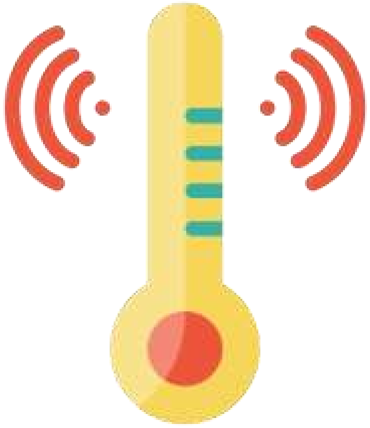
Embedded Systems

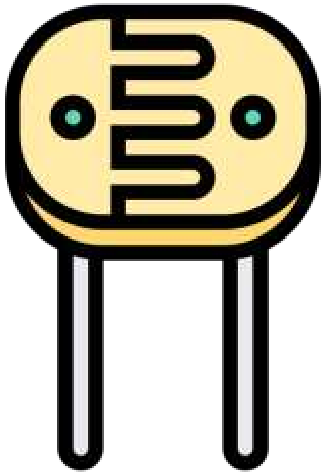
Remote Car Prototype

Abdallah El Ghamry

Sensors

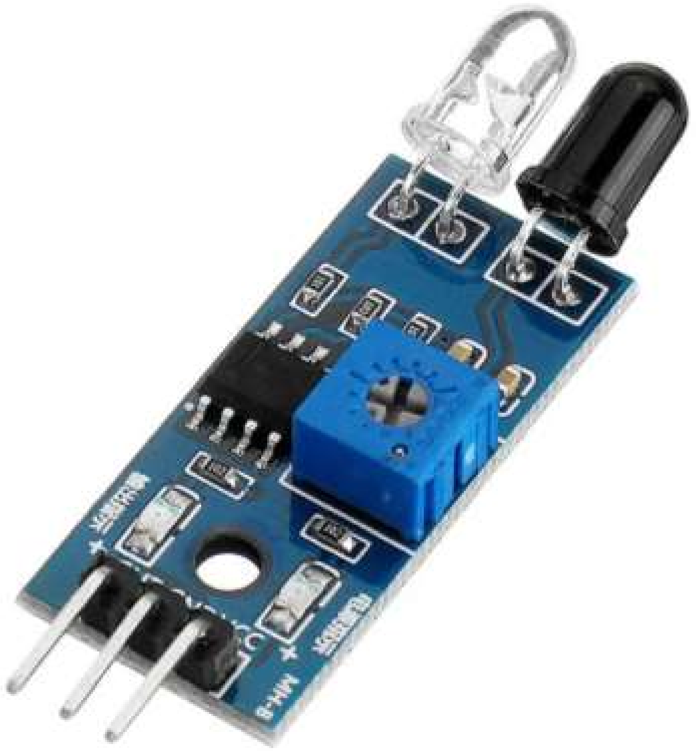
- A sensor is a device that detects some type of input from the physical environment.
- The input can be light, heat, motion, pressure or any number of other environmental phenomena.





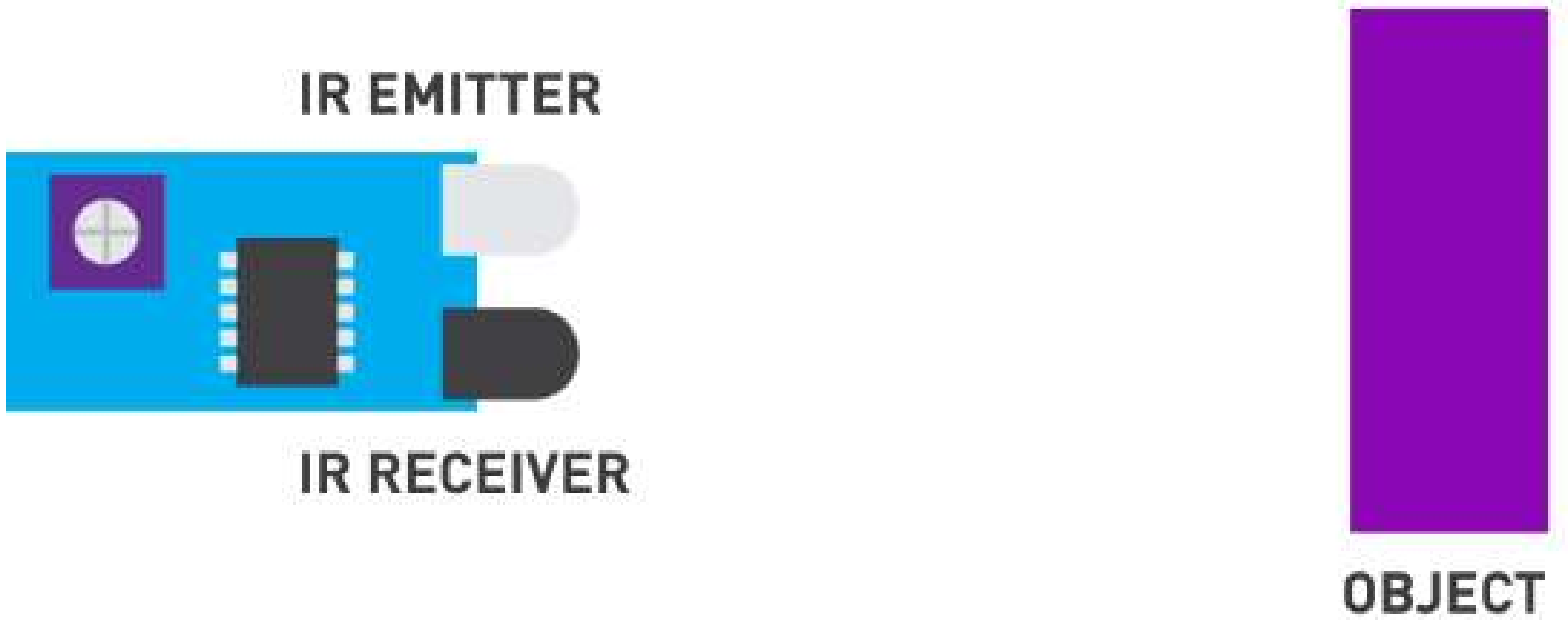
Infrared (IR) Sensor

- IR sensor is a simple Infrared Sensor that is used to detect obstacles.
- This is a multipurpose sensor and is used in many robotic applications like obstacle avoidance robots and line follower robots.

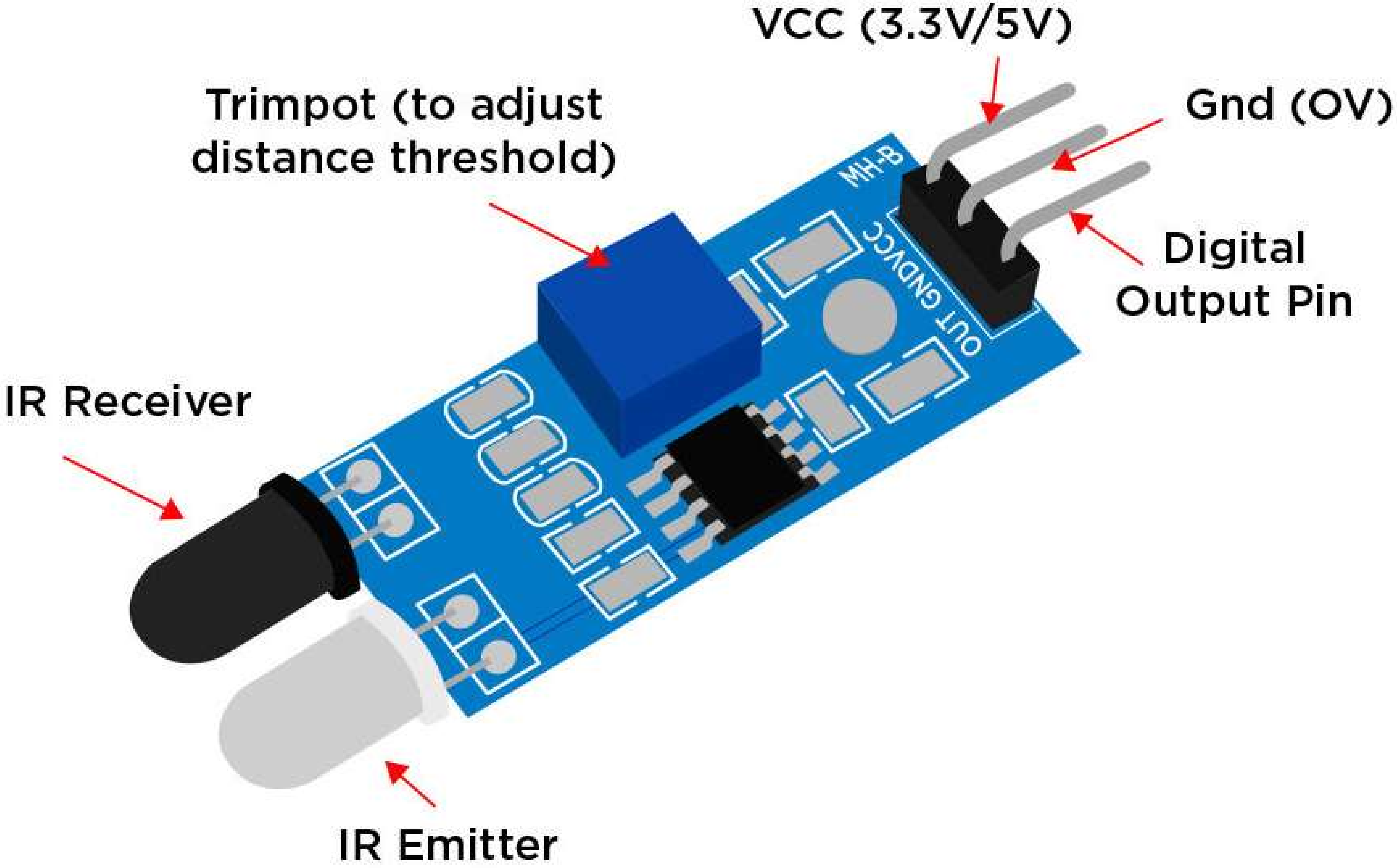


Infrared (IR) Sensor

- The Infrared Obstacle Sensor has IR transmitter that sends out IR signal and the IR receiver looks for reflected IR signal to detect the presence of any obstacle in front of the sensor.



Infrared (IR) Sensor



VCC (3.3V/5V)

Gnd (OV)

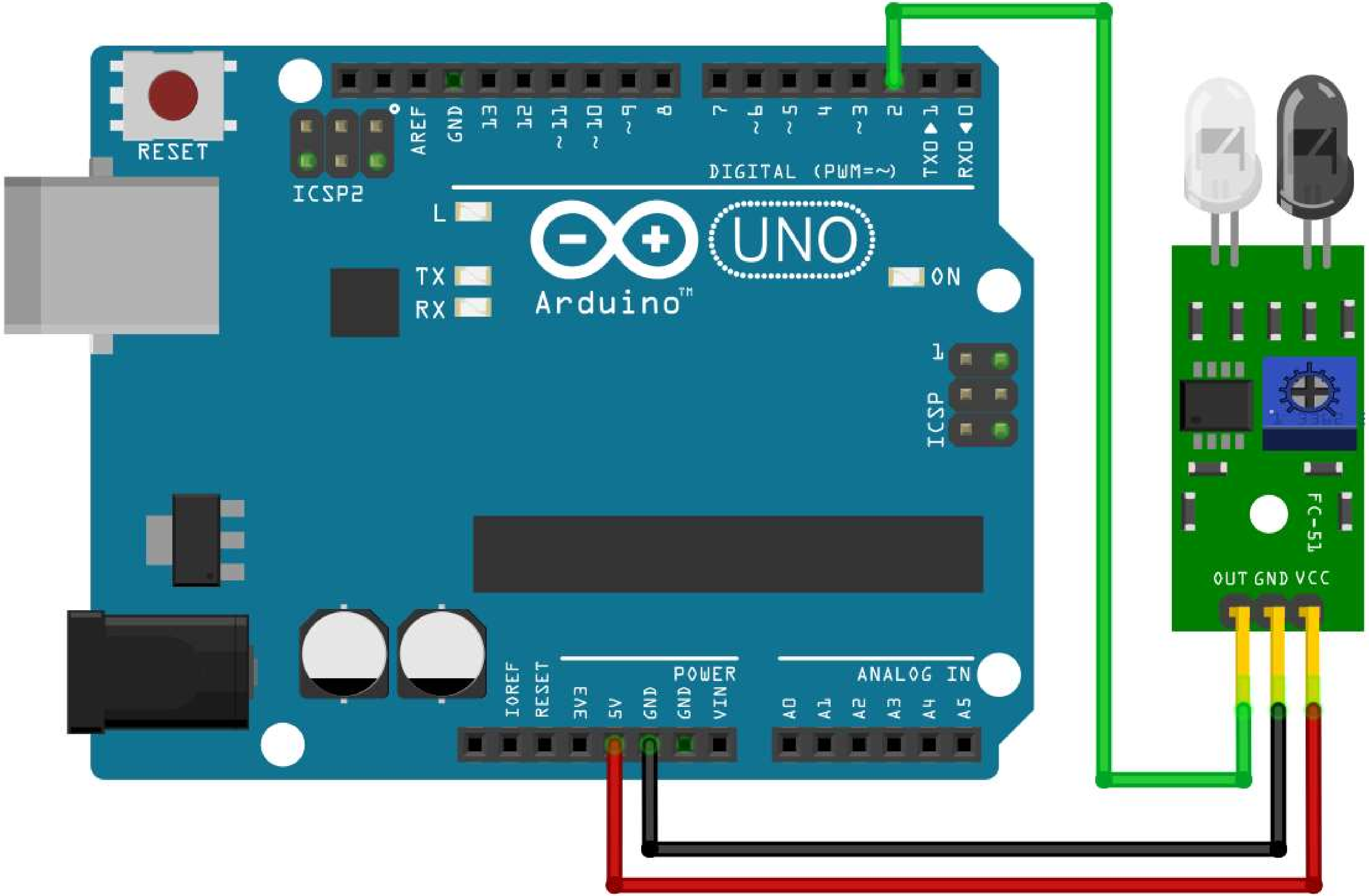
Trimpot (to adjust distance threshold)

