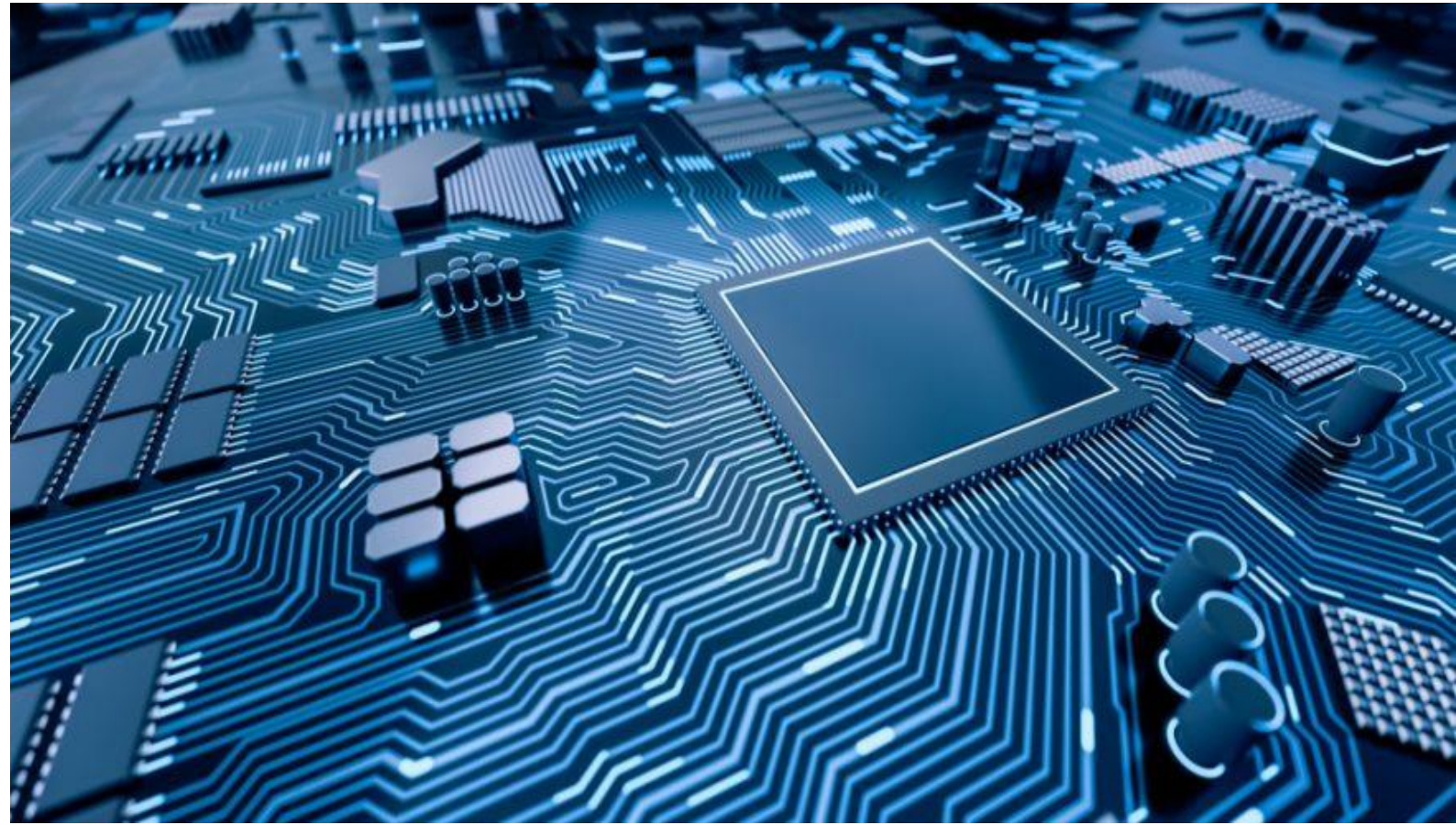




VNU – University of Engineering and Technology



Faculty of Electronics & Telecommunications

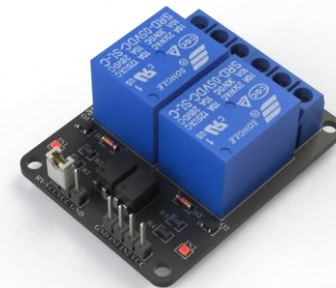


Internet of Things: Practice

- Lecturer: Nguyễn Ngọc Tân
- Email: nguyen.tan17089@gmail.com
- Phone: 0904.183.123

CONTROLLING IOT ACTUATORS

- ❖ First Rules
- ❖ Getting Started with an ESP8266
- ❖ Playing with LEDs
- ❖ Controlling DC Motor
- ❖ Controlling via a Relay



First Rules!

