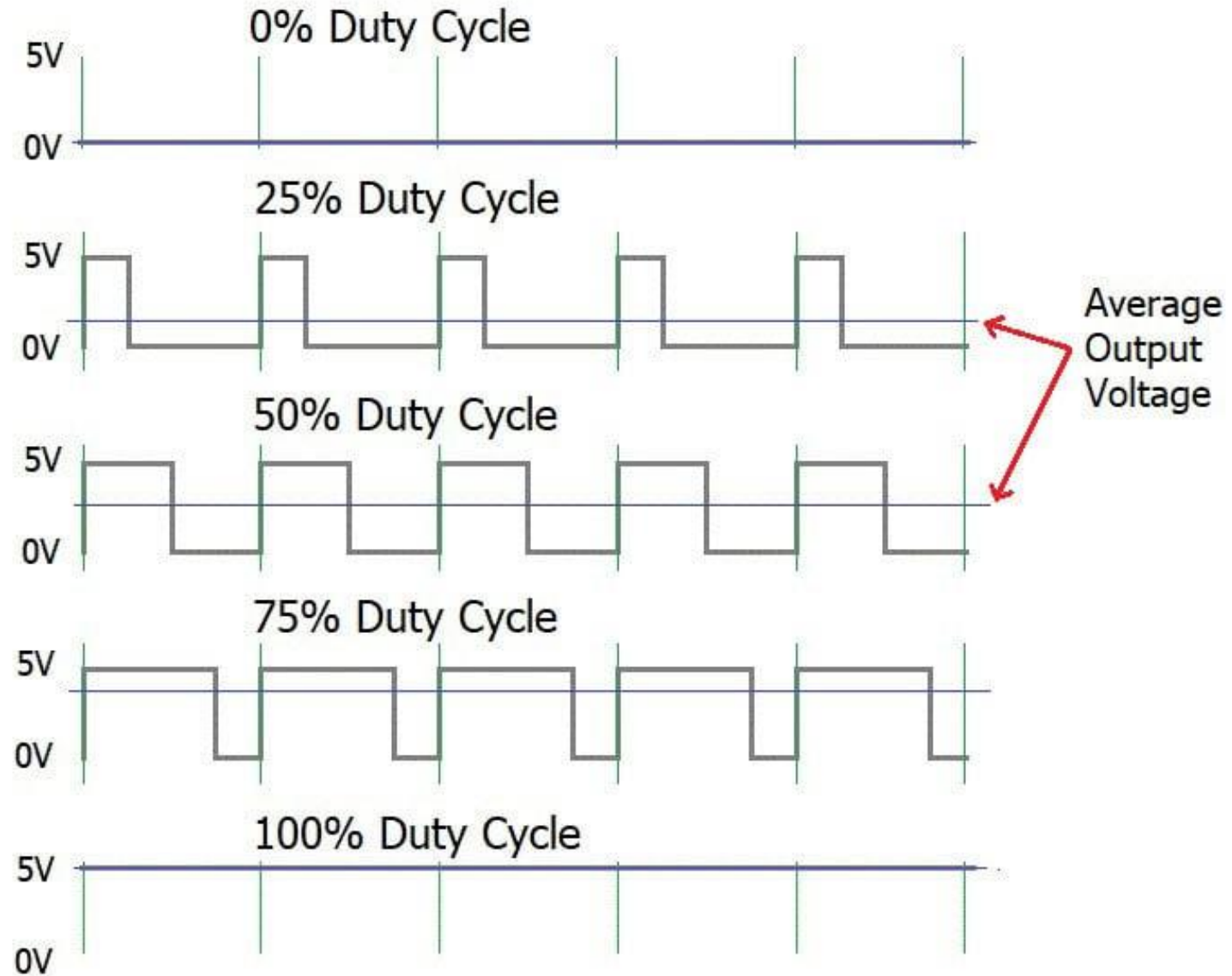


# Interface – serial communication

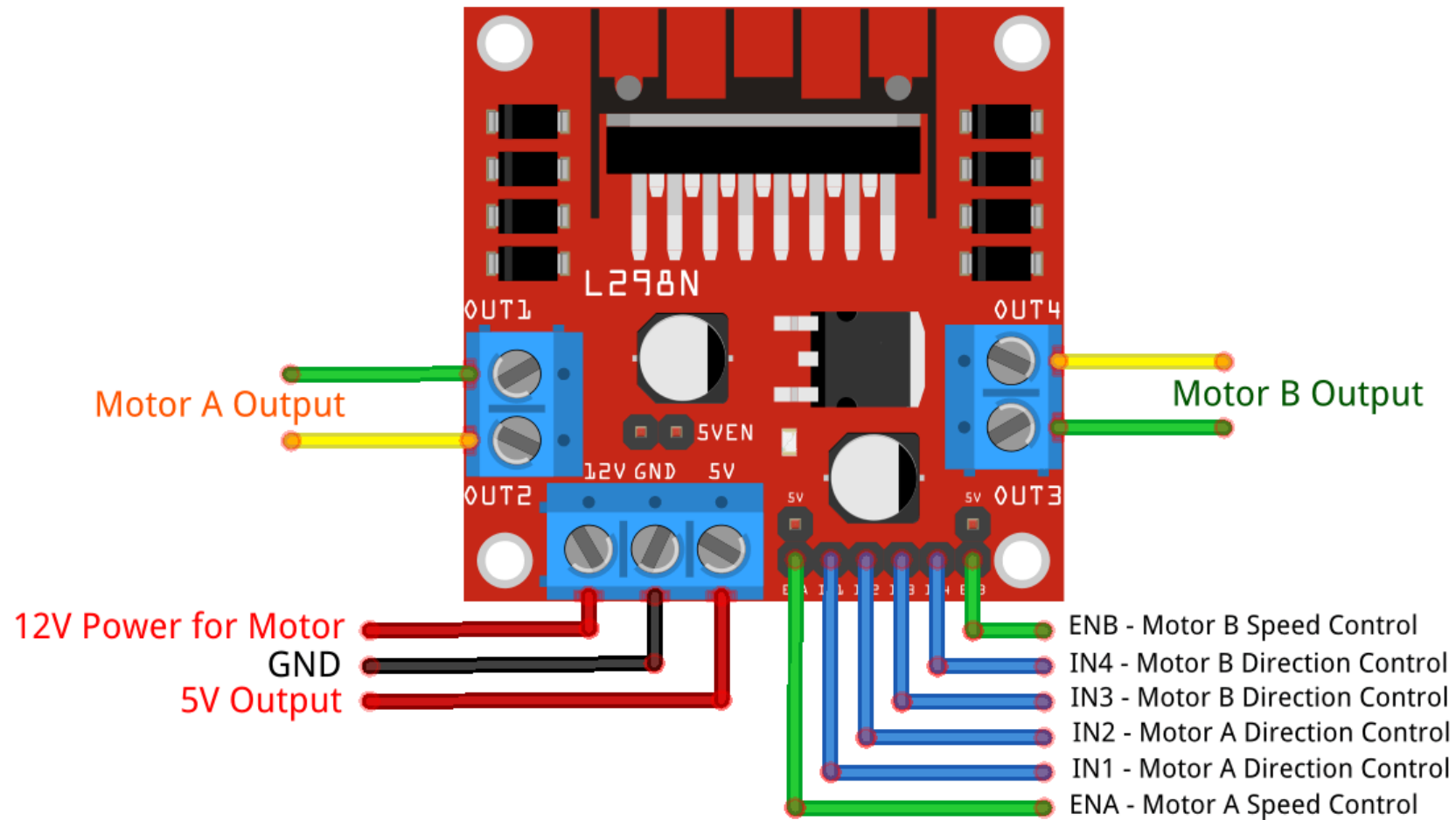


# PWM – pulse width modulation

- We can't do analog
- Let's fake it!
- (motors can't take it)