Digital Output Pin

IR Receiver

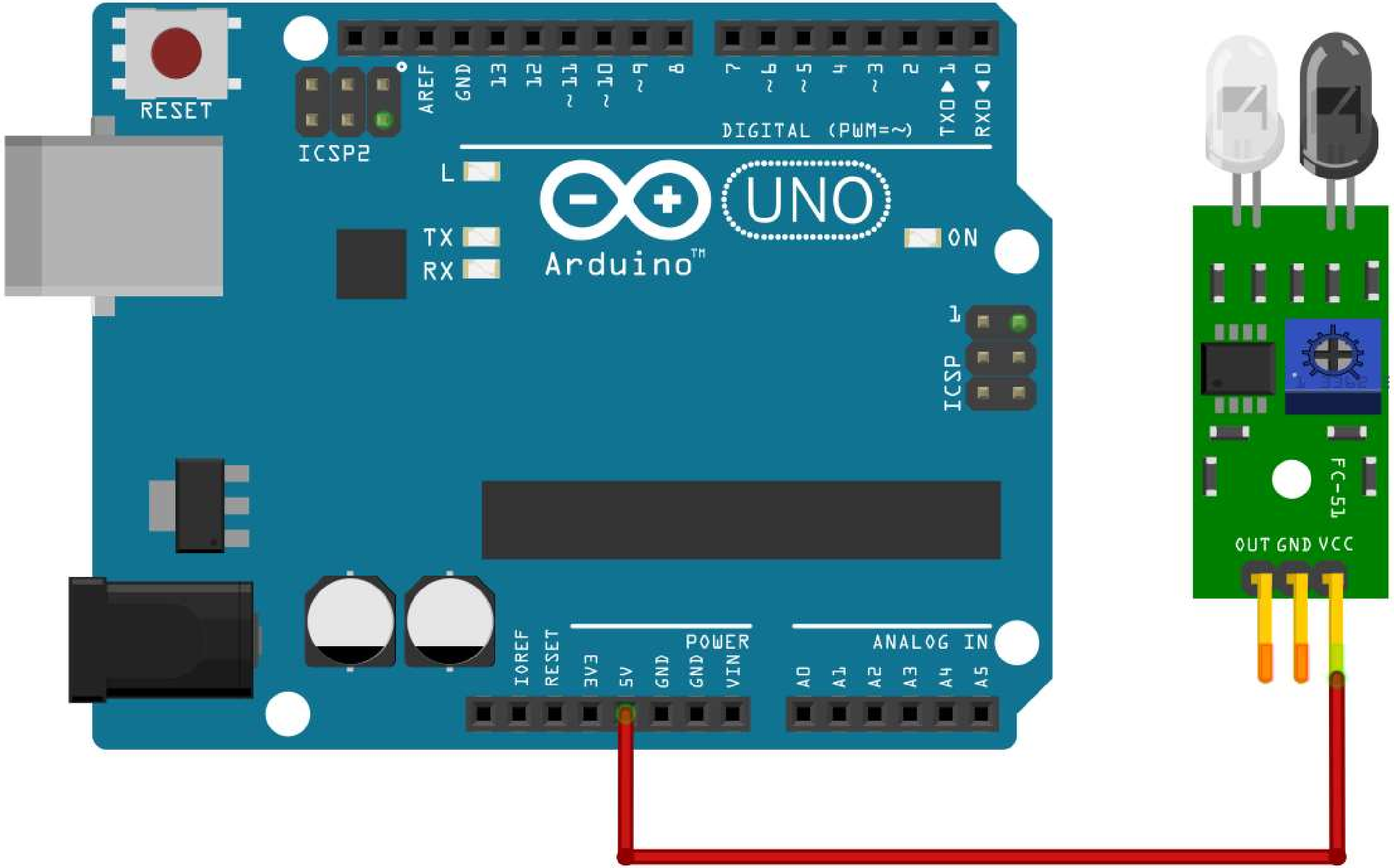
IR Emitter

Infrared (IR) Sensor: Circuit



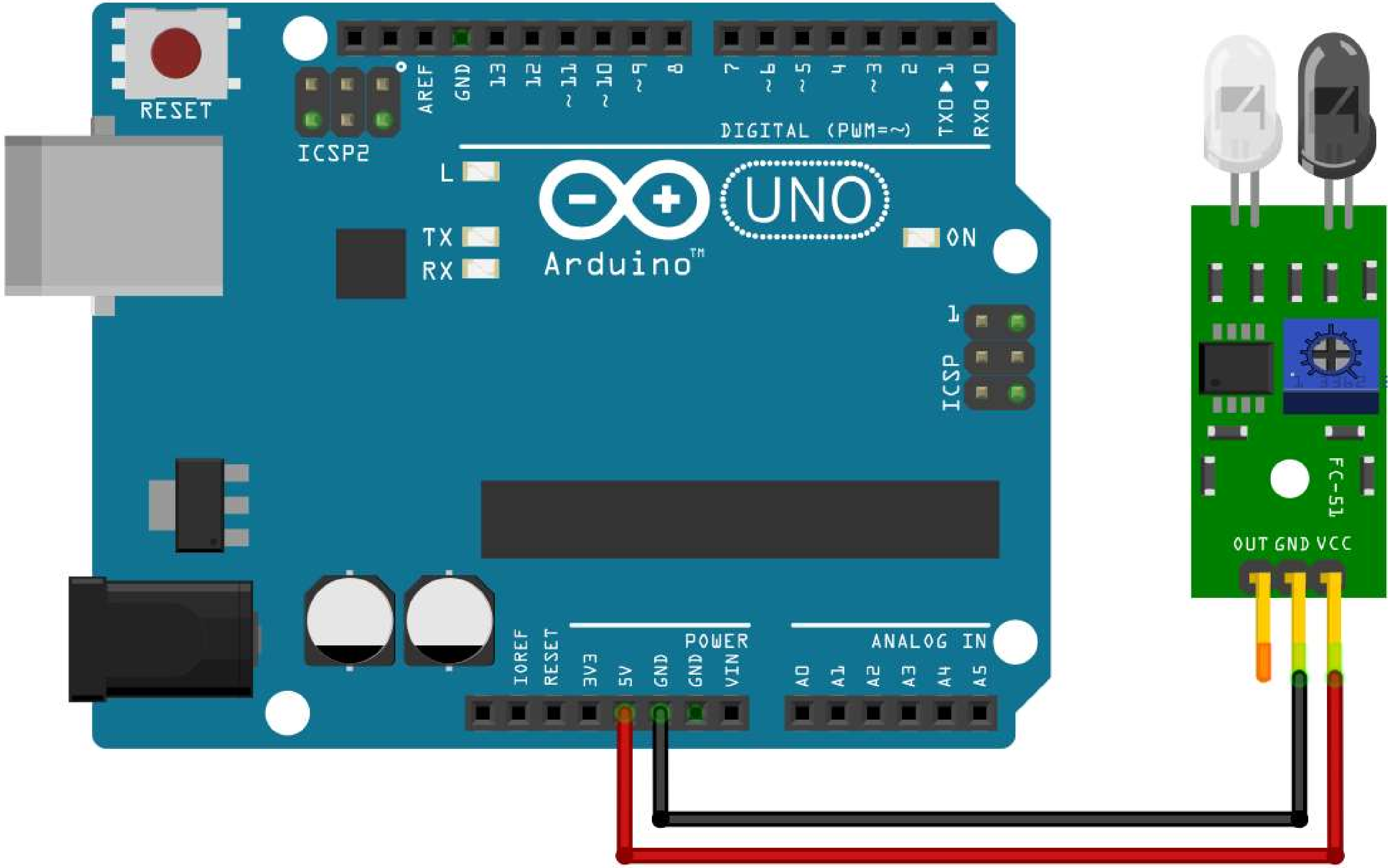
Infrared (IR) Sensor: Steps

1. The VCC pin of the sensor connects to the 5V on Arduino.



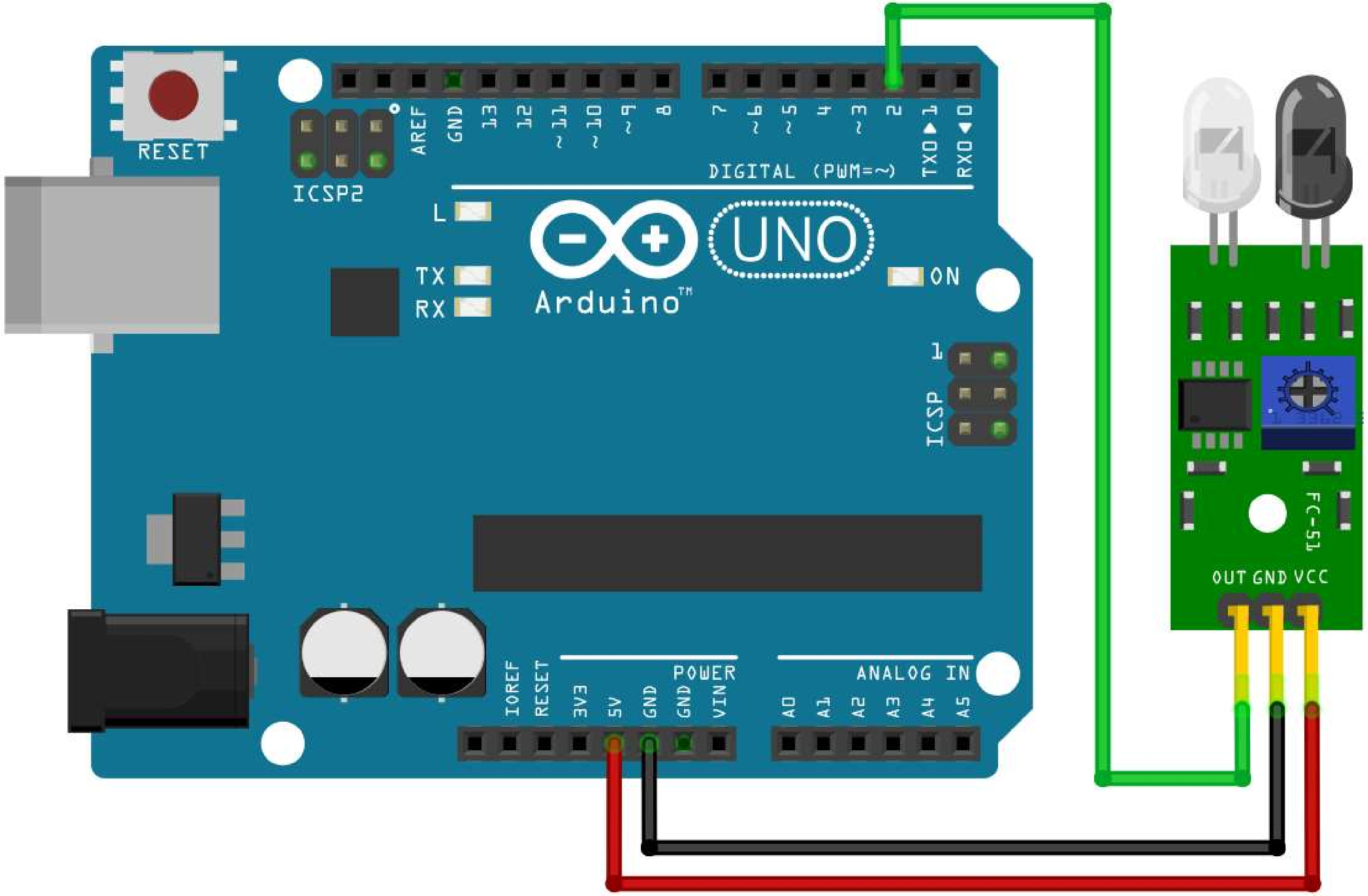
Infrared (IR) Sensor: Steps

2. The GND pin of the sensor connects to the ground on Arduino.



Infrared (IR) Sensor: Steps

3. The Digital OUT pin of the sensor connects to pin 2 on Arduino.

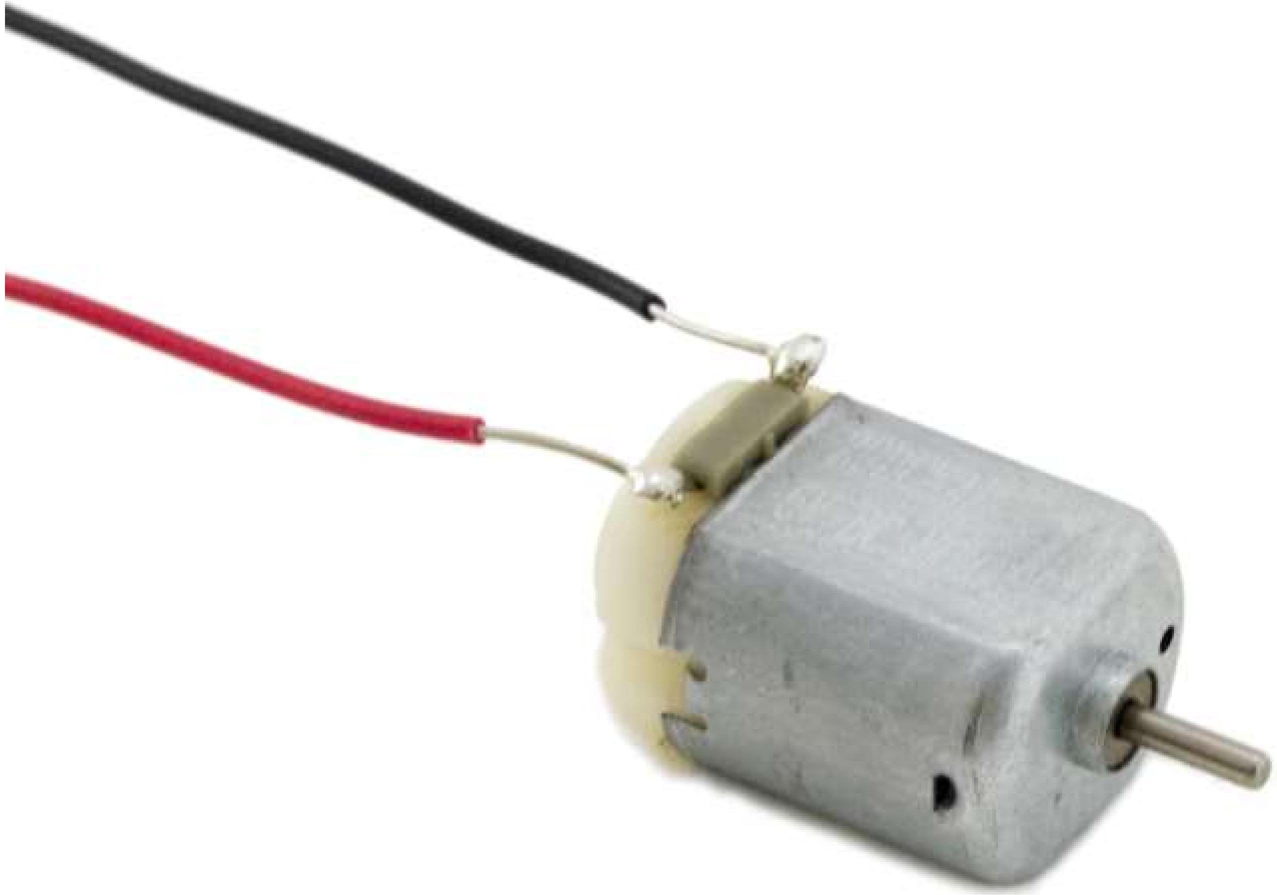


Infrared (IR) Sensor: Code

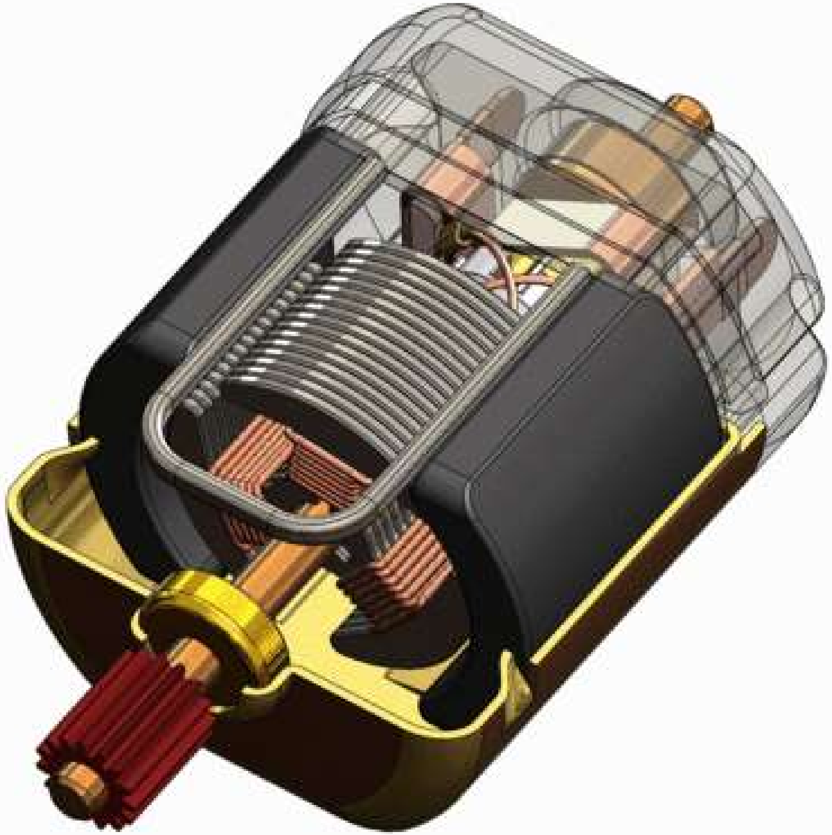
```
#define IR_PIN 2                            // IR Sensor pin
#define LED_PIN 13                          // LED pin

bool state;                                 // Variable to hold the sensor value

void setup() {
  pinMode(IR_PIN, INPUT);                   // Set IR_PIN as input
  pinMode(LED_PIN, OUTPUT);                 // Set LED_PIN as output
}

void loop() {
  state = digitalRead(IR_PIN);              // Read the value of IR sensor

  if(state == LOW)                          // If an object is detected,
    digitalWrite(LED_PIN, HIGH);            // turn on the LED
  else                                      // If no object is detected,
    digitalWrite(LED_PIN, LOW);             // turn off the LED
}
```

DC Motor

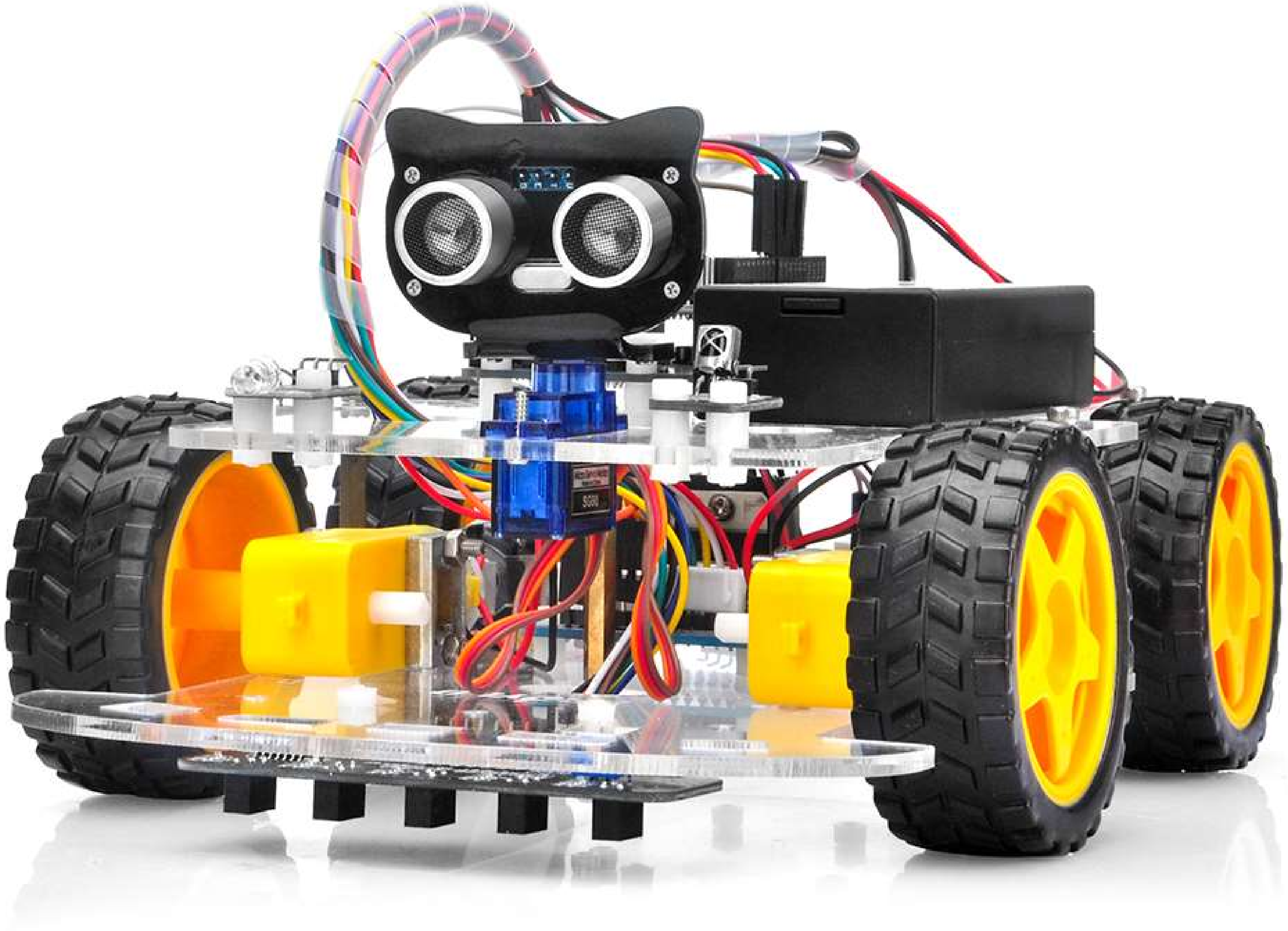
- A Direct Current (DC) is the most common type of motor.
- DC motors normally have just two leads, one positive and one negative.