❖ You might actually damage hardware

- Double check when you do wiring
- Do not connect any unintended GPIO pins, or you will short circuit the ESP8266
- During wiring, Power off the ESP8266 (or unplugged microUSB connection)
- Do not touch with wet fingers

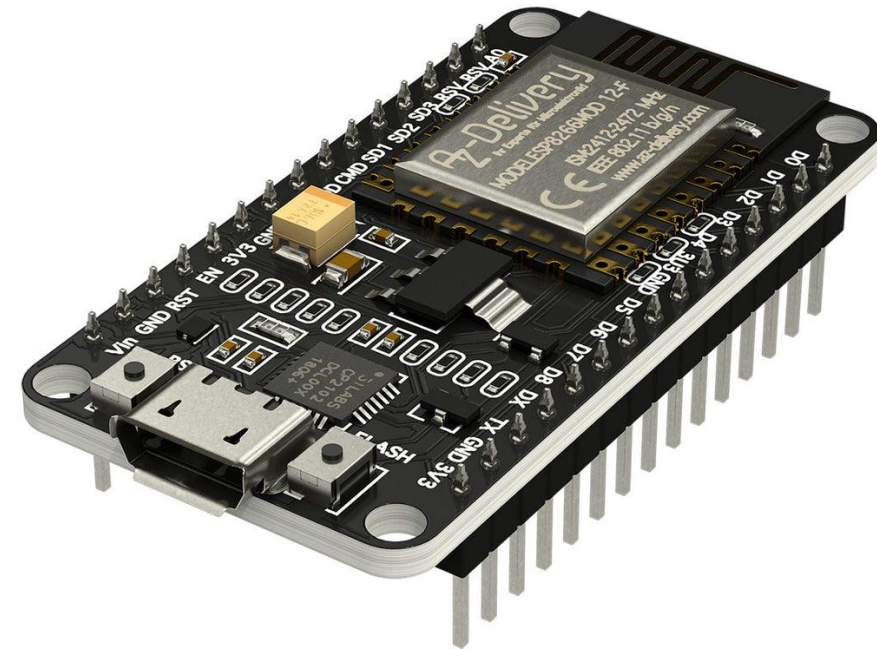
❖ Think "why", and consult hardware reference for better understanding

- Reading hardware references is a must-have skill
- You don't have to read a reference thoroughly at the beginning (Later, you will)