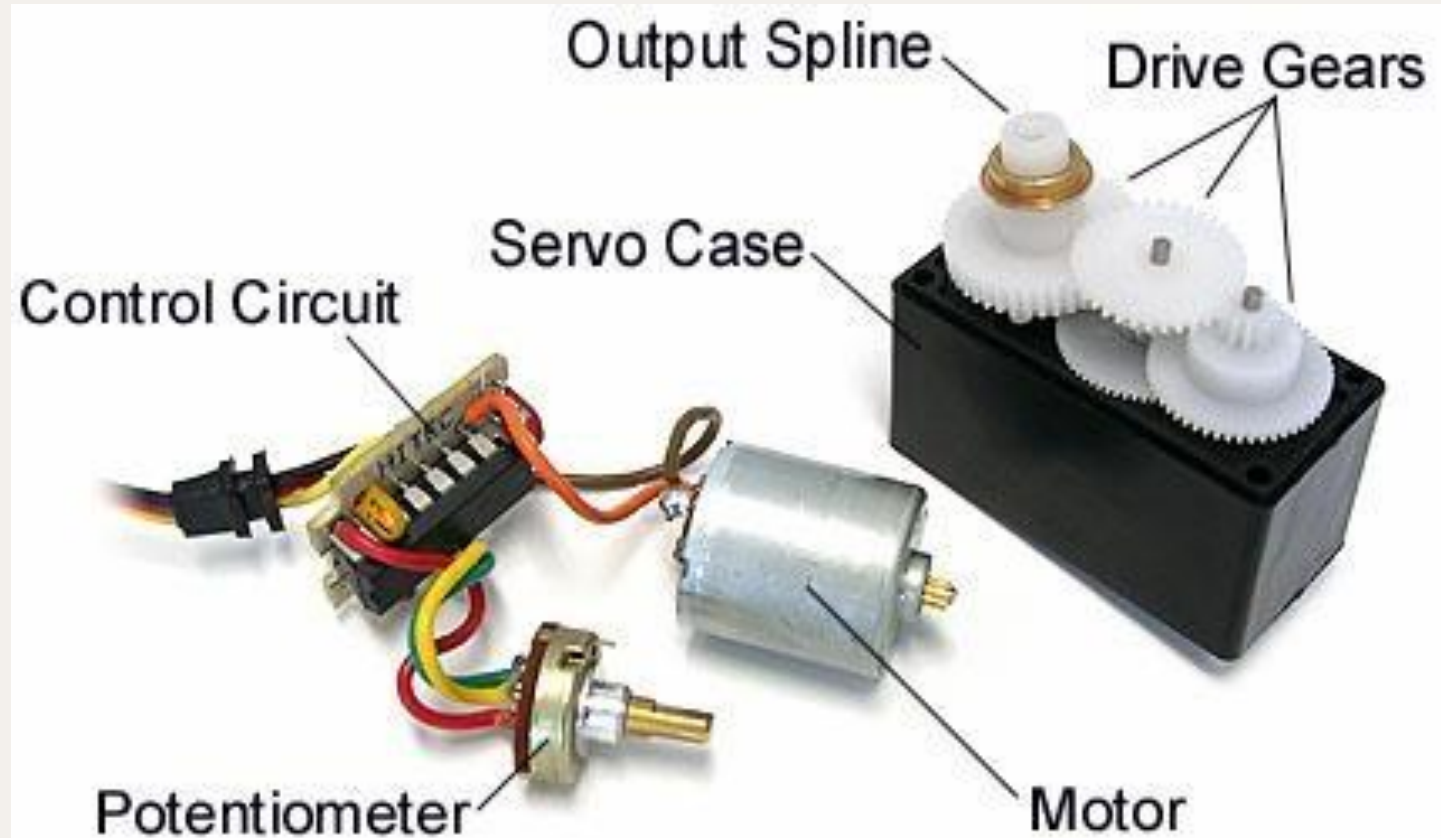


# Motor controller



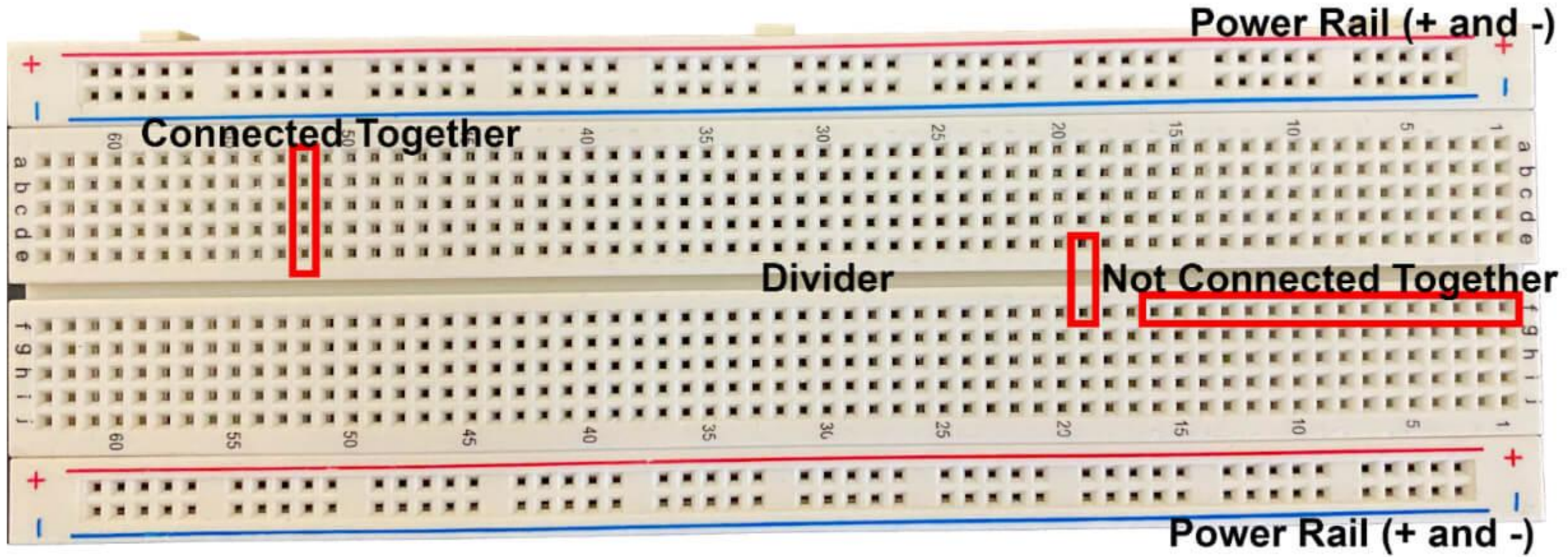
# Microservo motor

- Angle
- PWM





# Breadboard



# Programming basics

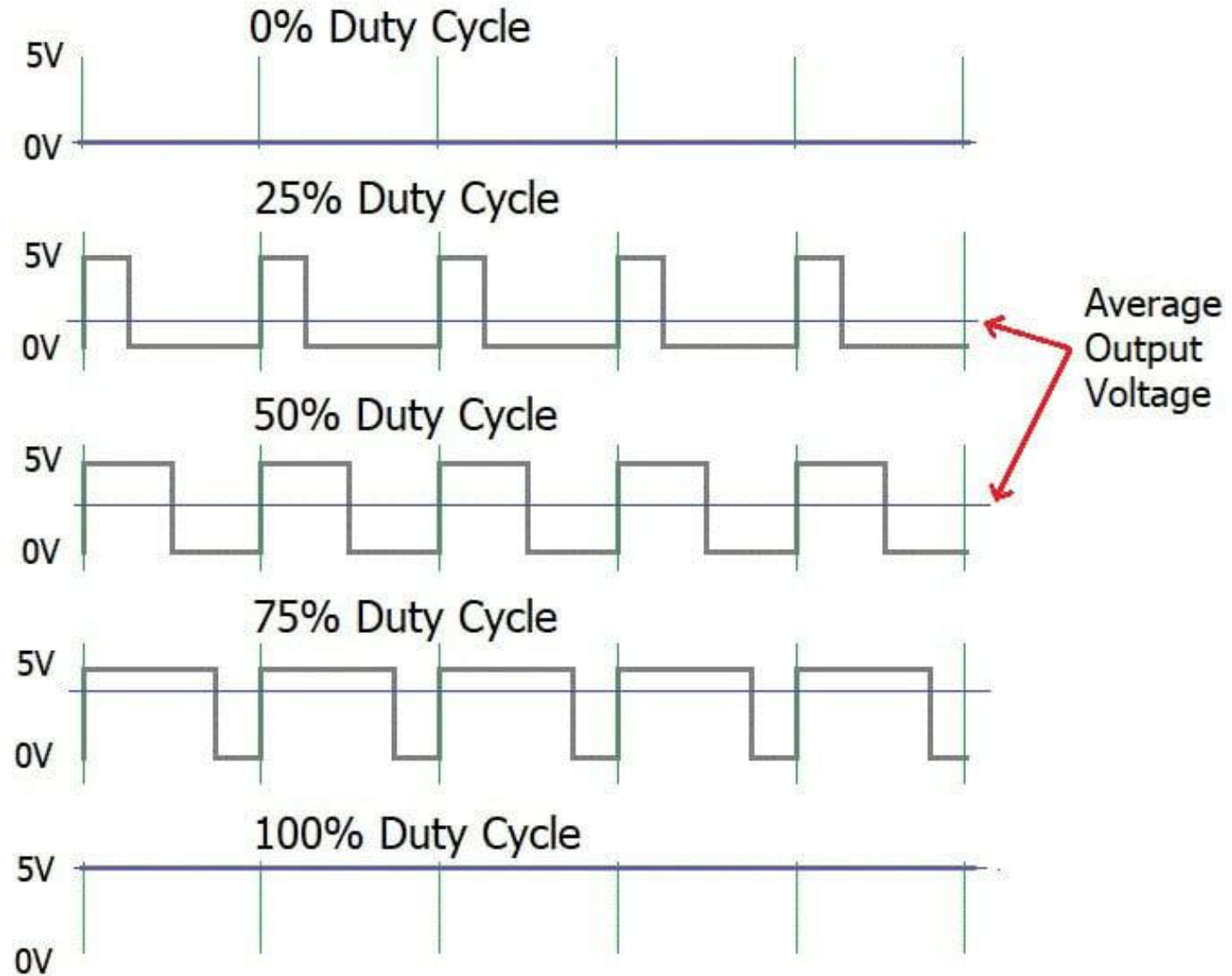
- Variable (has a type and a value)
- Function (has input(s) and output(s))
- Syntax
  - close everything [ { (
  - ; after every line
  - Space doesn't matter (if it's not inside a word)
- Neither enter

# Arduino functions

- `void setup()` – once on every start
- `void loop()` – runs constantly
- `delay([waiting in ms])` – waits till the parameter
- `pinMode(pin, OUTPUT/INPUT)` – sets a pin in/output
- `digitalWrite(pin, 1 / 0)` – turns in or out a pin
- `analogWrite(pin, 0-255)` – writes a PWM signal