DC Motor



DC Motor: Application

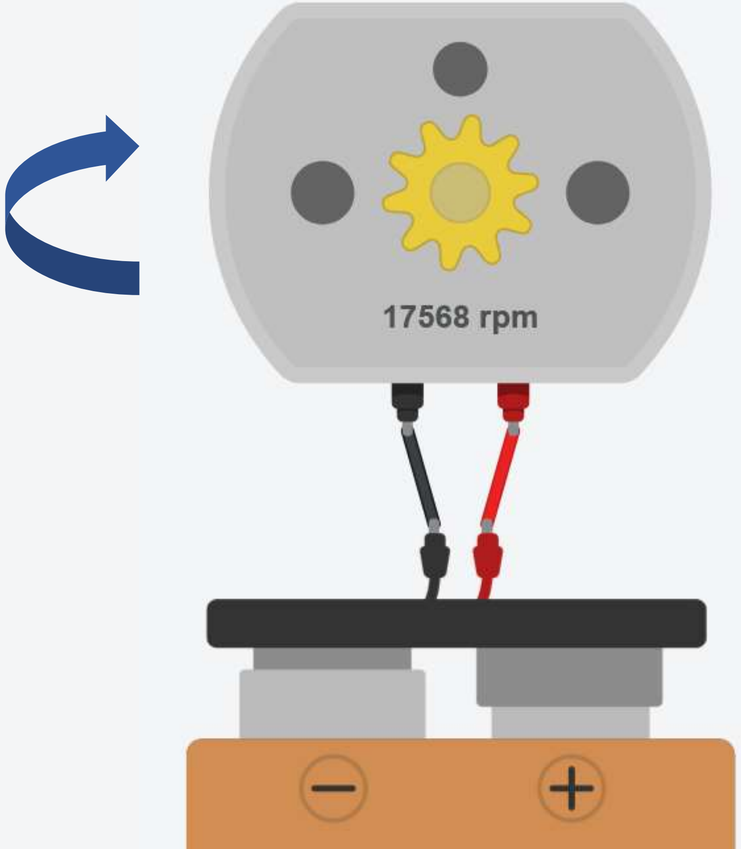


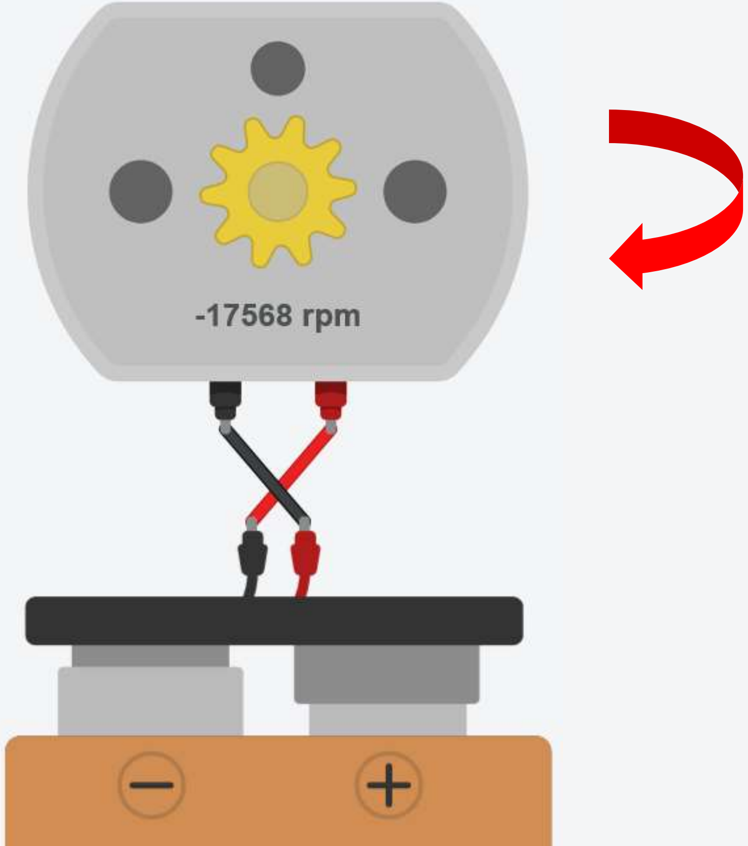
DC Motor: Application

DC Motor: Application

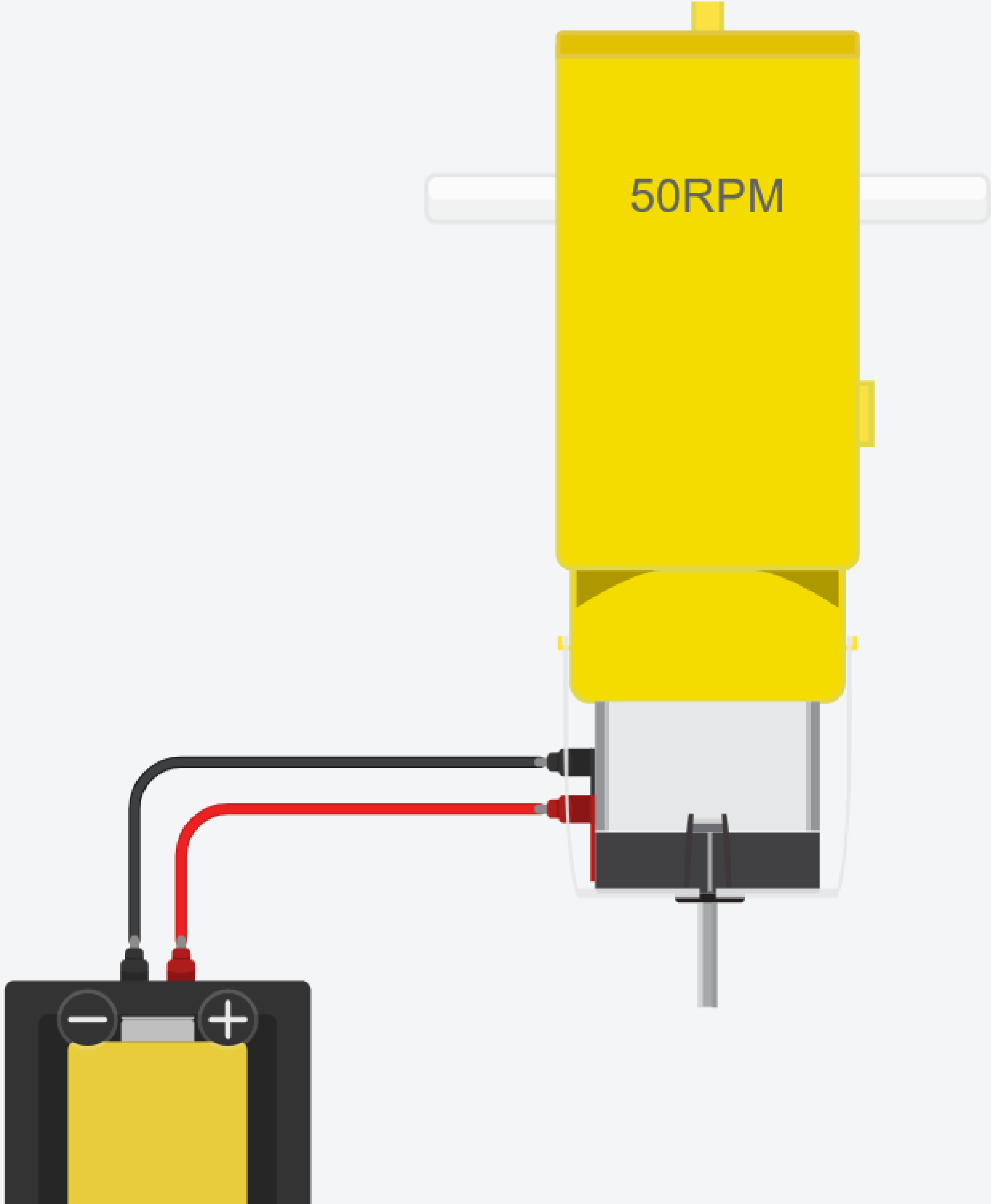
DC Motor: Rotation Direction

- DC motors normally have just two leads, one positive and one negative.
- If you connect these two leads directly to a battery, the motor will rotate.
- If you switch the leads, the motor will rotate in the opposite direction.