Getting Started with an ESP8266

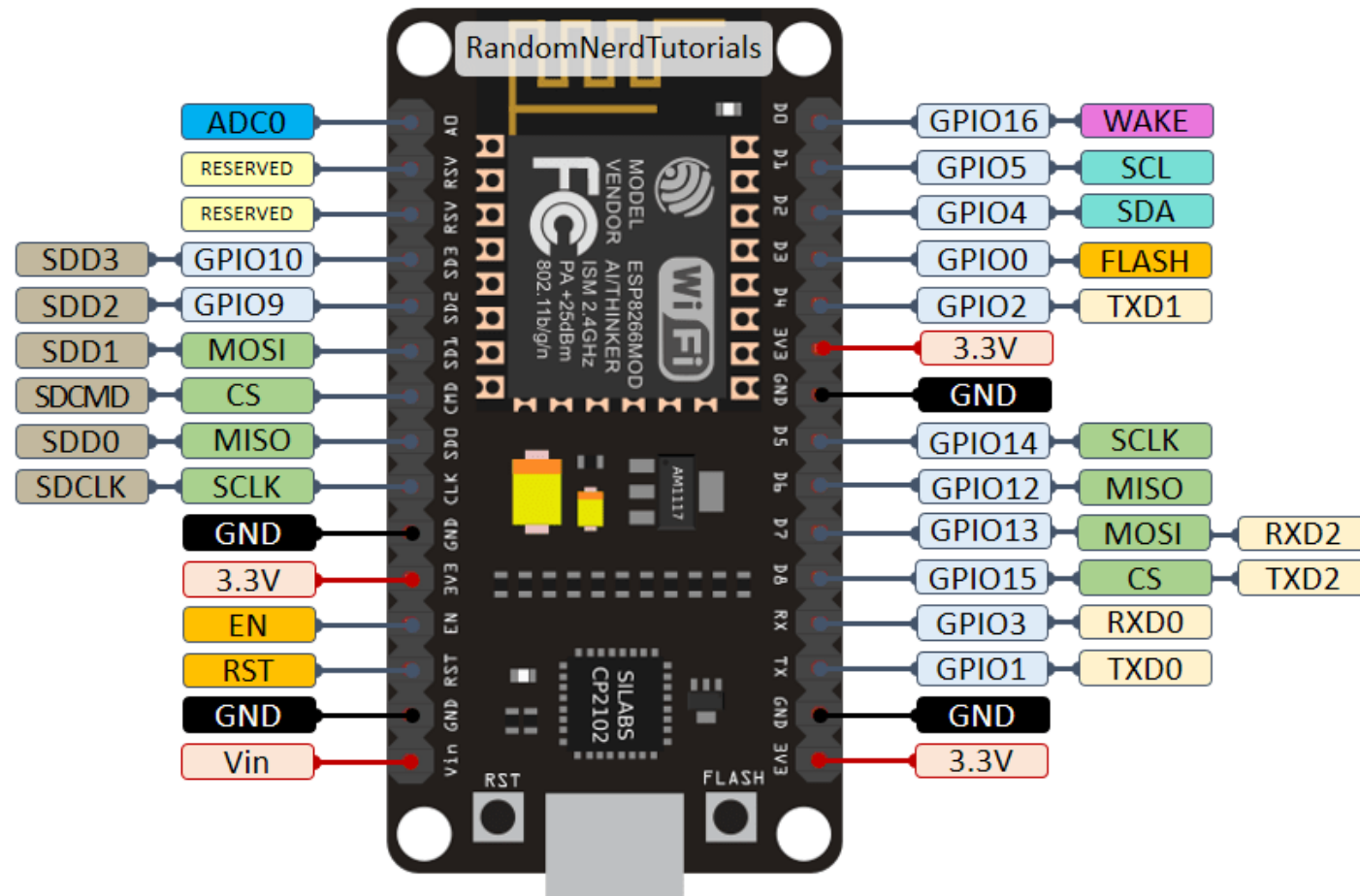
❖ NodeMCU ESP8266

- Model: NodeMCU ESP8266 ESP-12E
- Operating Voltage: 3.3V
- Input Voltage (recommended): 4.5V → 10V
- Digital I/O Pins: 11
- Analog Input Pins: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- Wi-Fi built in: 802.11b/g/n