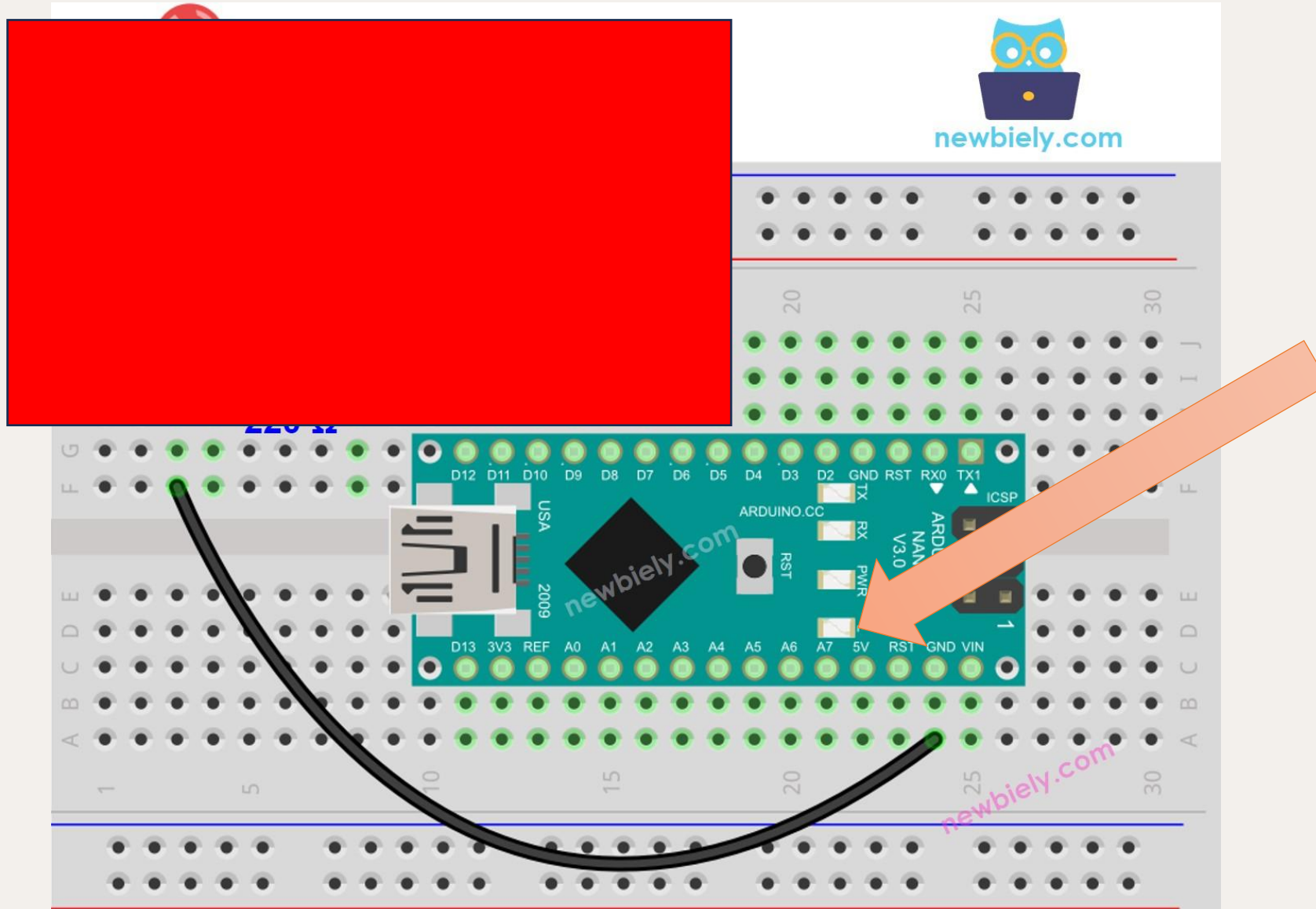
# PWM – pulse width modulation

- We can't do analog
- Let's fake it!
- (motors can't take it)