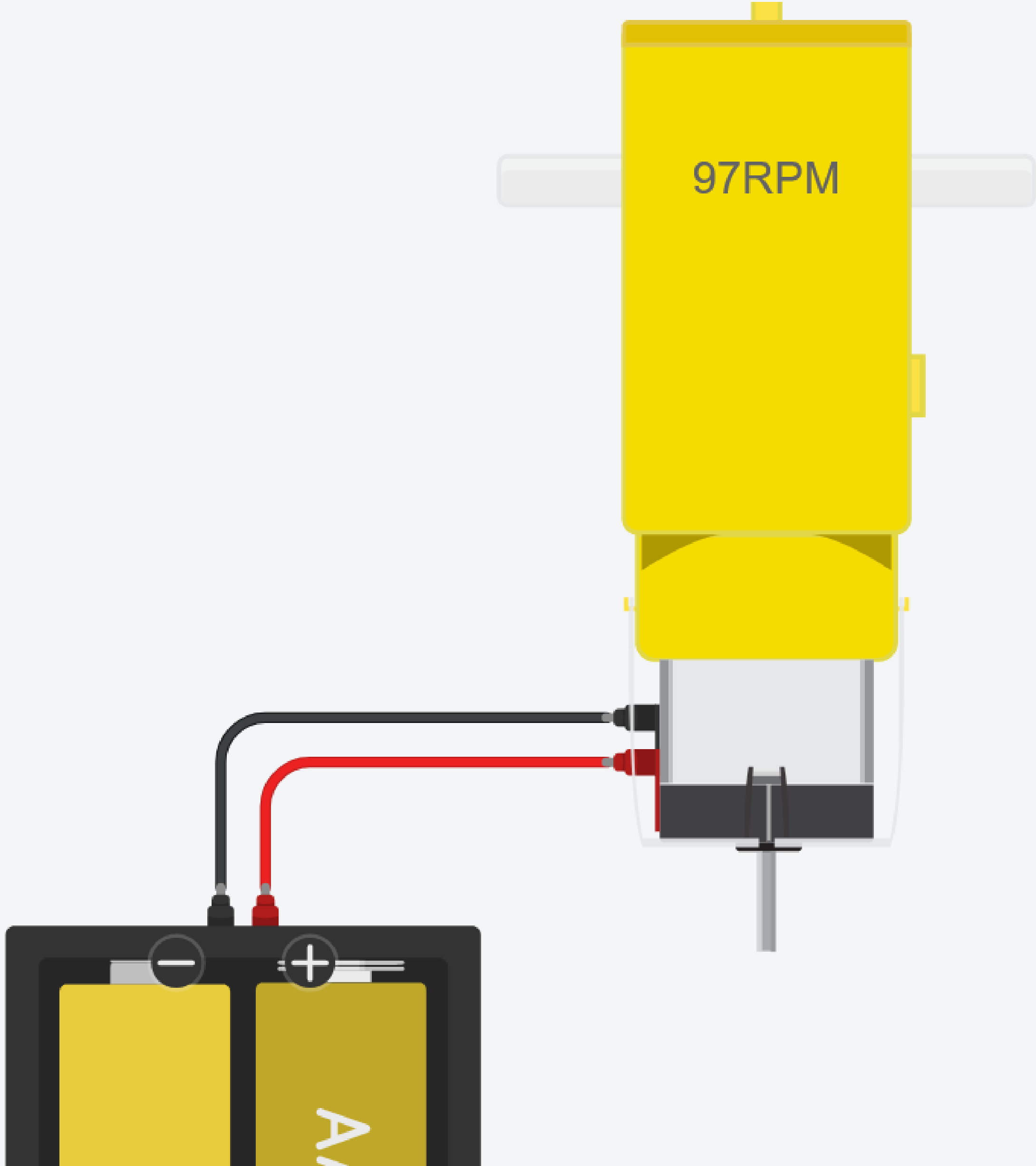




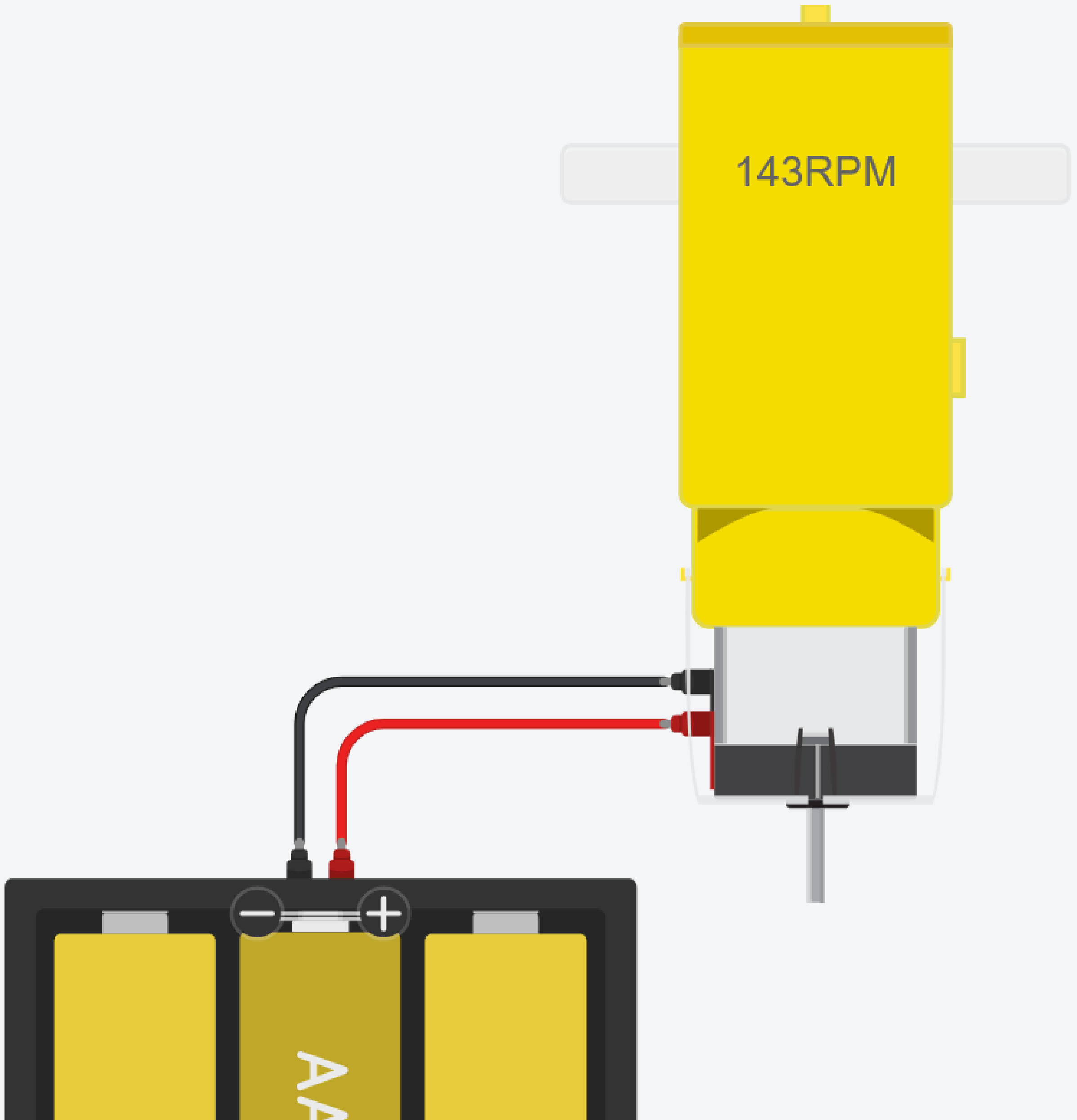
DC Motor: Speed



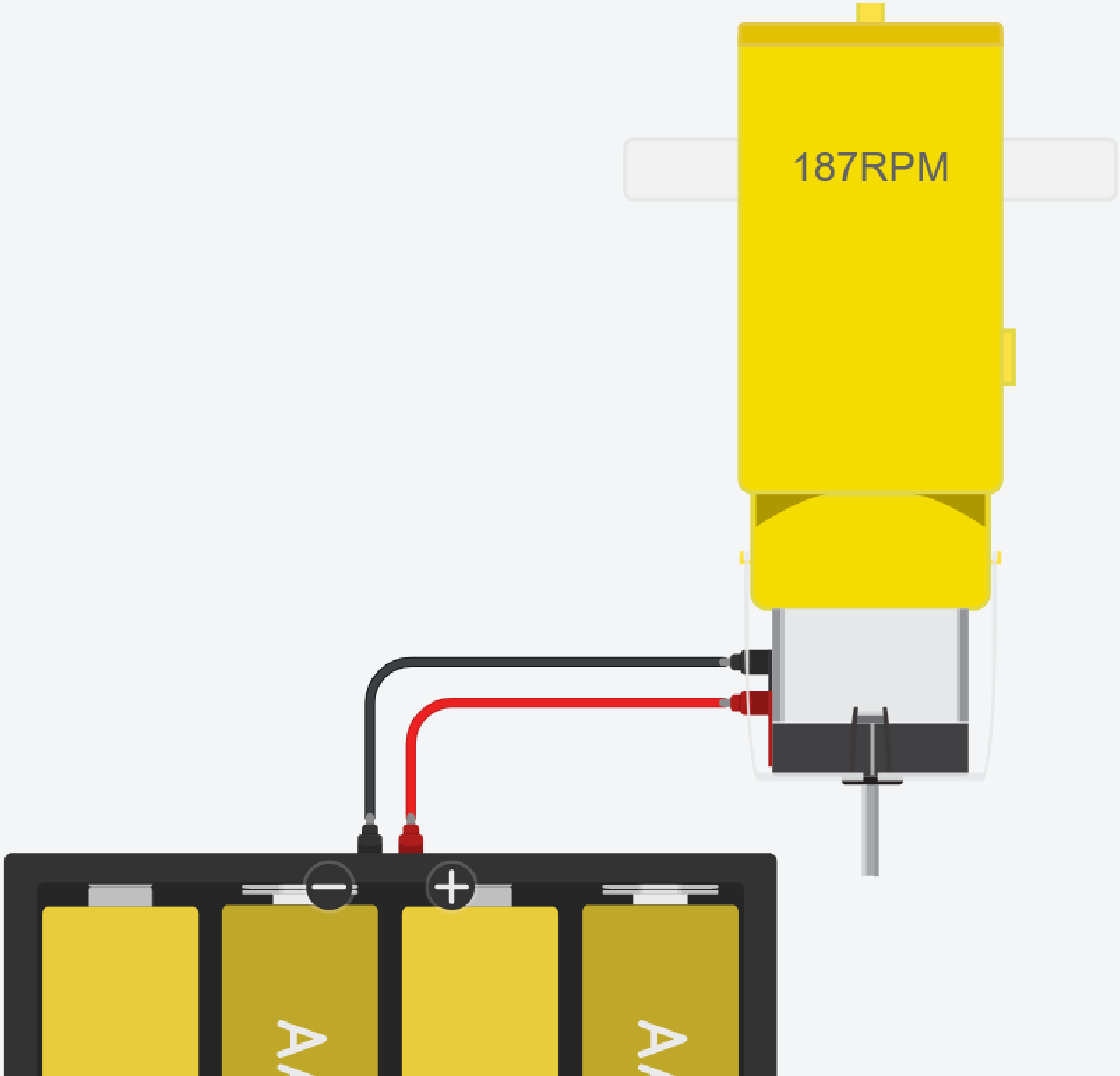
DC Motor: Speed



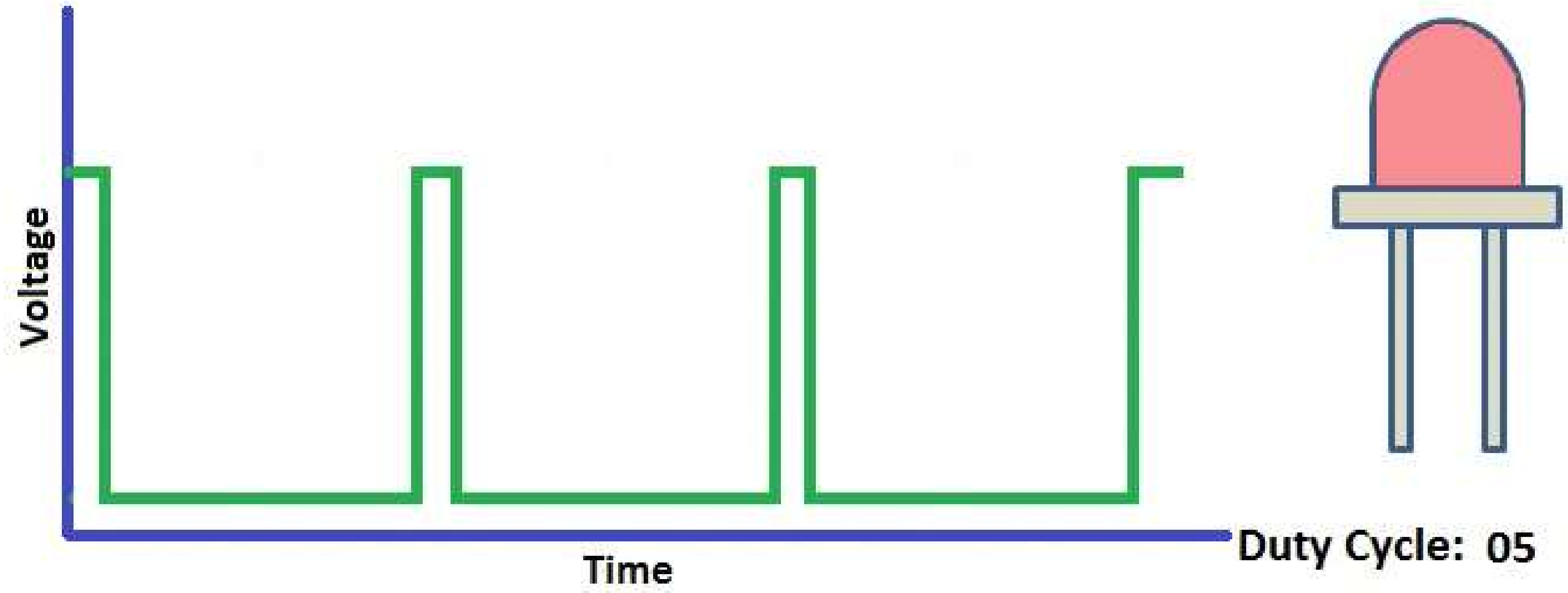
DC Motor: Speed



DC Motor: Speed



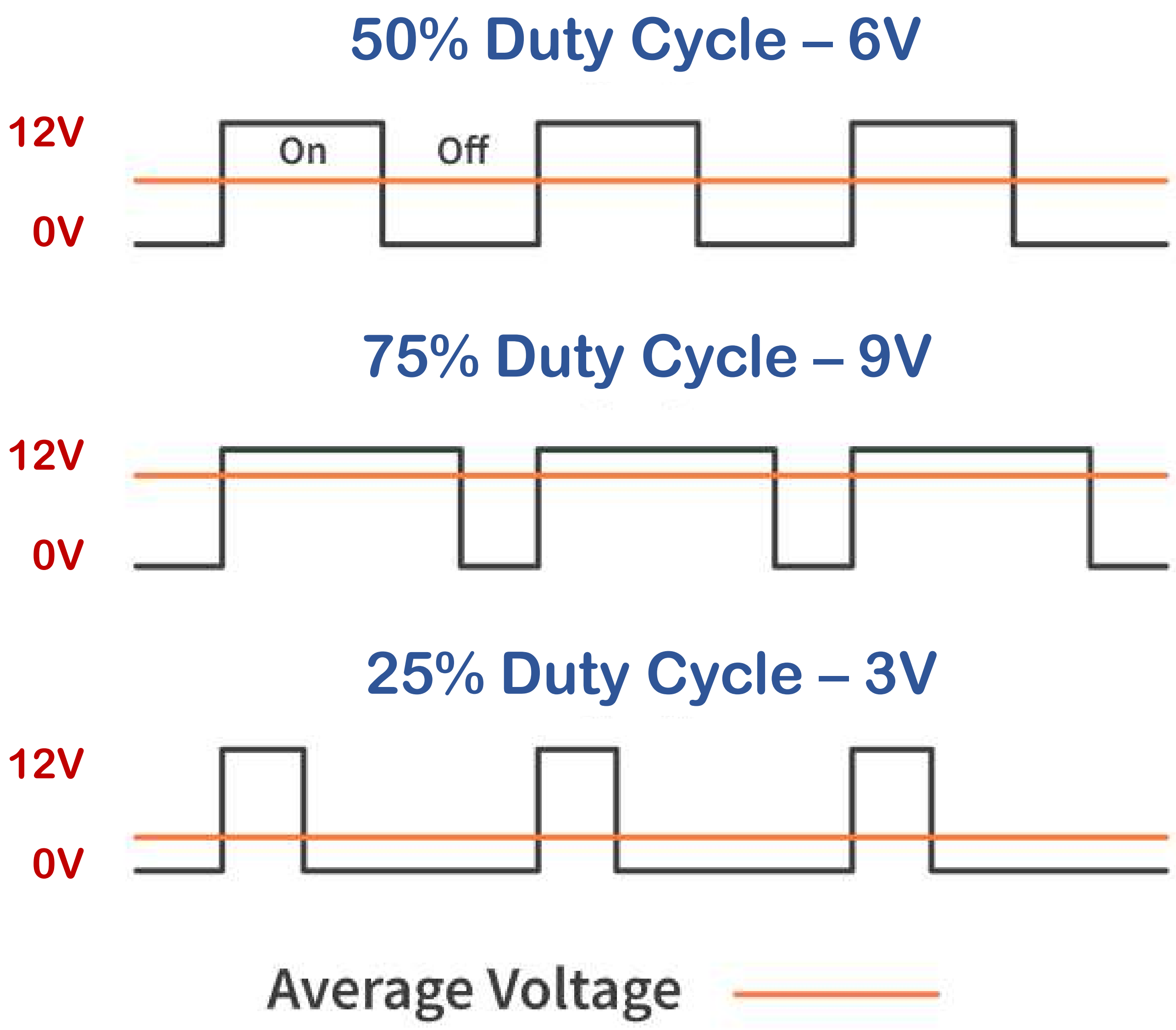
DC Motor: Speed – PWM



DC Motor: Speed – PWM

- We can control the speed of the DC motor by simply changing the input voltage to the motor by using PWM signal.
- The Pulse Width Modulation (PWM) is a technique which allows us to adjust the average value of the voltage.
- The average voltage depends on the duty cycle, or the amount of time the signal is HIGH versus the amount of time the signal is LOW in a single period of time.
- This average voltage is proportional to the width of the pulses, which is referred to as the Duty Cycle.

DC Motor: Speed – PWM



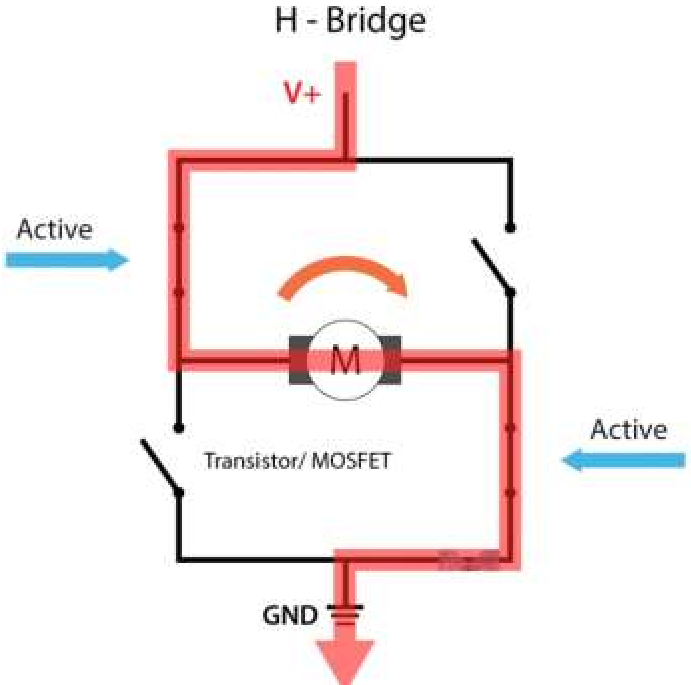
50% Duty Cycle – 6V

75% Duty Cycle – 9V

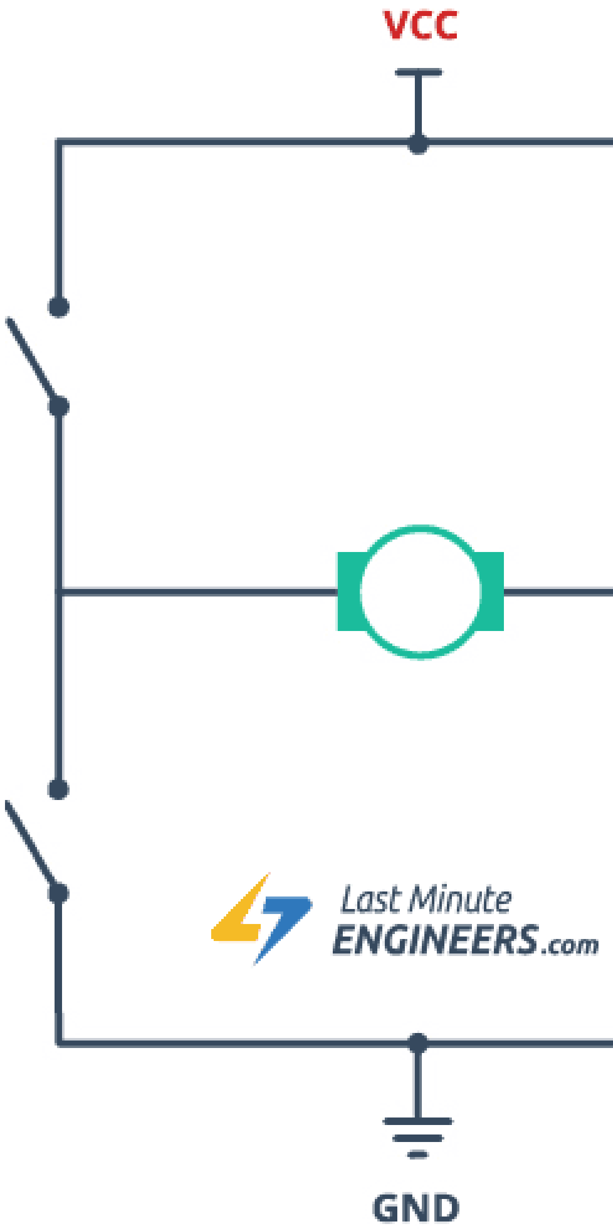
25% Duty Cycle – 3V

H-Bridge

- The spinning direction of a DC motor can be controlled by changing the polarity of its input voltage.
- A widely used technique to accomplish this is to use an H-bridge.
- An H-bridge circuit is made up of four switches arranged in a H shape, with the motor in the center.

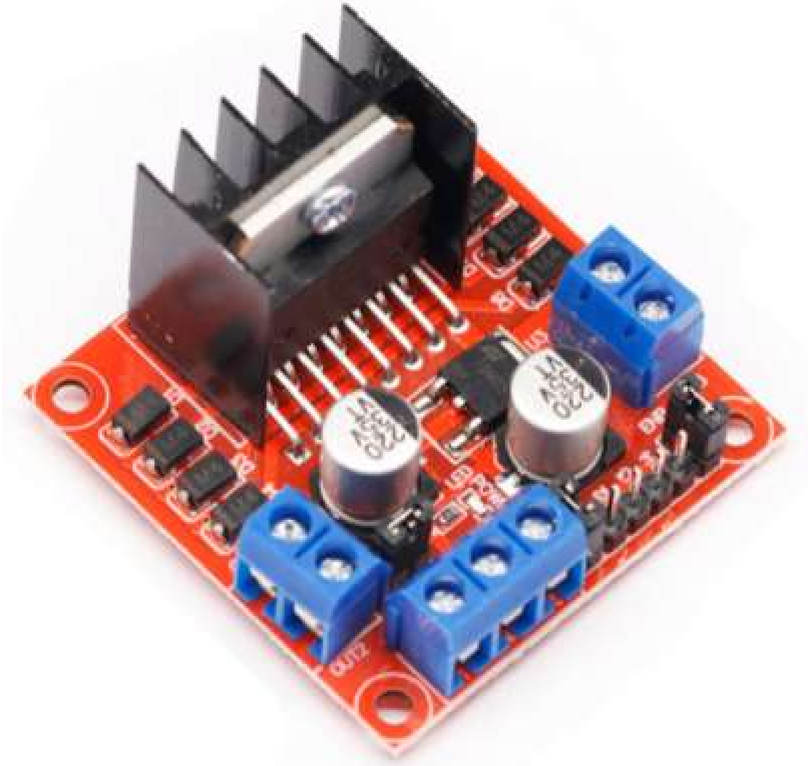


H-Bridge



L298 Motor Driver

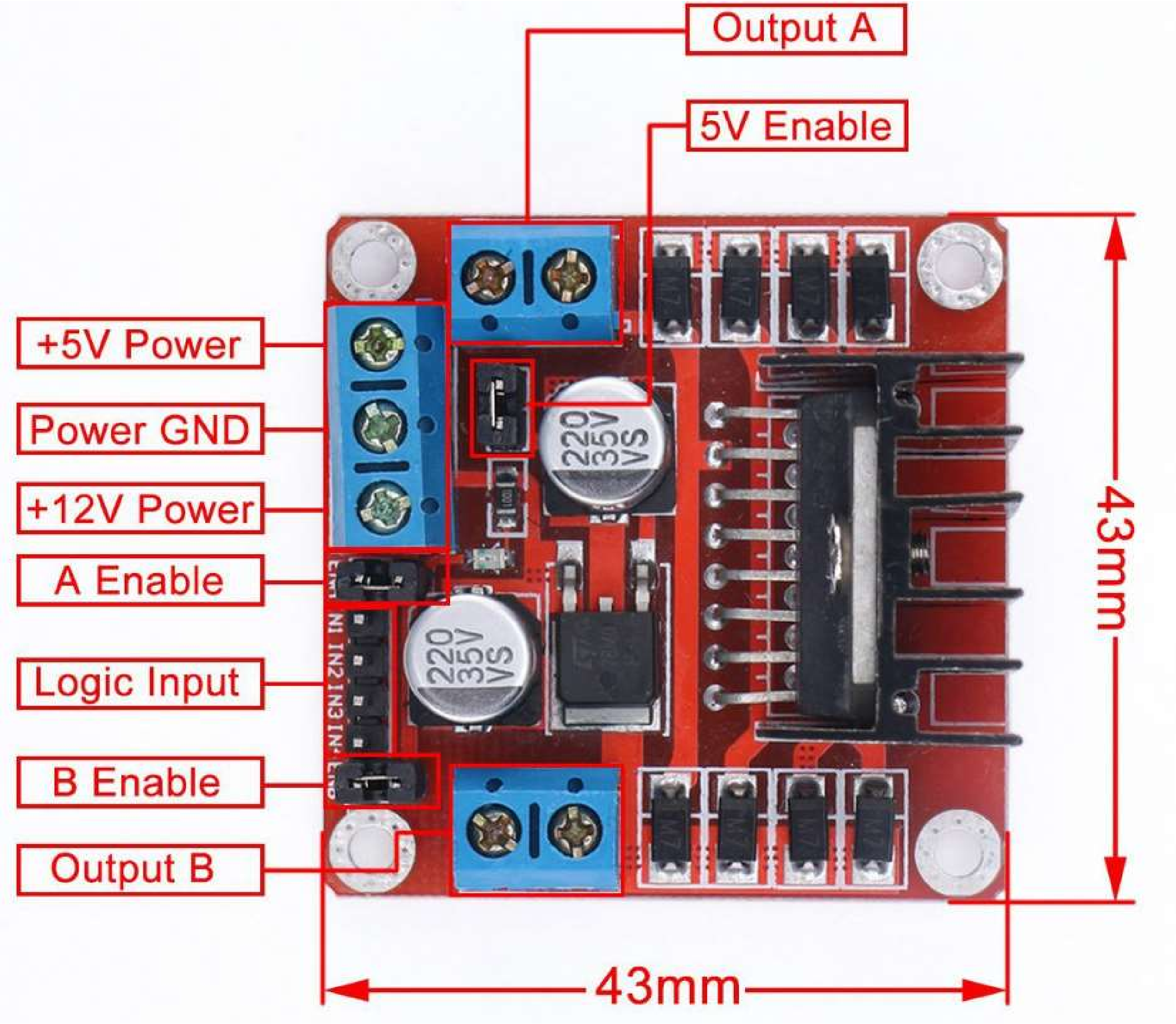
- The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time.
- The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.