Getting Started with an ESP8266

❖ PIN Diagram



Getting Started with an ESP8266

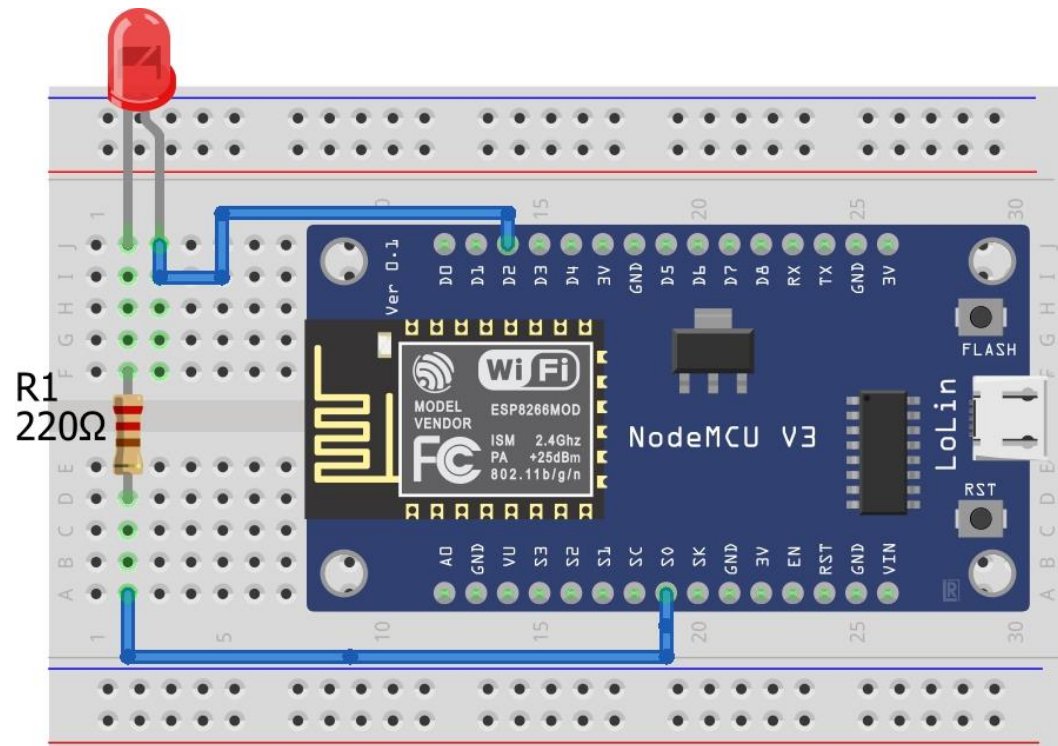
❖ GPIO (General-Purpose Input/Output)

- A general-purpose input/output is an uncommitted digital signal pin on an integrated circuit or electronic circuit board which may be used as an input or output, or both, and is controllable by software
- Maximum current of GPIO pins is 12mA
- All GPIO pins are pulled HIGH
- GPIO01 & GPIO03 are TX and RX of UART0 which is used to flash the ESP8266 → Should not use when flashing
- GPIO0, GPIO2, GPIO15 are used for mode configuration of ESP8266 → Should not use when flashing
- GPIO9 & GPIO10 are used to connect with External Flash of ESP8266 → Should not use
- GPIO4 (D2), GPIO5 (D1), GPIO12 (D6), GPIO13 (D7), GPIO14 (D5), & GPIO16 (D0) can be used for any purpose

Playing with LEDs

❖ Circuit diagram

- Connect a LED to ESP8266 as the following circuit diagram.