# First code – blink a LED

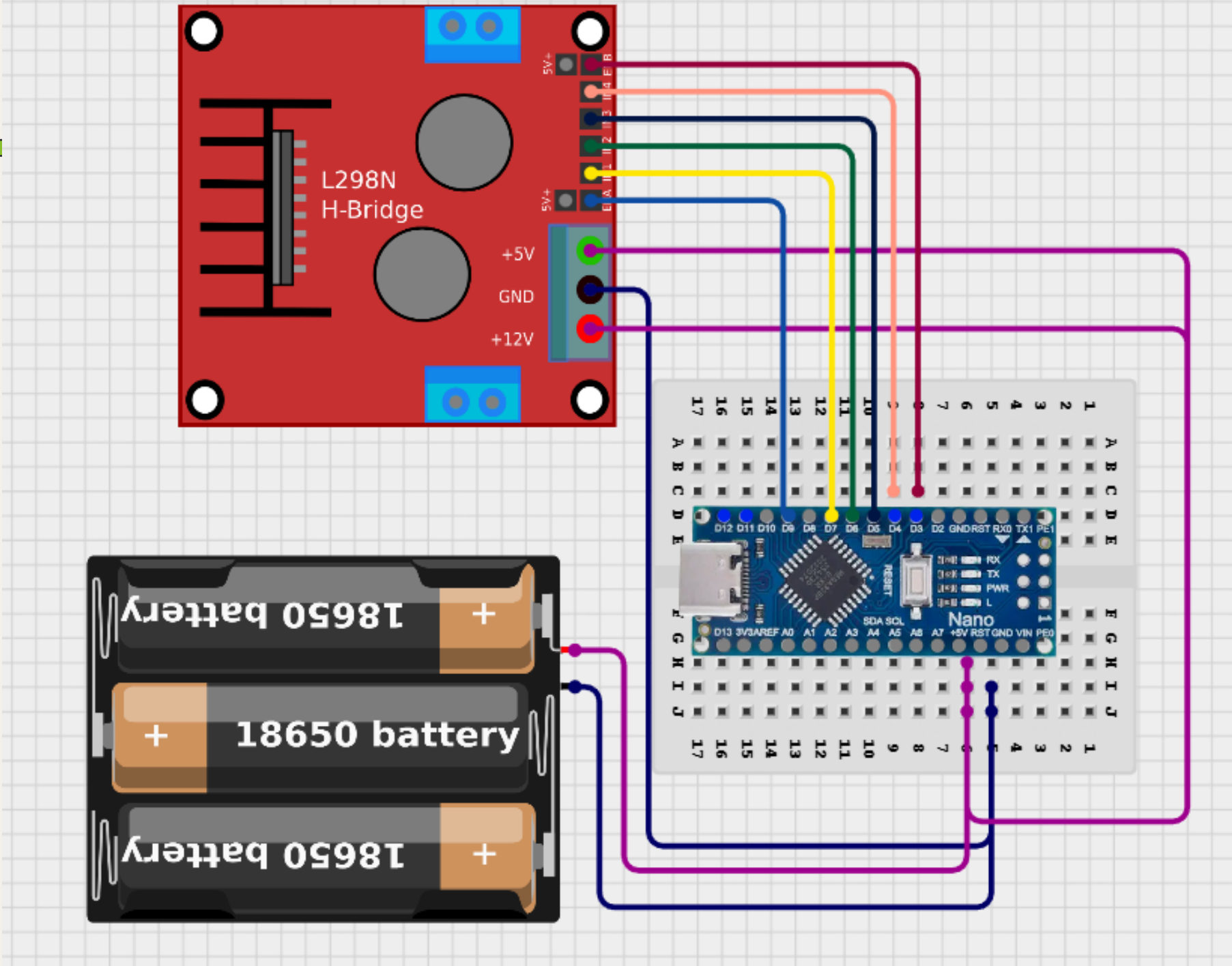


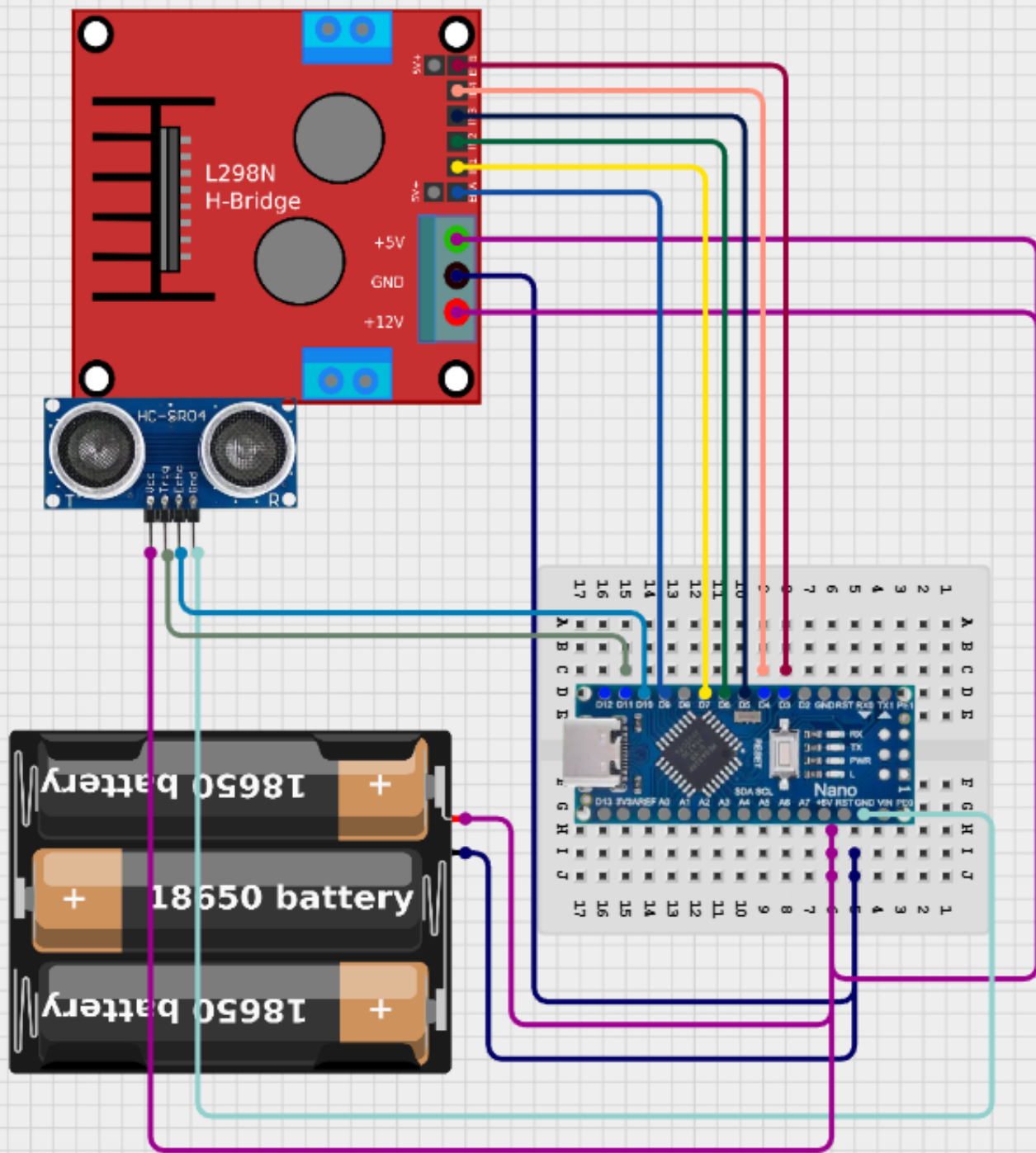
# Make the wiring!

Do not connect – and +  
GND and 5v  
GND and 3v3  
GND and VIN  
OR



Arduino pin	Connected
D3	L298N ENB
D4	L298N IN4
D5	L298N IN3
D6	L298N IN2
D7	L298N IN1
D9	ENA
D10	HC-SR04 echo
D11	HC-SR04 trig
5V	Battery +
5V	L298N +5V
5V	L298N +12V
GND	Battery GND
GND	L298N GND
GND	HC-SR04 GND
5V	HC-SR04 Vcc