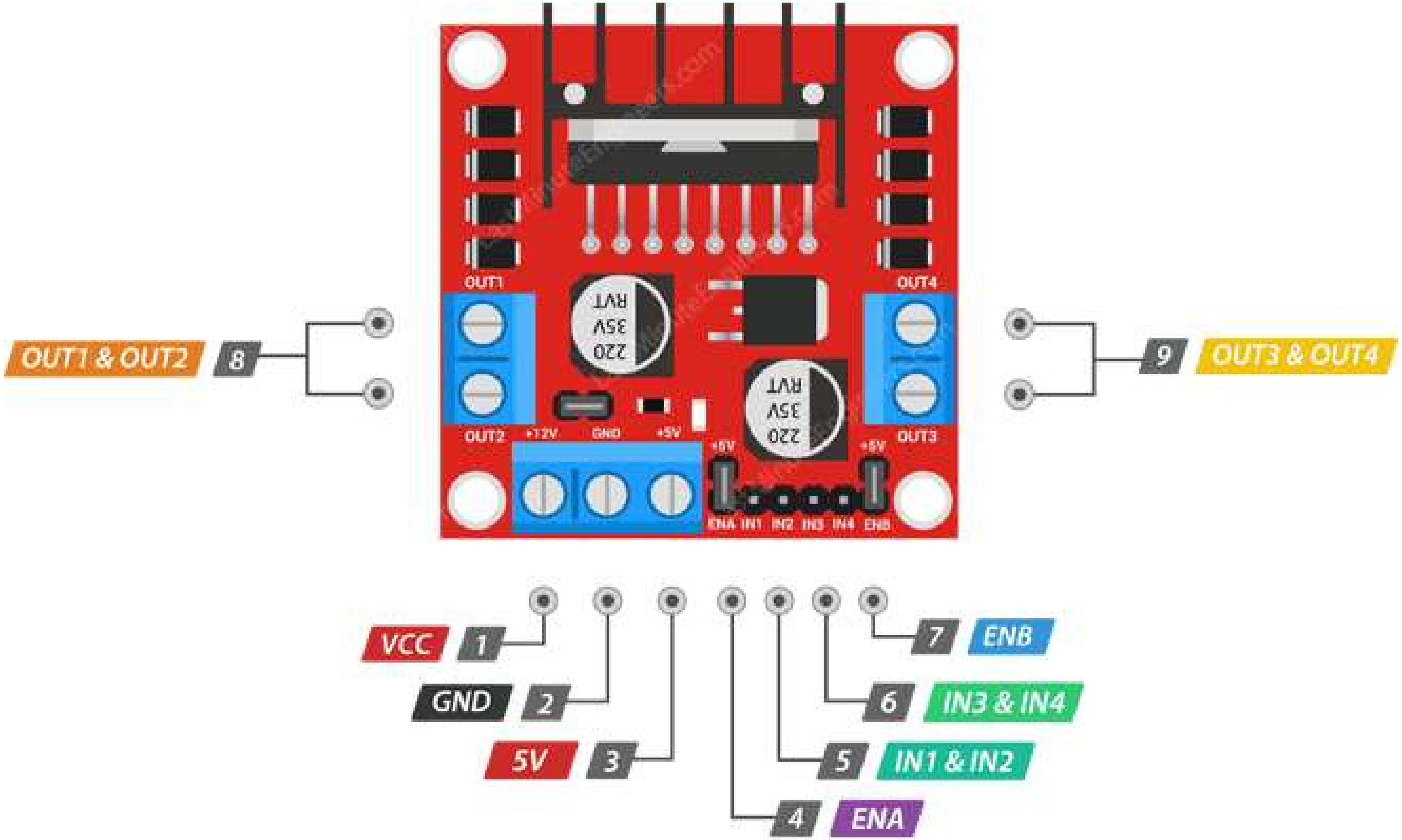
L298 Motor Driver: Specifications

Feature	Description
Driver Model	L298N Dual H-Bridge
Drive Voltage	5V-35V
Drive Current	2A (MAX)
Logical Voltage	5V
Logical Current	0-36mA

L298 Motor Driver: Pinout



L298 Motor Driver: Pinout

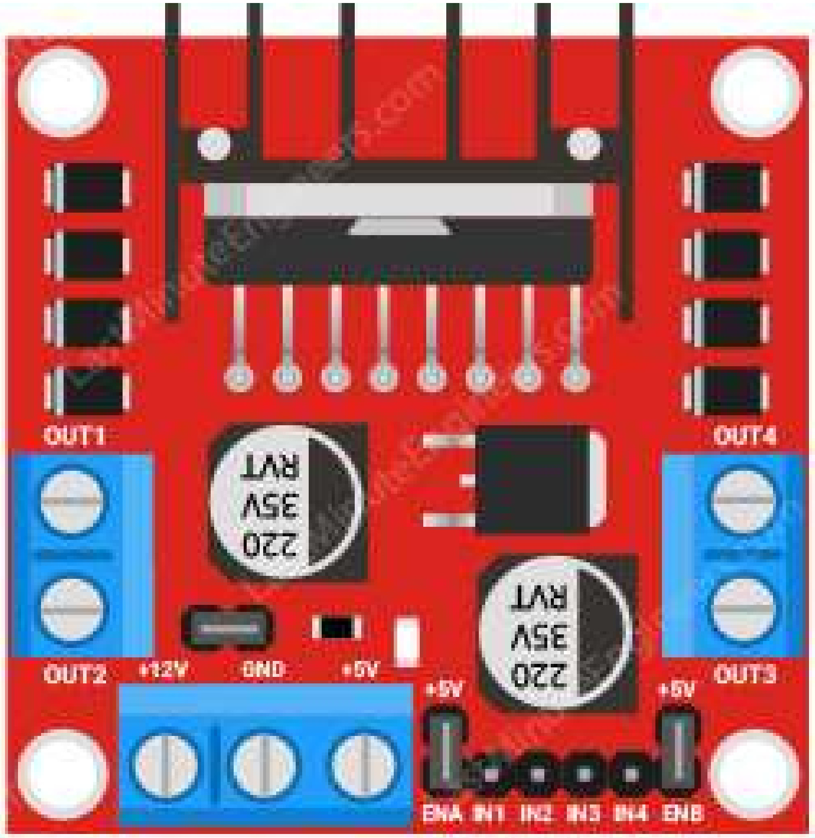


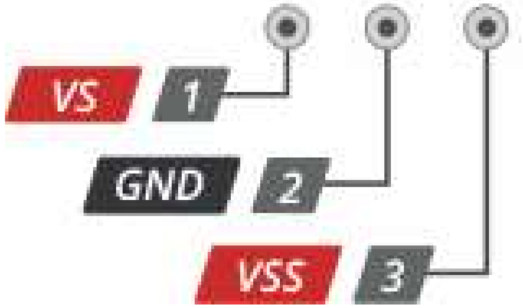
L298 Motor Driver: Power Pins

- The VS pin powers the internal H-Bridge, which drives the motors. This pin accepts input voltages ranging from 5 to 12V.

- The VSS pin is used to power the logic circuitry within the L298N IC, and can range between 5V and 7V.

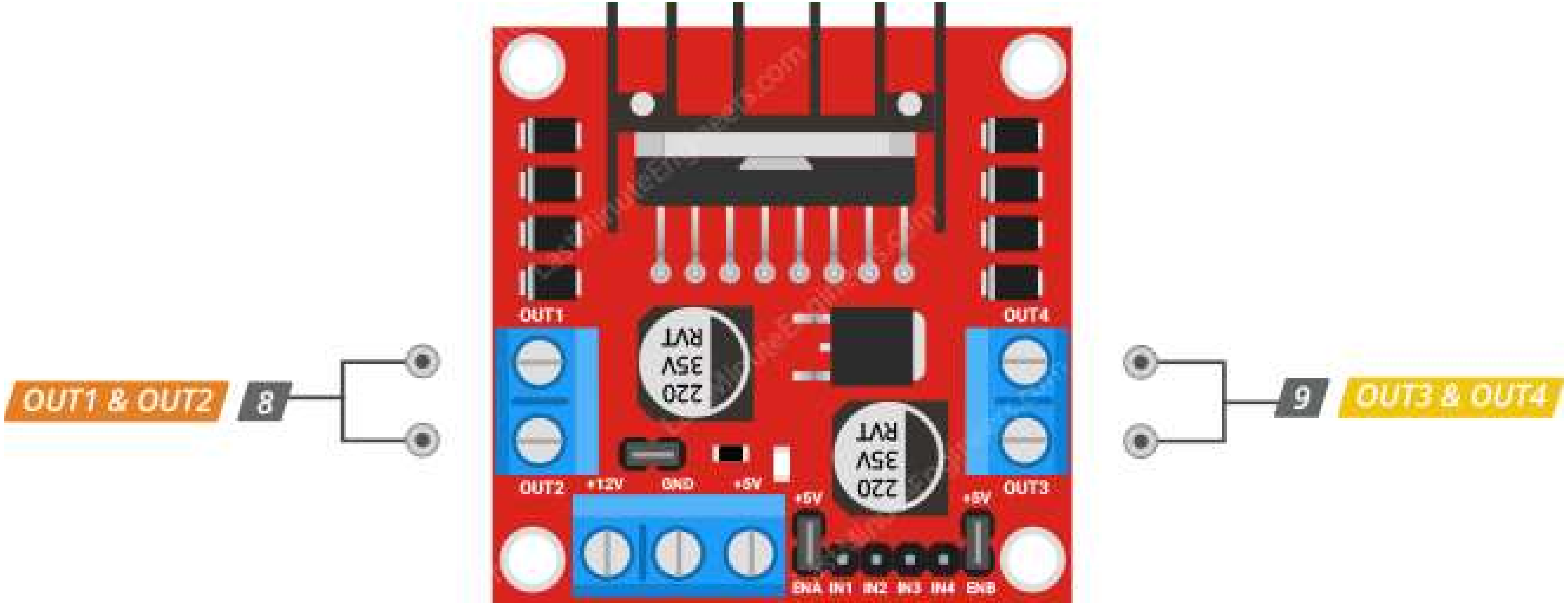
- The GND pin is the common ground pin.





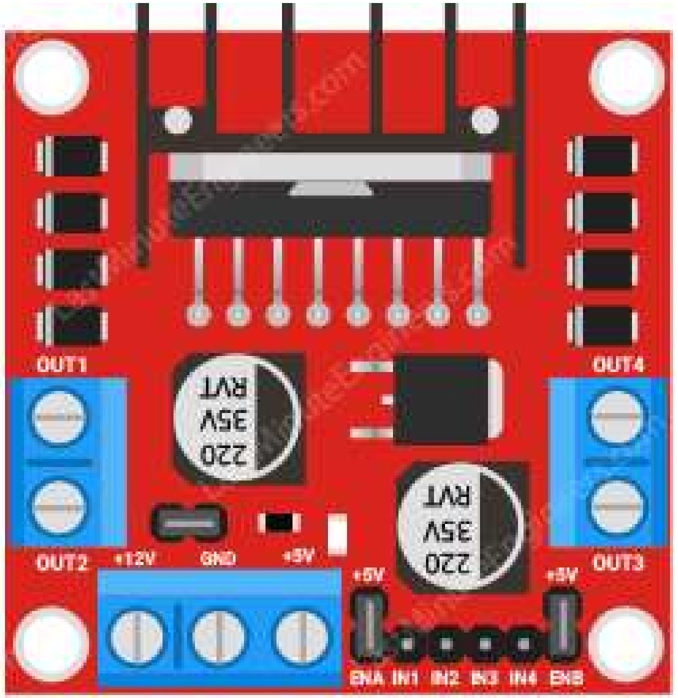
L298 Motor Driver: Output Pins

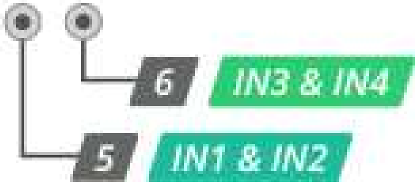
- The output channels of the L298N motor driver, OUT1 and OUT2 for motor A and OUT3 and OUT4 for motor B.
- You can connect two 5-12V DC motors to these terminals.



L298 Motor Driver: Direction Control Pins

- The IN1 and IN2 pins control the spinning direction of motor A.
- The IN3 and IN4 control the spinning direction of motor B.





6 IN3 & IN4

5 IN1 & IN2

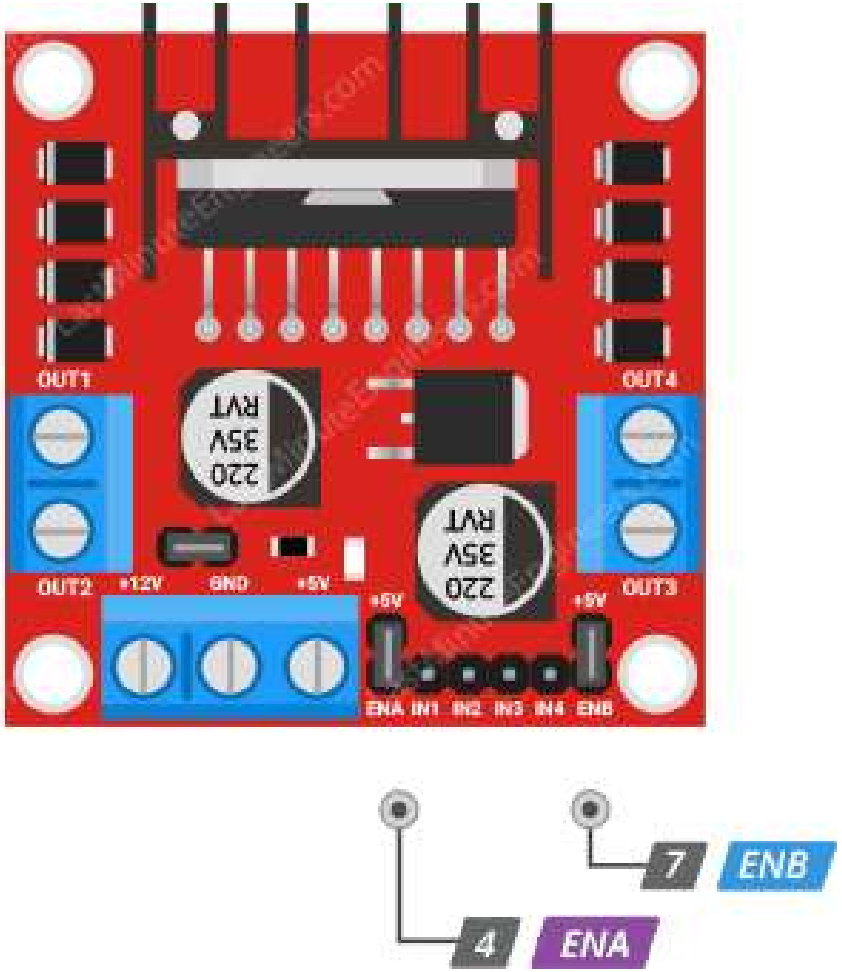
L298 Motor Driver: Direction Control Pins

- The spinning direction of the motor can be controlled by applying logic HIGH or logic LOW to these inputs.

Input1	Input2	Spinning Direction
Low (0)	Low (0)	Motor OFF
High (1)	Low (0)	Forward
Low (0)	High (1)	Backward
High (1)	High (1)	Motor OFF

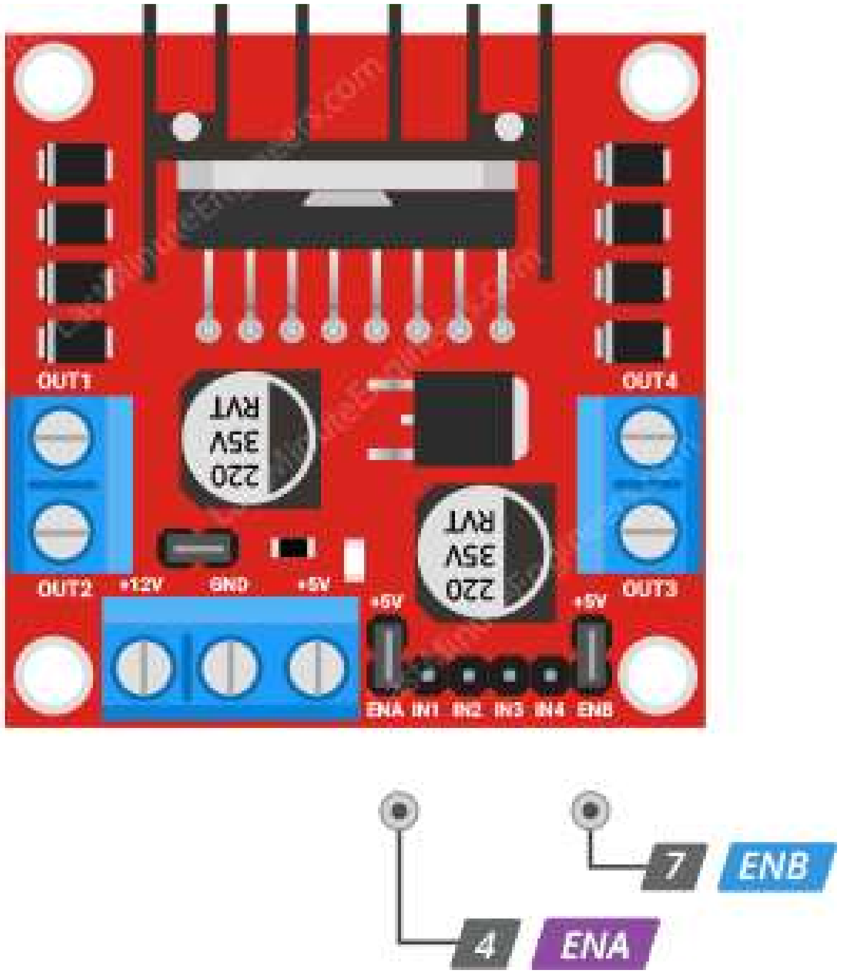
L298 Motor Driver: Speed Control Pins

- The speed control pins ENA and ENB are used to turn on/off the motors and control their speed.