Playing with LEDs

❖ Programming

```
Blink | Arduino 1.8.19
File Edit Sketch Tools Help

Blink $

/*
  ESP8266 Blink
  Blink the red LED on the ESP-01 module
  */

//#define LED D2;    // D2 ~ GPIO4
//#define LED 4;
//int LED = D2;
int LED = 4;

void setup() {
  pinMode(LED, OUTPUT);    // Initialize the LED pin as an output
}

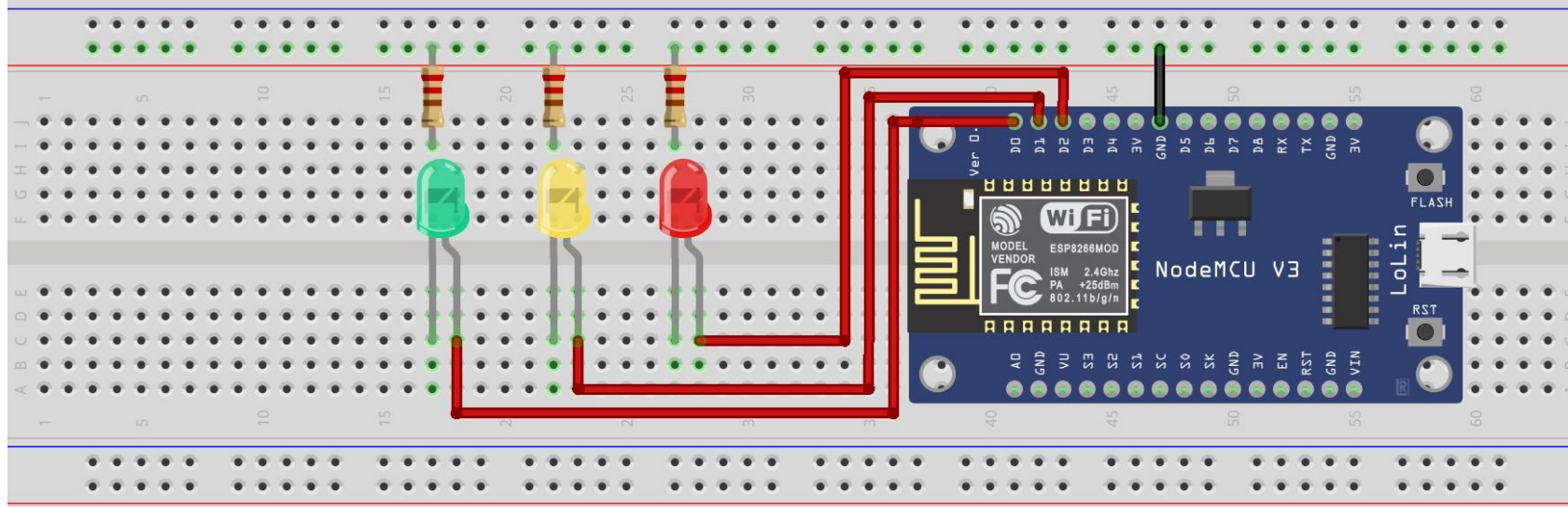
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH);    // Turn the LED on
  delay(2000);                // Wait for two seconds (to demonstrate the active low LED)
  digitalWrite(LED, LOW);    // Turn the LED off by making the voltage HIGH
  delay(1000);                // Wait for a second
}

Done compiling
Global variables use 27984 bytes (34%) of dynamic memory, leaving 53936 bytes for local variables. Maximum is 81920 bytes.
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL cipher (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, HMB (FS:2MB OTA~1010KB), 2. v2 Lower Memory, Disabled, None, Only Sketch, 115200
```


Playing with LEDs

❖ Explore more...

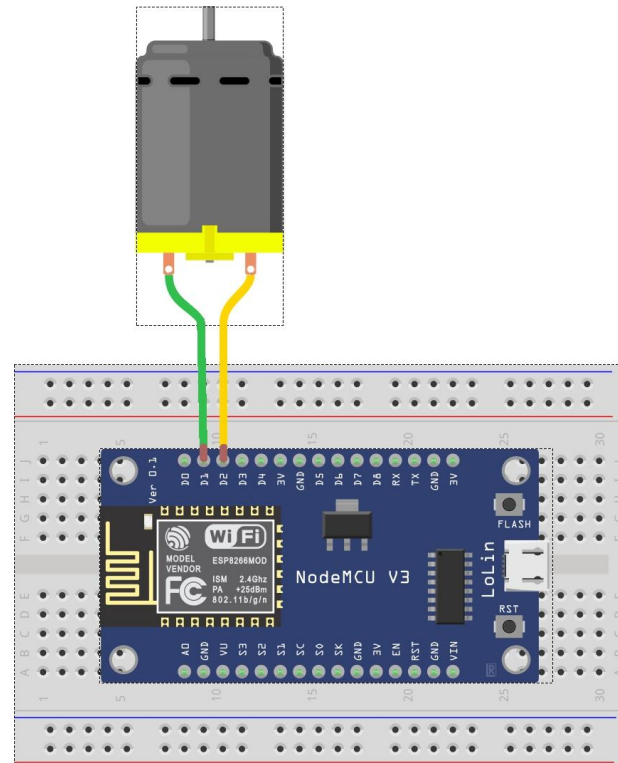
- Build a traffic-light system with three LEDs in green, yellow, and red colors.