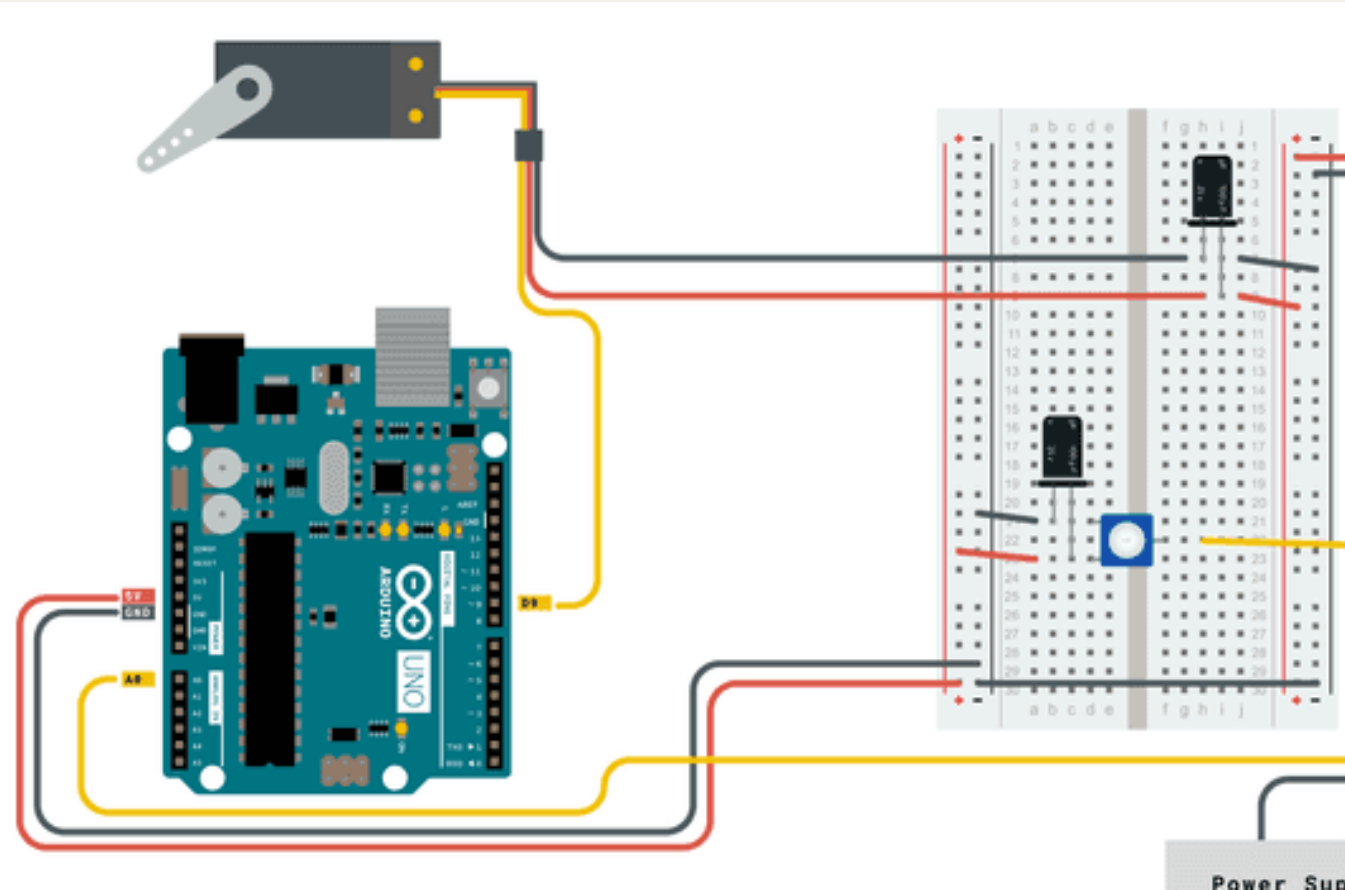


Arduino pin	Connected
D3	L298N ENB
D4	L298N IN4
D5	L298N IN3
D6	L298N IN2
D7	L298N IN1
D9	ENA
D10	HC-SR04 echo
D11	HC-SR04 trig
5V	Battery +
5V	L298N +5V
5V	L298N +12V
GND	Battery GND
GND	L298N GND
GND	HC-SR04 GND
5V	HC-SR04 Vcc



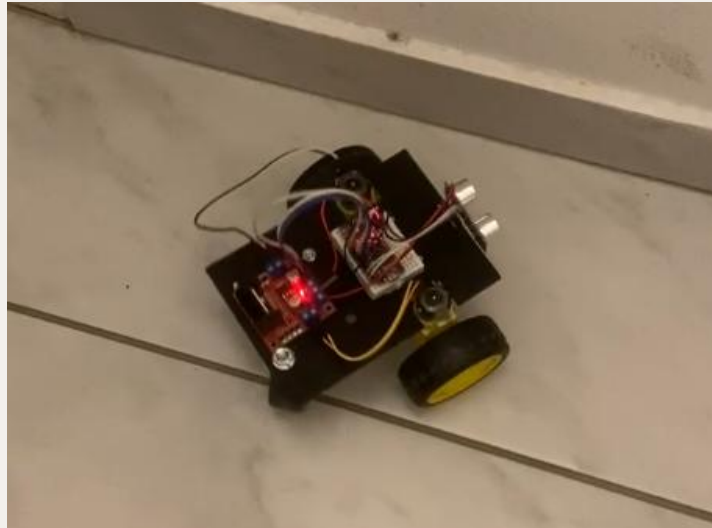
# Servo functions

- Connect to a PWM pin (and + & -)
- `#include <Servo.h>`
- `Servo myServo(pinNum)`
- `myServo.write(angle)`