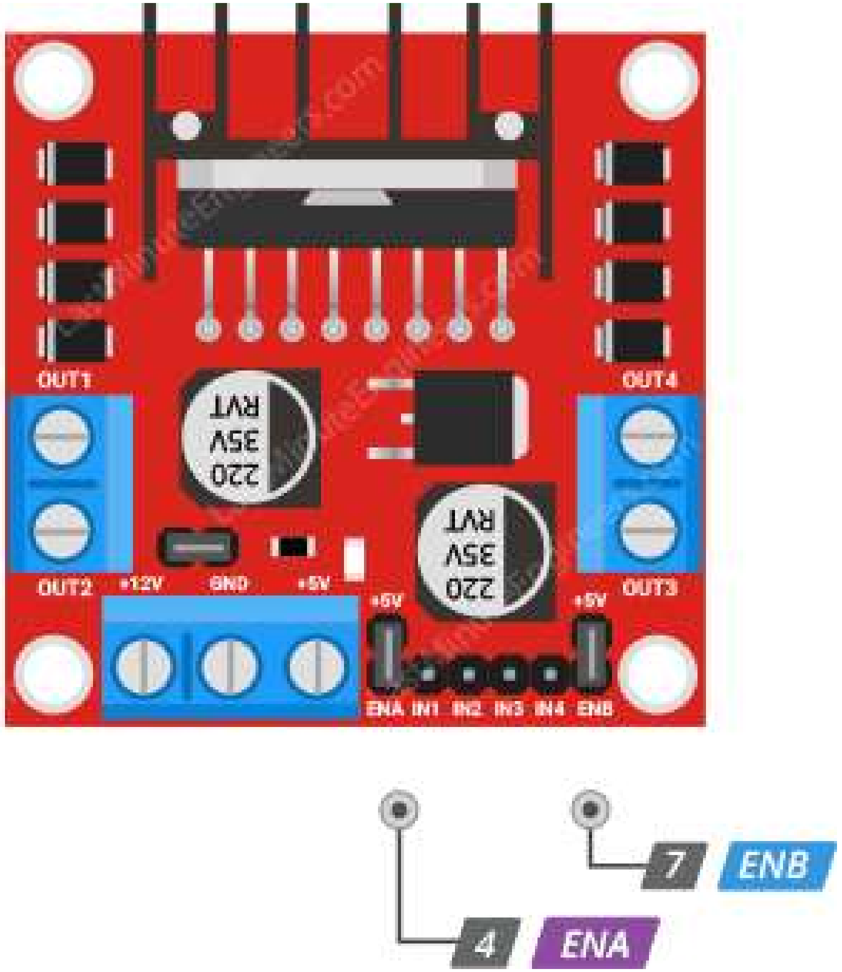
L298 Motor Driver: Speed Control Pins

- Pulling these pins HIGH will cause the motors to spin, while pulling them LOW will stop them.



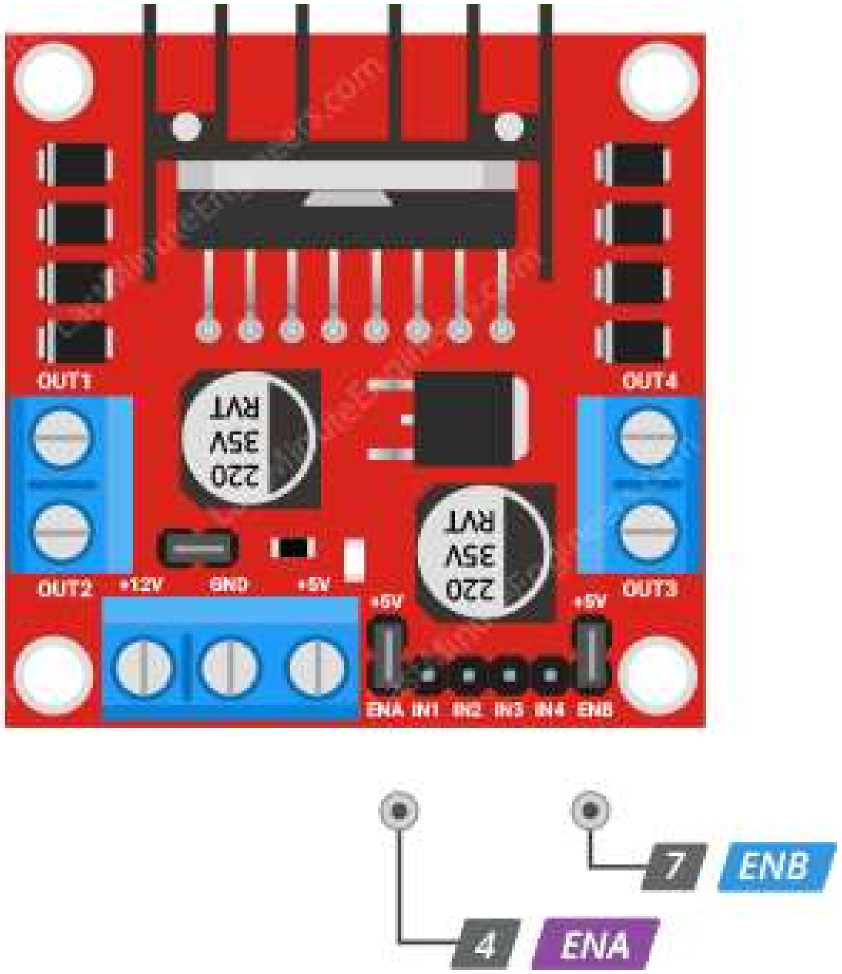
L298 Motor Driver: Speed Control Pins

- The module usually comes with a jumper on these pins.
- When this jumper is in place, the motor spins at full speed.



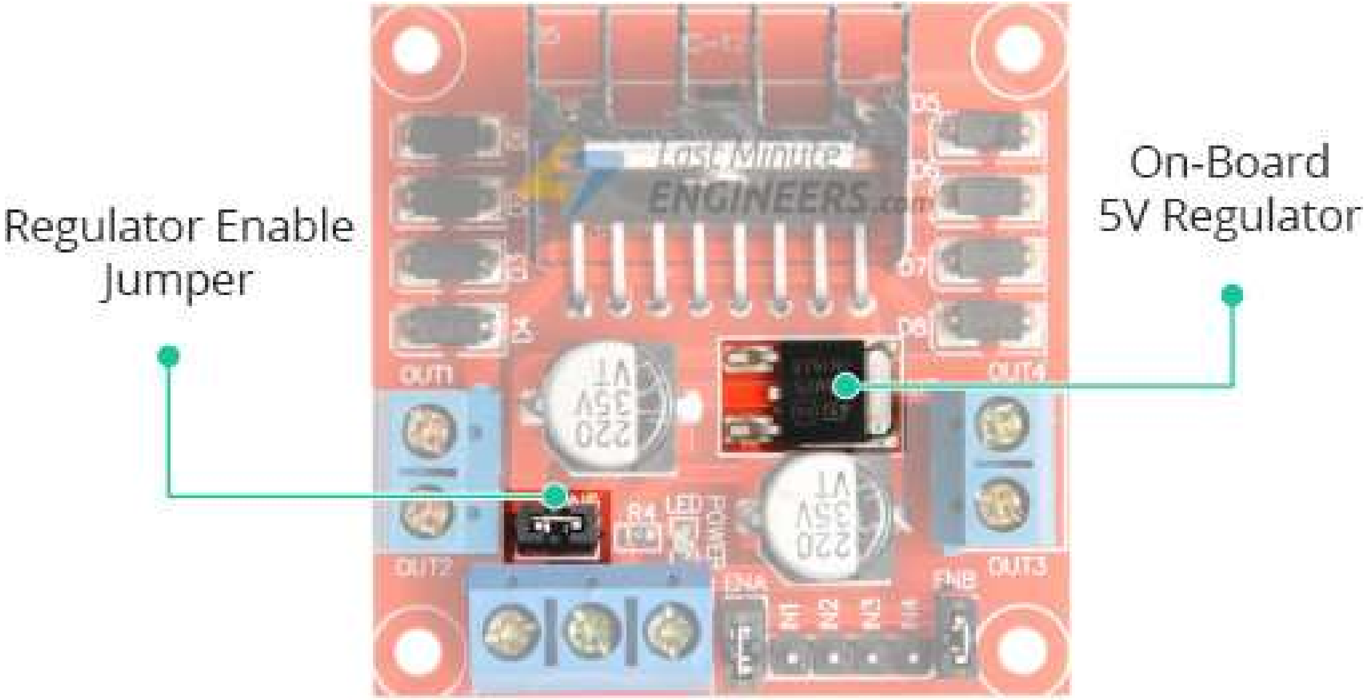
L298 Motor Driver: Speed Control Pins

- If you want to control the speed of the motors programmatically, remove the jumpers and connect them to the Arduino’s PWM-enabled pins.



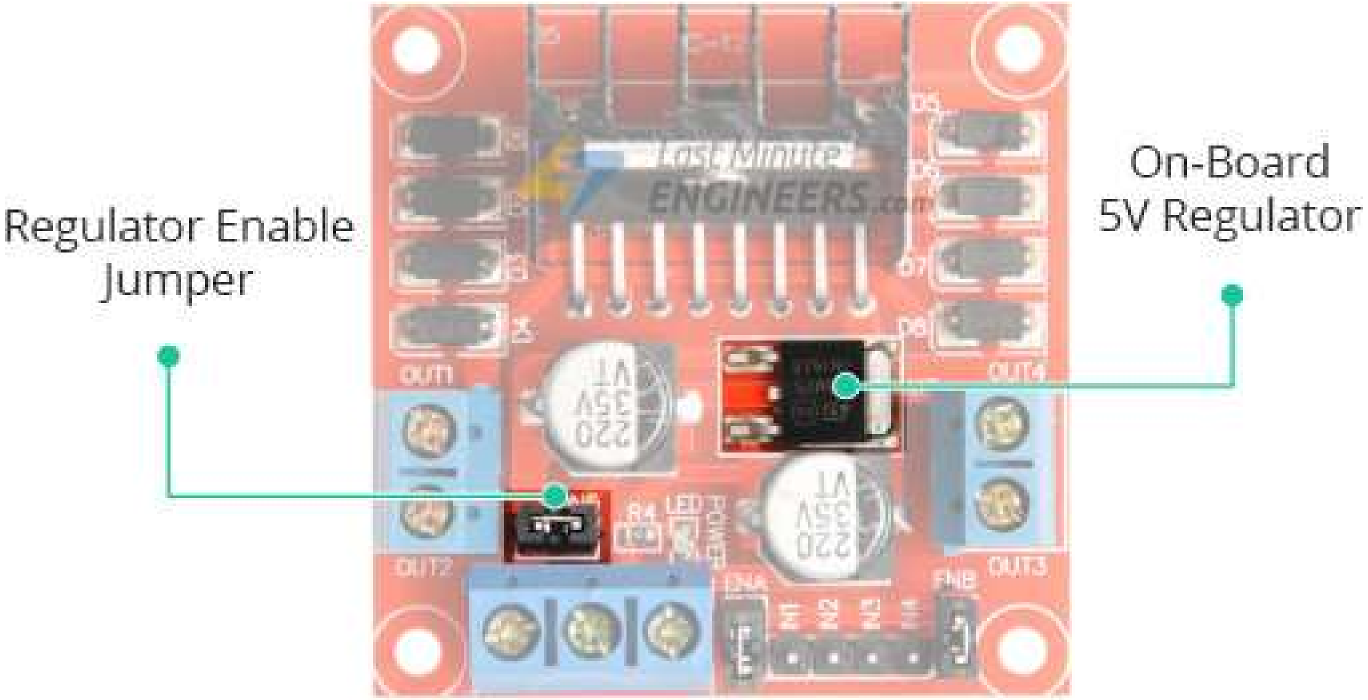
L298 Motor Driver: On-board 5V Regulator

- The module includes a 78M05 5V regulator that can be enabled or disabled via a jumper.
- When this jumper is in place, the 5V regulator is enabled.
- In this case, the 5V input acts as the output pin, delivering 5V 0.5A.



L298 Motor Driver: On-board 5V Regulator

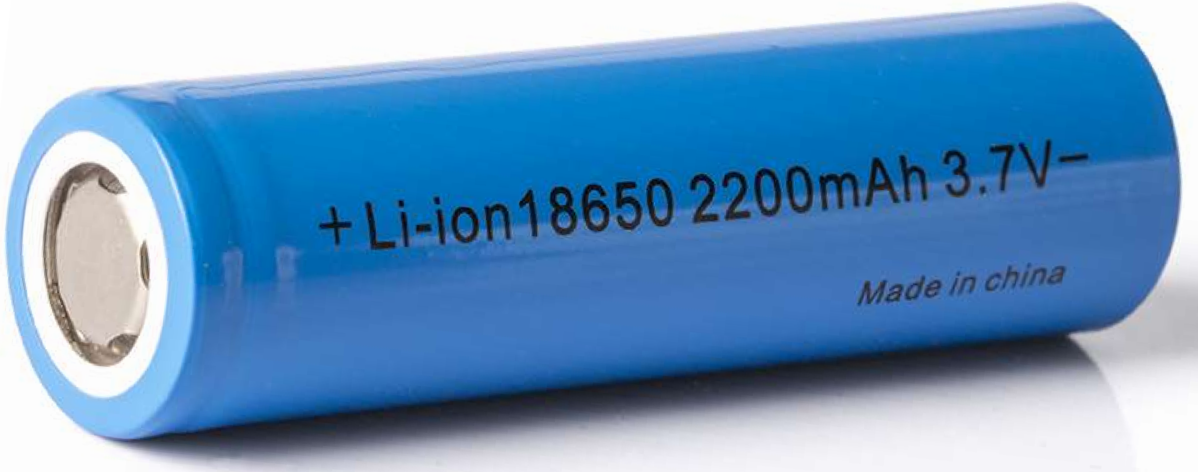
- You can use VSS pin to power an Arduino that needs 5V power.
- If the motor power supply is less than 12V, keep the jumper in place.
- If it is greater than 12V, the jumper must be removed to prevent damage to the onboard 5V regulator.



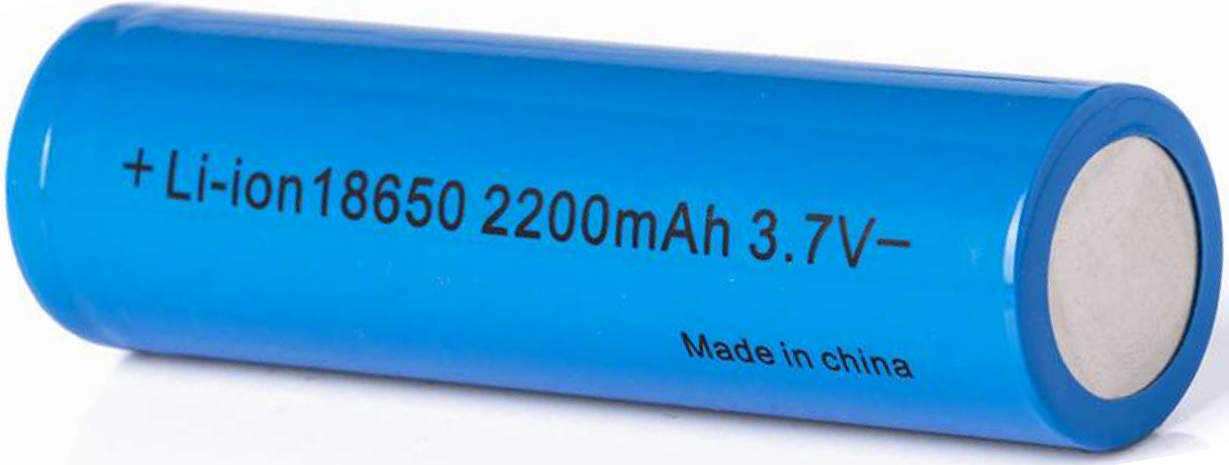
Lithium Li-ion 18650 Battery

- We will use three 3.7V 18650 batteries in series to power L298 module.