Controlling DC motor

❖ Circuit diagram

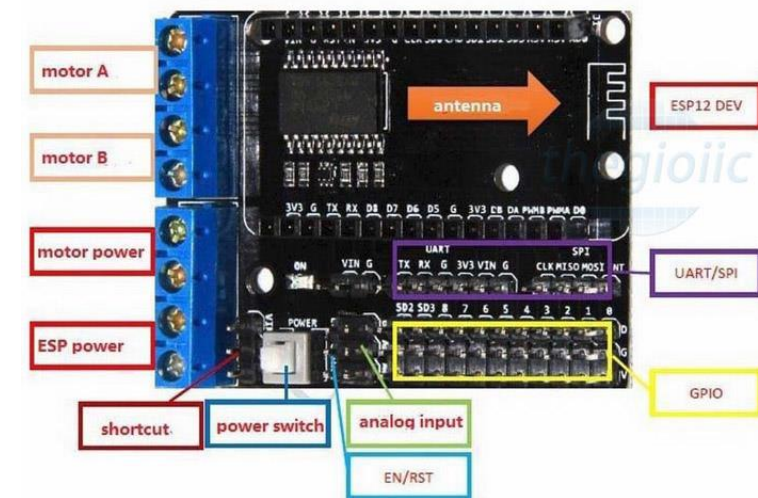
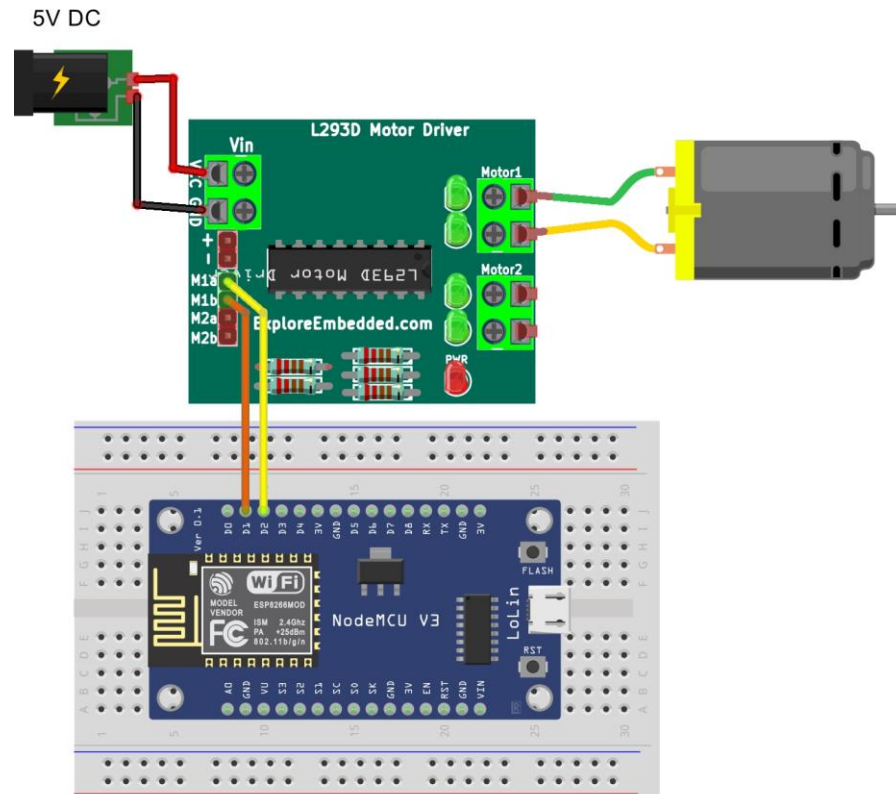
- Connect a DC motor to ESP8266 as the following circuit diagram.



Controlling DC motor

❖ Oops! Something's wrong?

- Remember the maximum output current of ESP8266 is 12mA. It needs a high-current driver (e.g., L293 driver) to control the DC motor or any actuators requiring high-current.