# If we have time...

## Overviews



- Onshape -> web-based CAD software

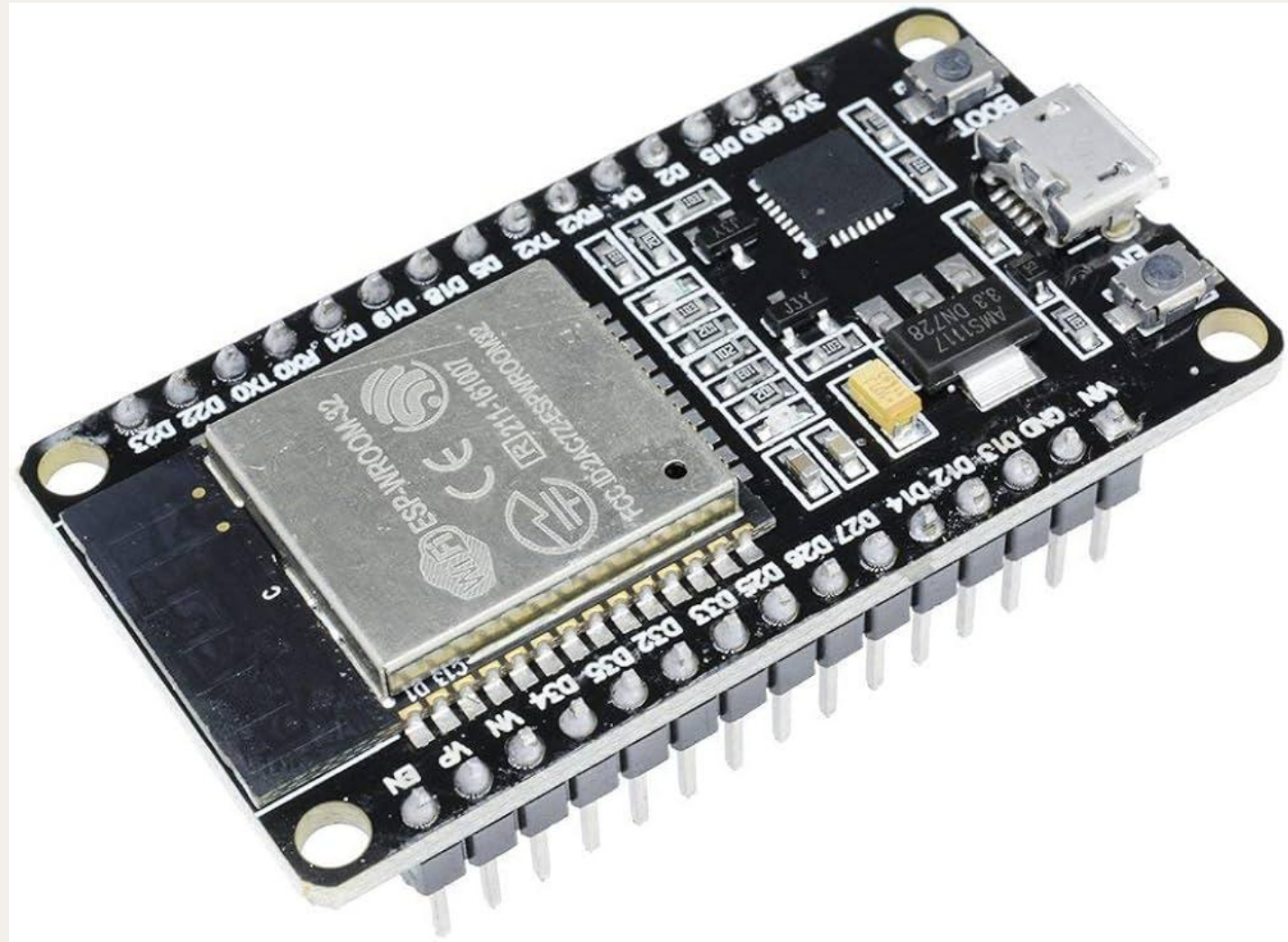
<https://cad.onshape.com/signin>

# Electrical design

- KiCad
- PCB and circuit design
- Schematic
- Board

# Good to know – ESP

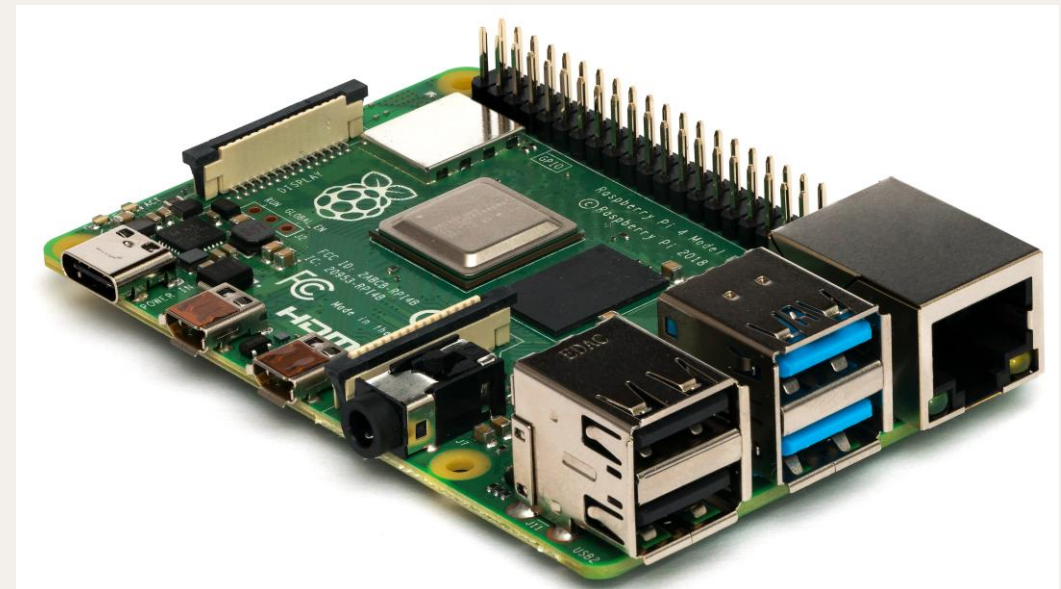
- WiFi capable microcontroller
- Bluetooth or WiFi
- Dev boards





# Good to know – Raspberry pi

- COMPUTER, not microcontroller
- Linux system
- GPIO pins
- Complex logic – written in any language



# How to build at home?

- Github repo
- BOM – buy
- 3D model – print (or make it from wood)
- Code!
- Buy other stuff and experiment!

<https://github.com/KerteszDomokos/robotcar-workshop>



# Thanks!

Ask me! [kerteszdomokos@gmail.com](mailto:kerteszdomokos@gmail.com)