Lithium Li-ion 18650 Battery



+ Terminal



- Terminal

Remote Car: IR Remote Control

⏻ 0x1FE7887

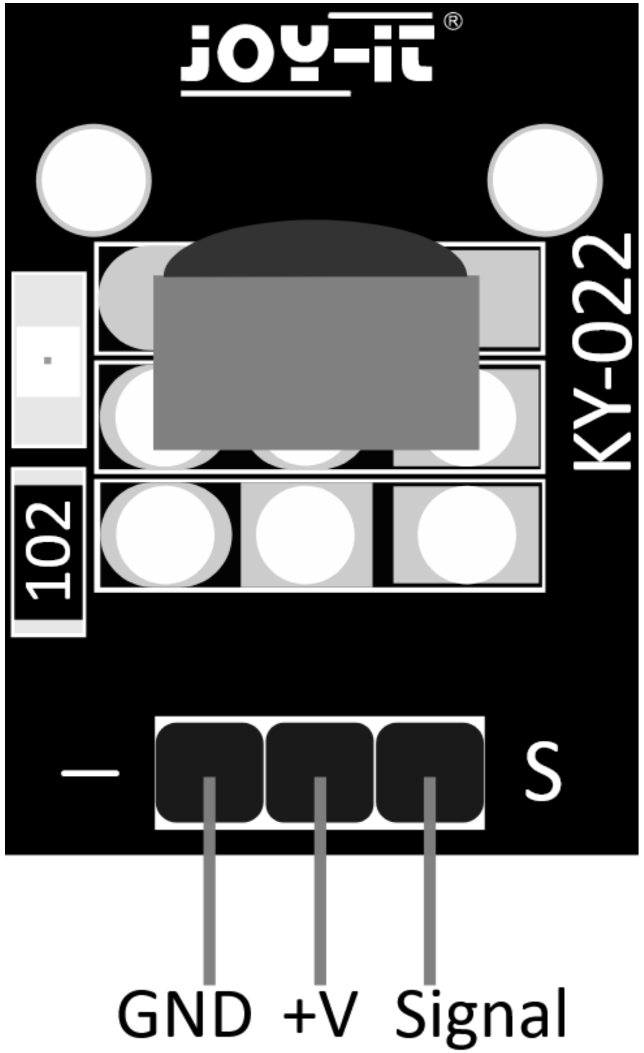
▲ 0x1FEA05F

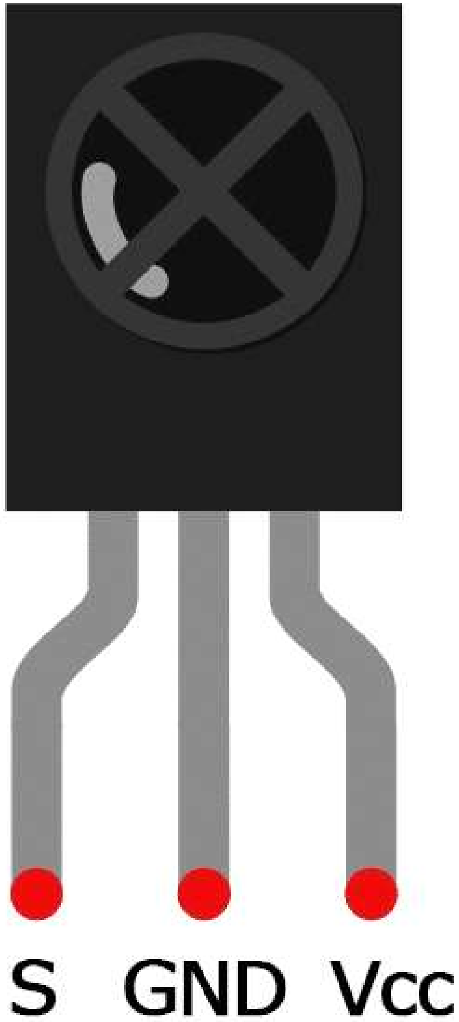
◀ 0x1FEE01F

▶ 0x1FE906F

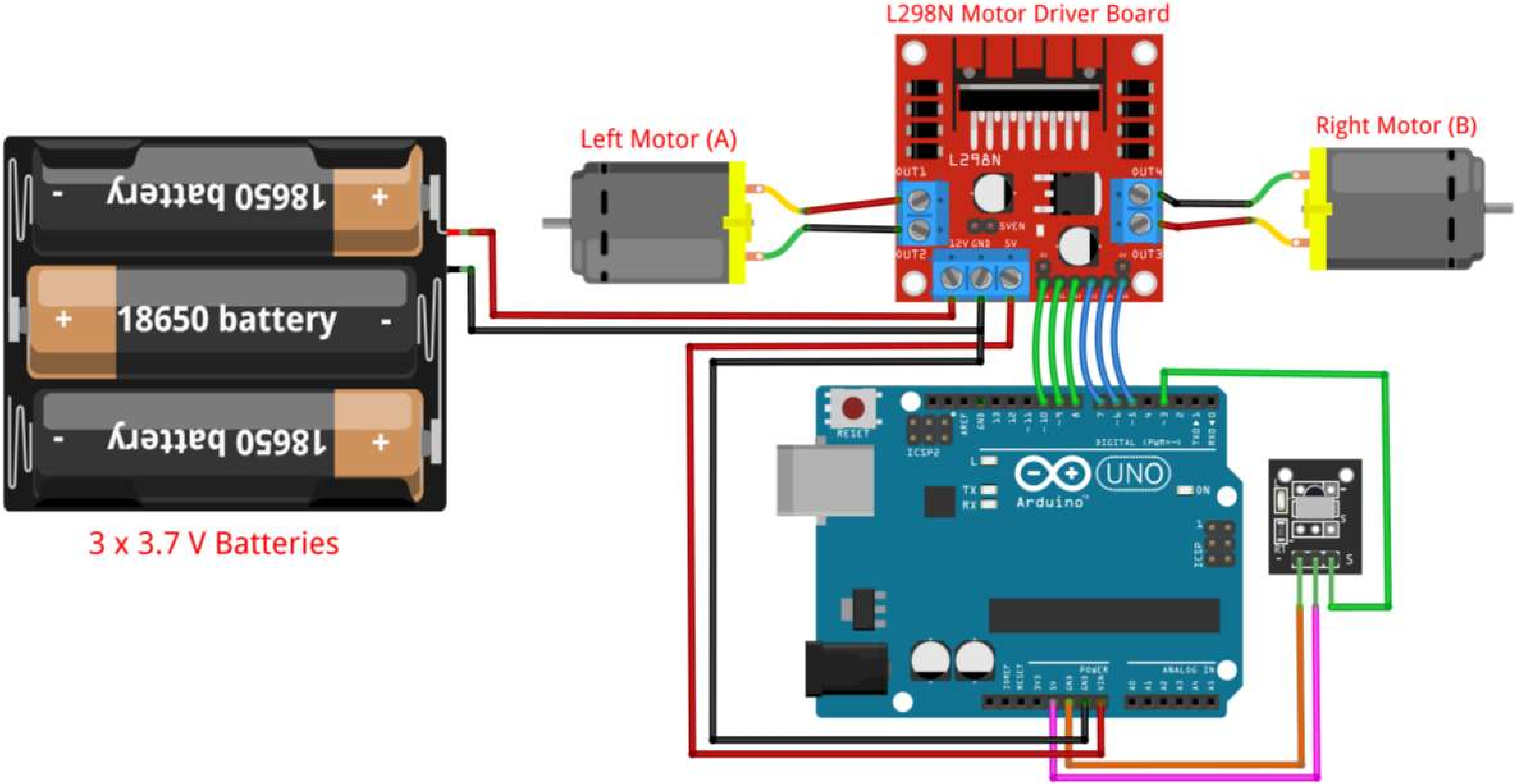
▼ 0x1FED827

Remote Car: IR Receiver





Remote Car: Circuit



L298N Motor Driver Board

Left Motor (A)

Right Motor (B)

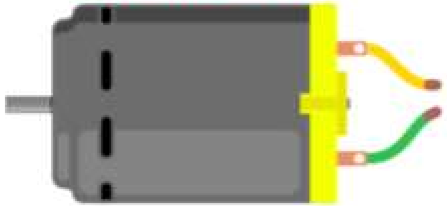
3 x 3.7 V Batteries

Remote Car: Components

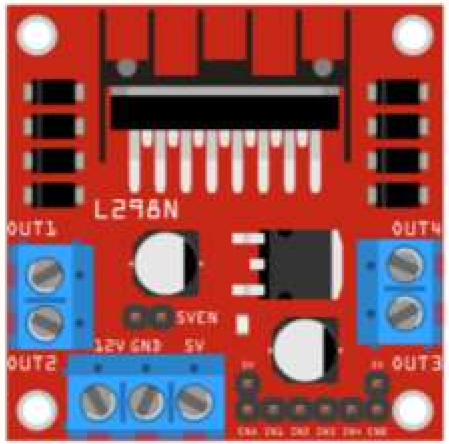
18650 battery
18650 battery
18650 battery
+
-

3 x 3.7 V Batteries

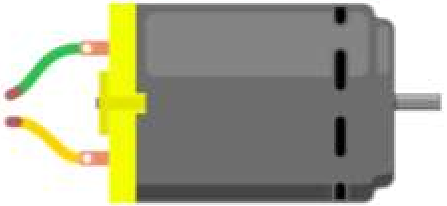
Left Motor (A)

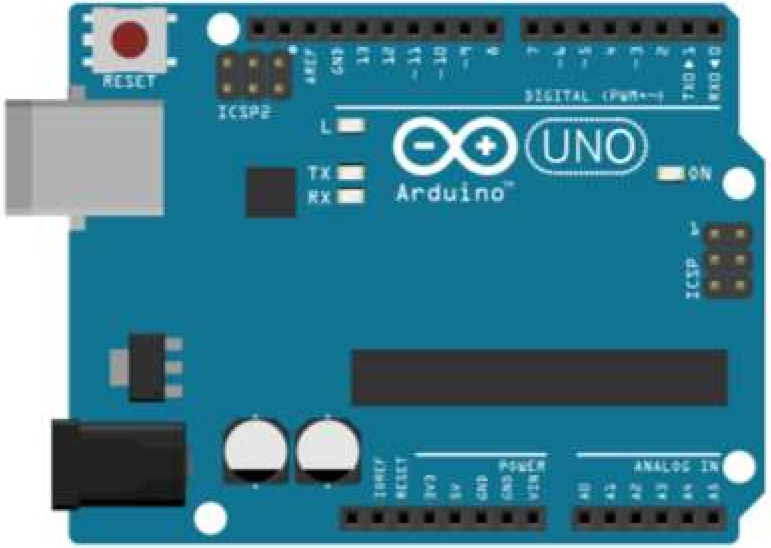


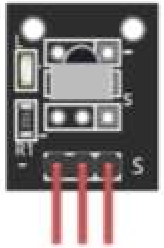
L298N Motor Driver Board



Right Motor (B)