ESP12E Lua L293D

Controlling via a Relay

❖ Demo Code

```
1293 §  
/*  
    Board pin | NodeMCU GPIO |  Arduino IDE  
    A-        |                |  
    A+        |                |  
    B-        |                |  
    B+        |                |  
*/  
const int pwmMotorA = D1;  
const int dirMotorA = D3;  
int motorSpeed = 100;  
void setup() {  
    Serial.begin(115200);  
    Serial.println();  
    pinMode(pwmMotorA, OUTPUT);  
    pinMode(dirMotorA, OUTPUT);  
    Serial.println("Motor SHield 12E Initialized");  
    delay(3000);  
}  
void loop() {  
    Serial.println("Activate A");  
    digitalWrite(pwmMotorA, motorSpeed);  
    digitalWrite(dirMotorA, LOW);  
    delay(4500);  
    Serial.println("Reverse A");  
    digitalWrite(dirMotorA, HIGH);  
    delay(4500);  
    Serial.println("Stop A");  
    digitalWrite(pwmMotorA, 0);  
    digitalWrite(dirMotorA, LOW);  
    delay(2000);  
}
```

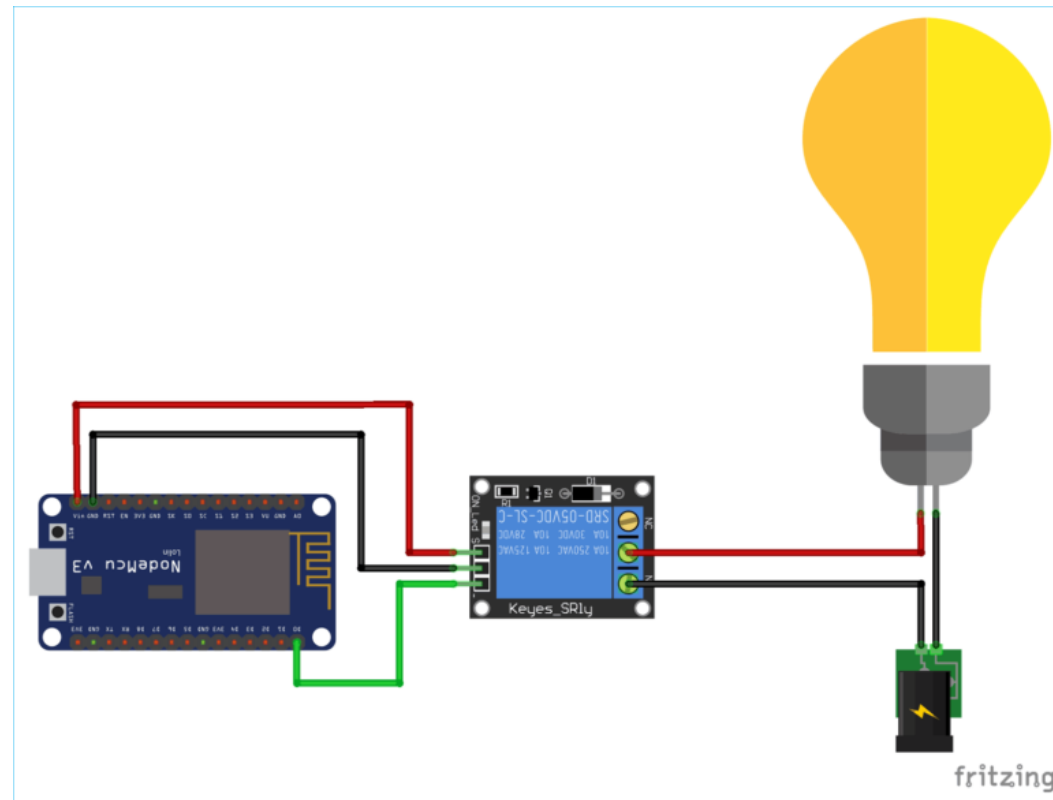
Controlling DC motor

❖ Explore more...

- Control the motor's speed by press buttons ↑↓ on your keyboard.

Controlling via a Relay

❖ Why do we must use a Relay?



Controlling via a Relay

❖ Demo Code

sketch_sep12b

```
const int Relay = D1;

void setup() {
  Serial.begin(115200);
  Serial.println();
  pinMode(Relay , OUTPUT);
  Serial.println("Motor SHield 12E Initialized");
  delay(5000);
}

void loop() {
  Serial.println("Turn On Light!");
  digitalWrite(Relay, HIGH);
  delay(4500);
  Serial.println("Turn Off Light!");
  digitalWrite(Relay, LOW);
  delay(4500);
}
```

THANK YOU!