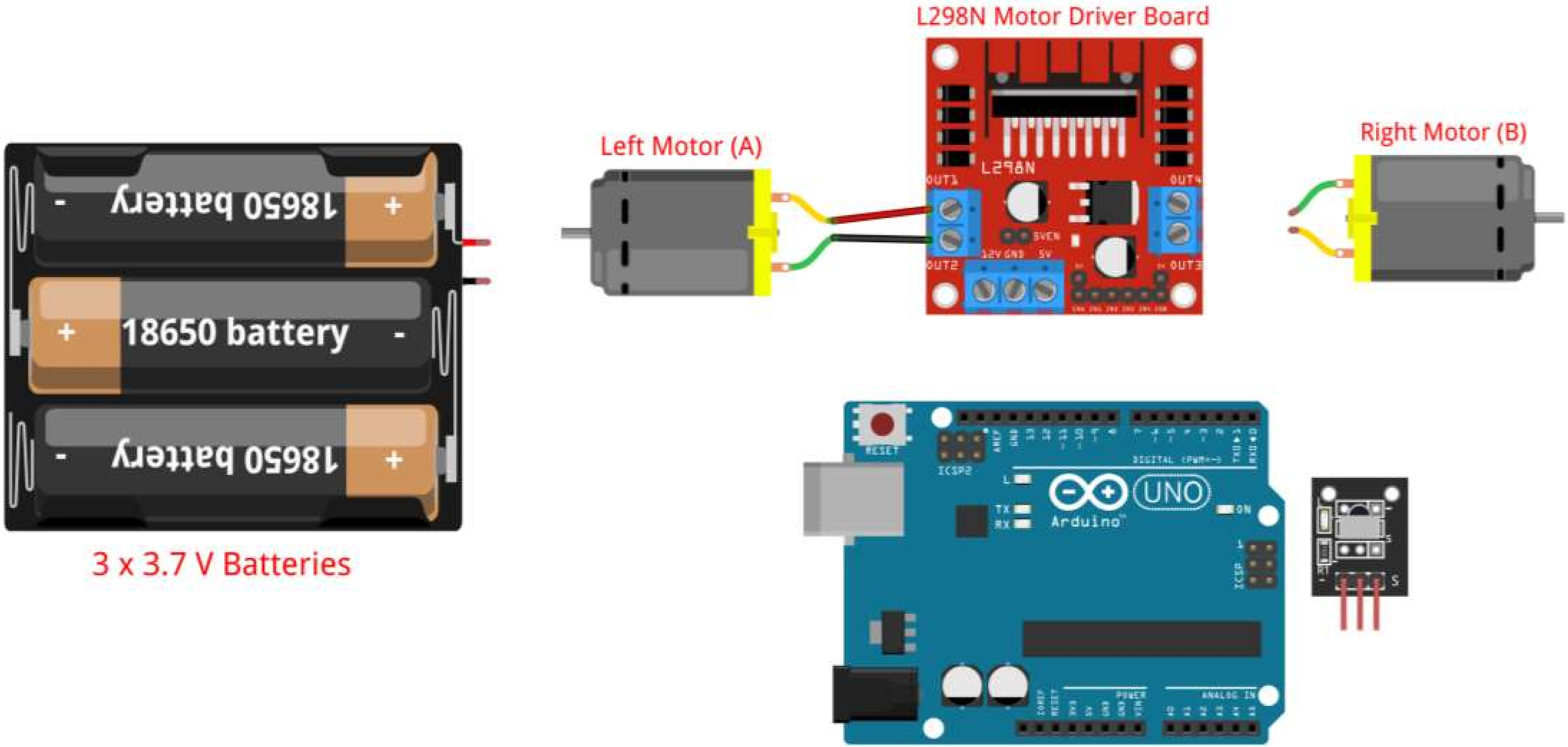






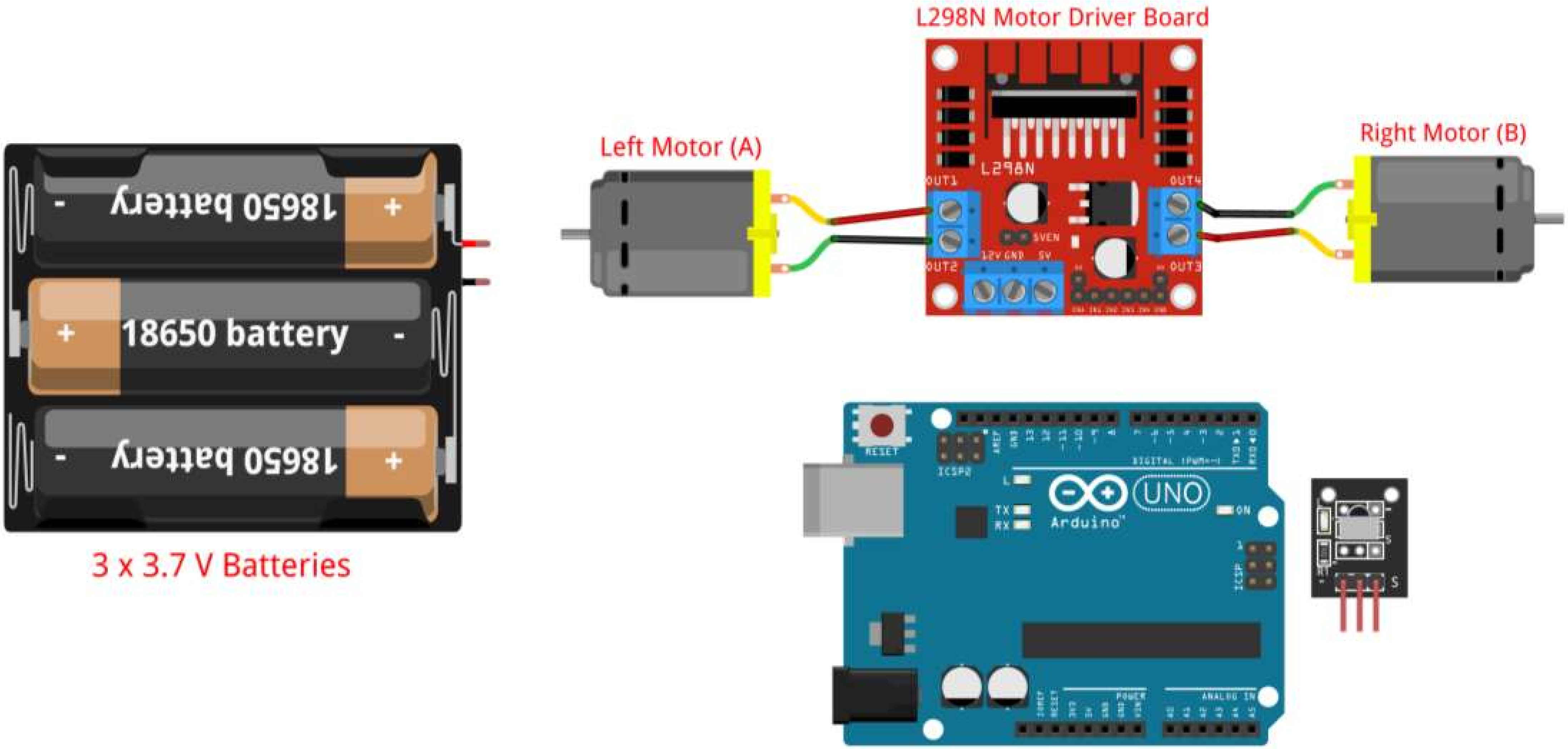
Remote Car: Steps

1. Connect the Left Motor (A) to OUT1 and OUT2 on L298N.



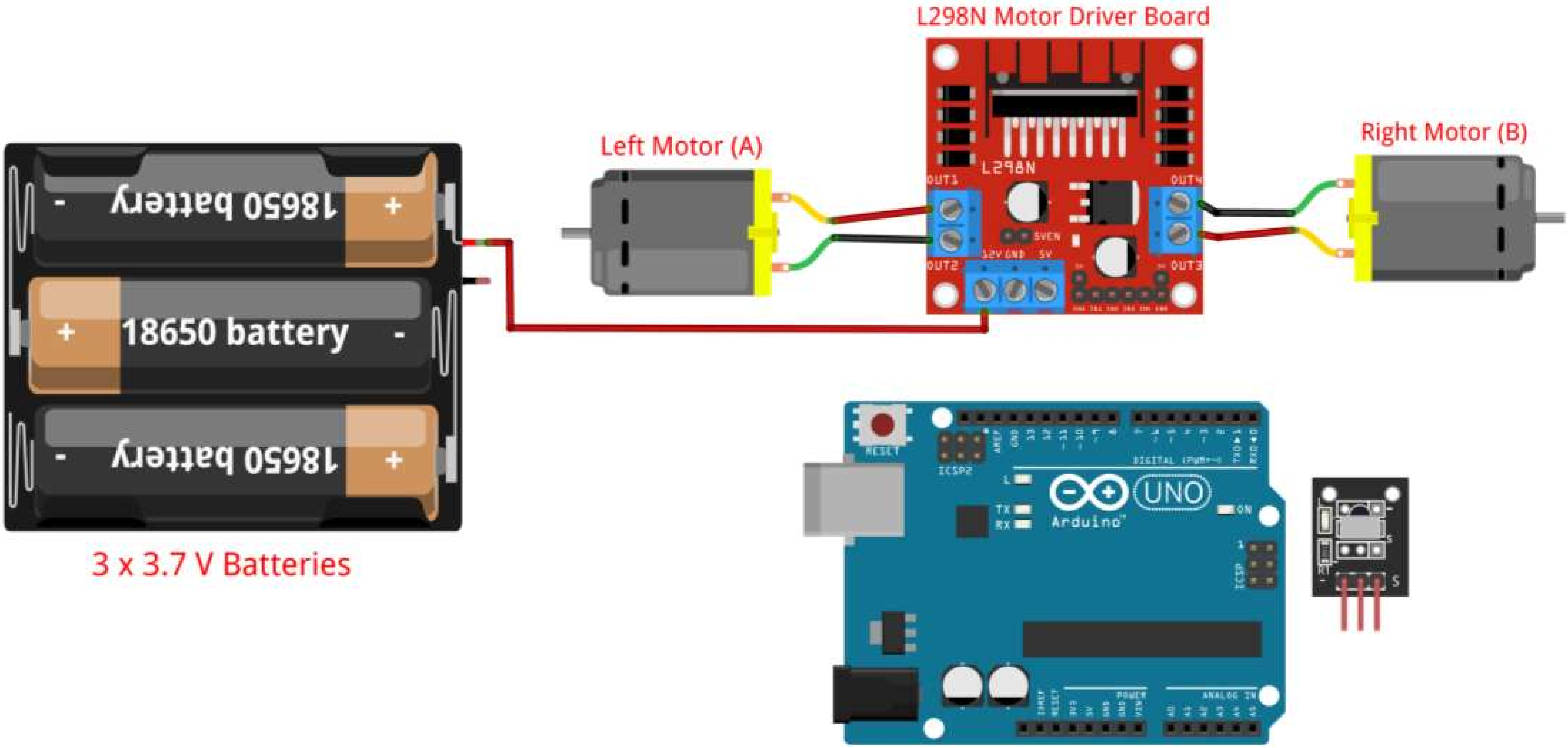
Remote Car: Steps

2. Connect the Right Motor (B) to OUT3 and OUT4 on L298N.



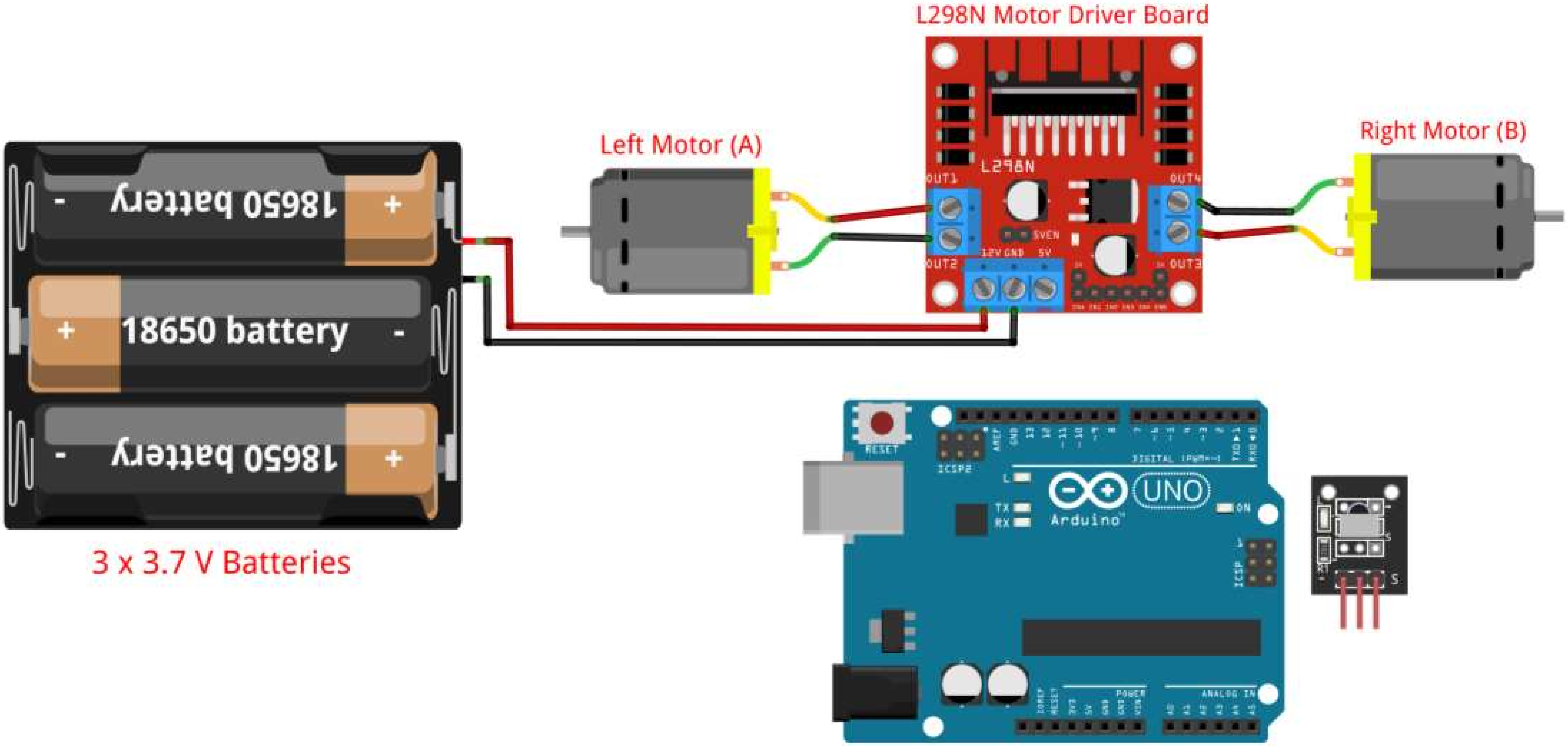
Remote Car: Steps

3. Connect the Positive Terminal (+) of Battery to 12V on L298N.



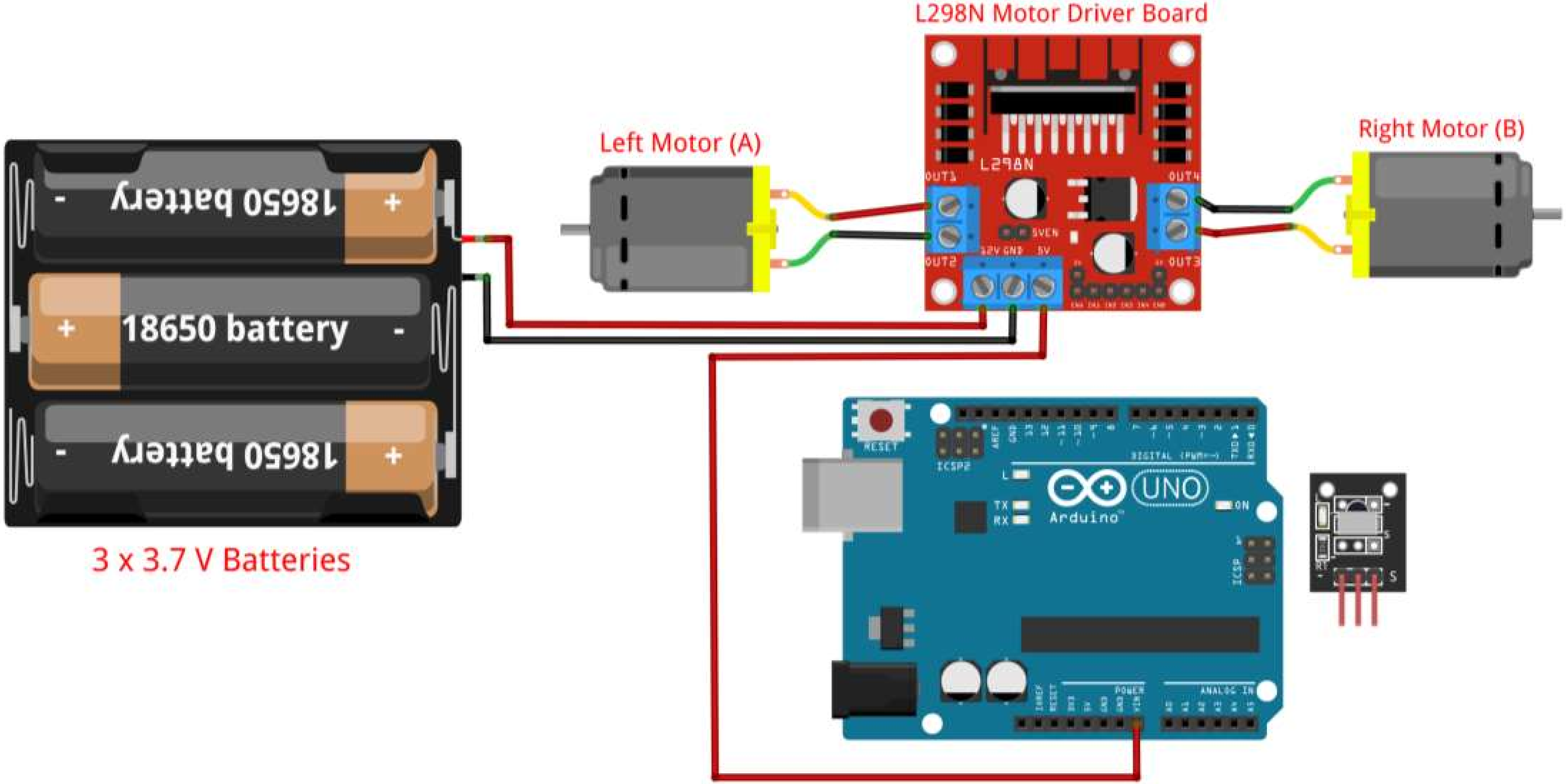
Remote Car: Steps

4. Connect the Negative Terminal (-) of Battery to GND pin on L298N.



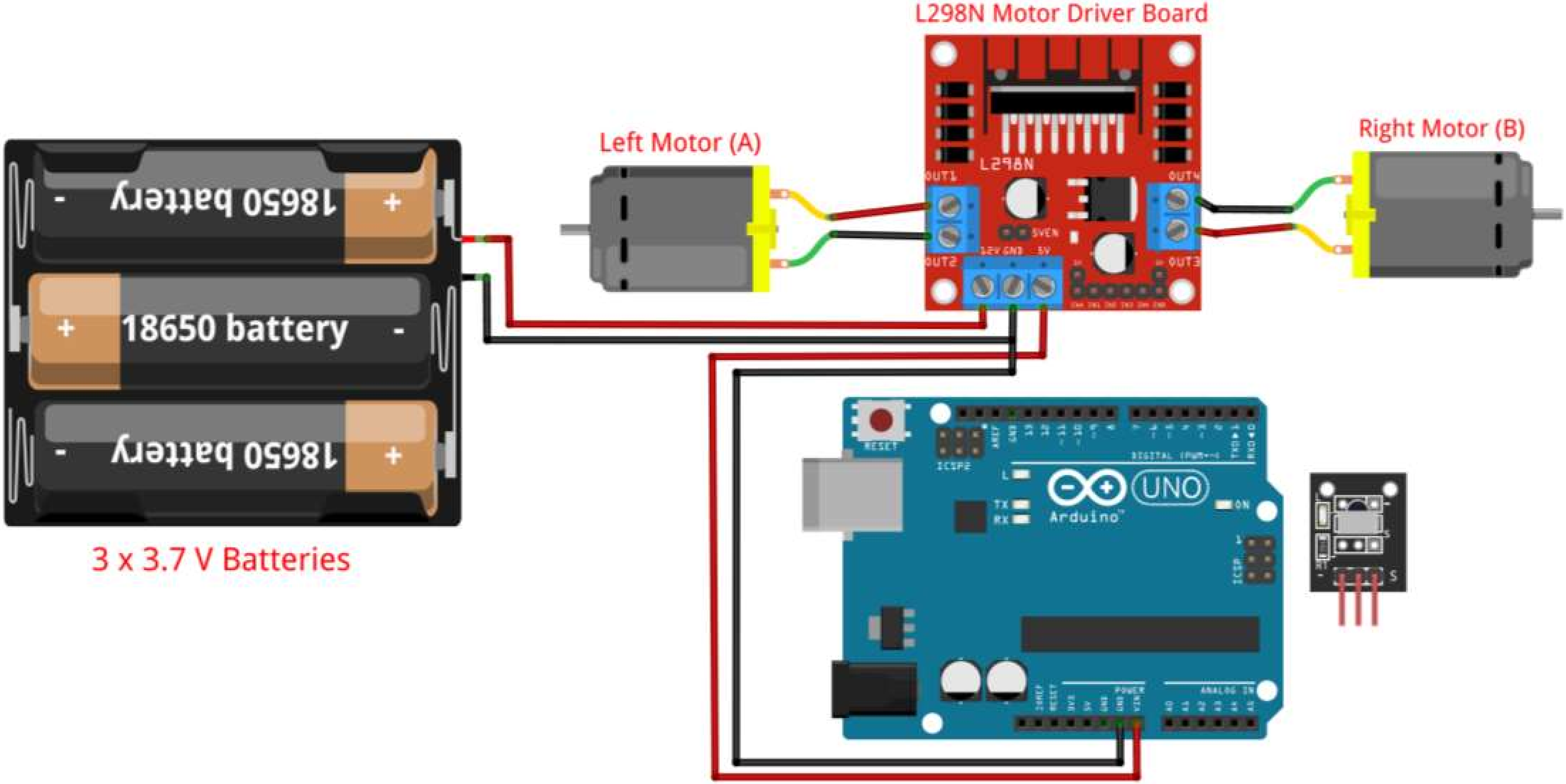
Remote Car: Steps

5. Connect the 5V of L298N to VIN pin on Arduino.



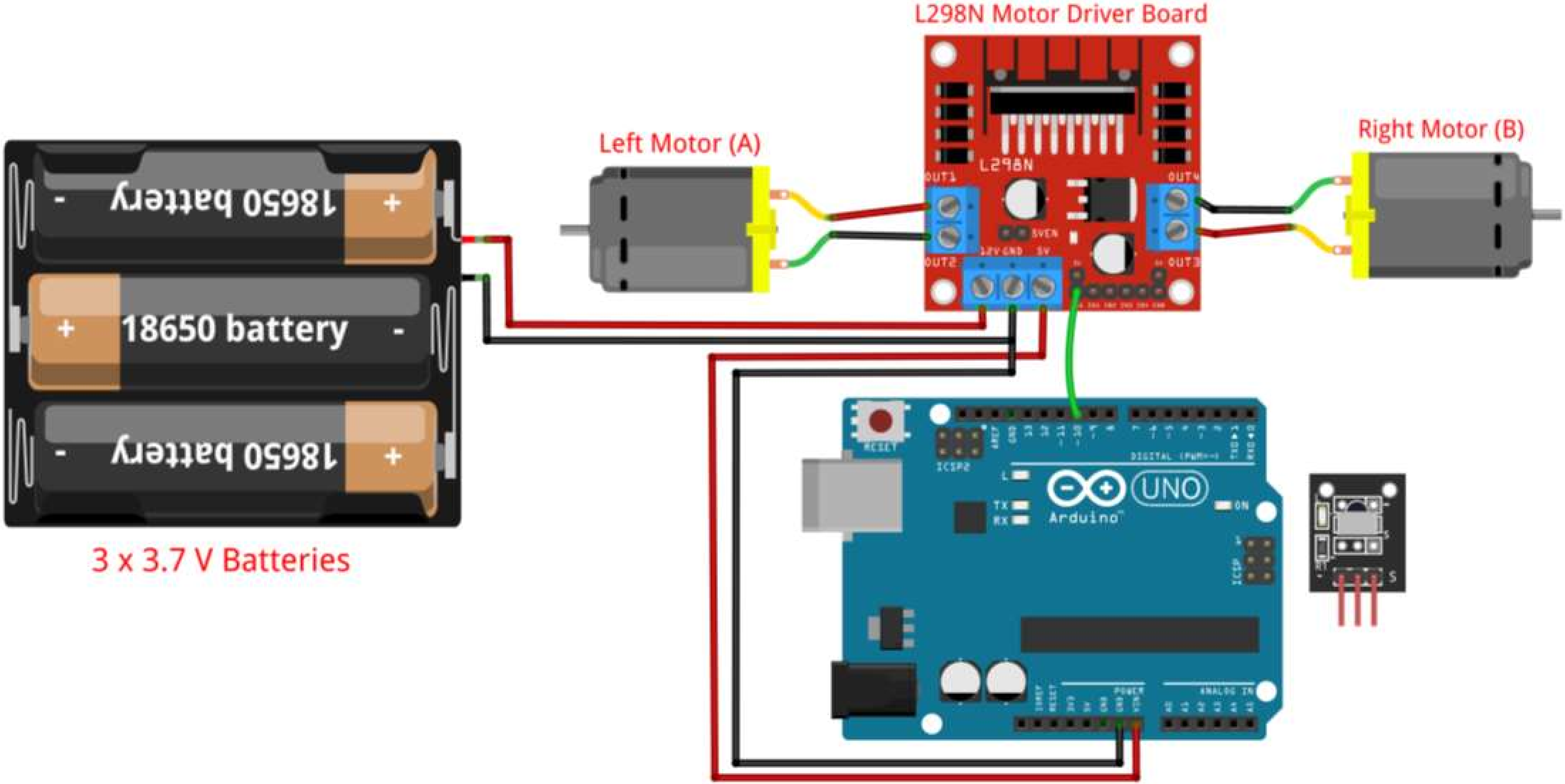
Remote Car: Steps

6. Connect the GND of L298N to GND pin on Arduino.



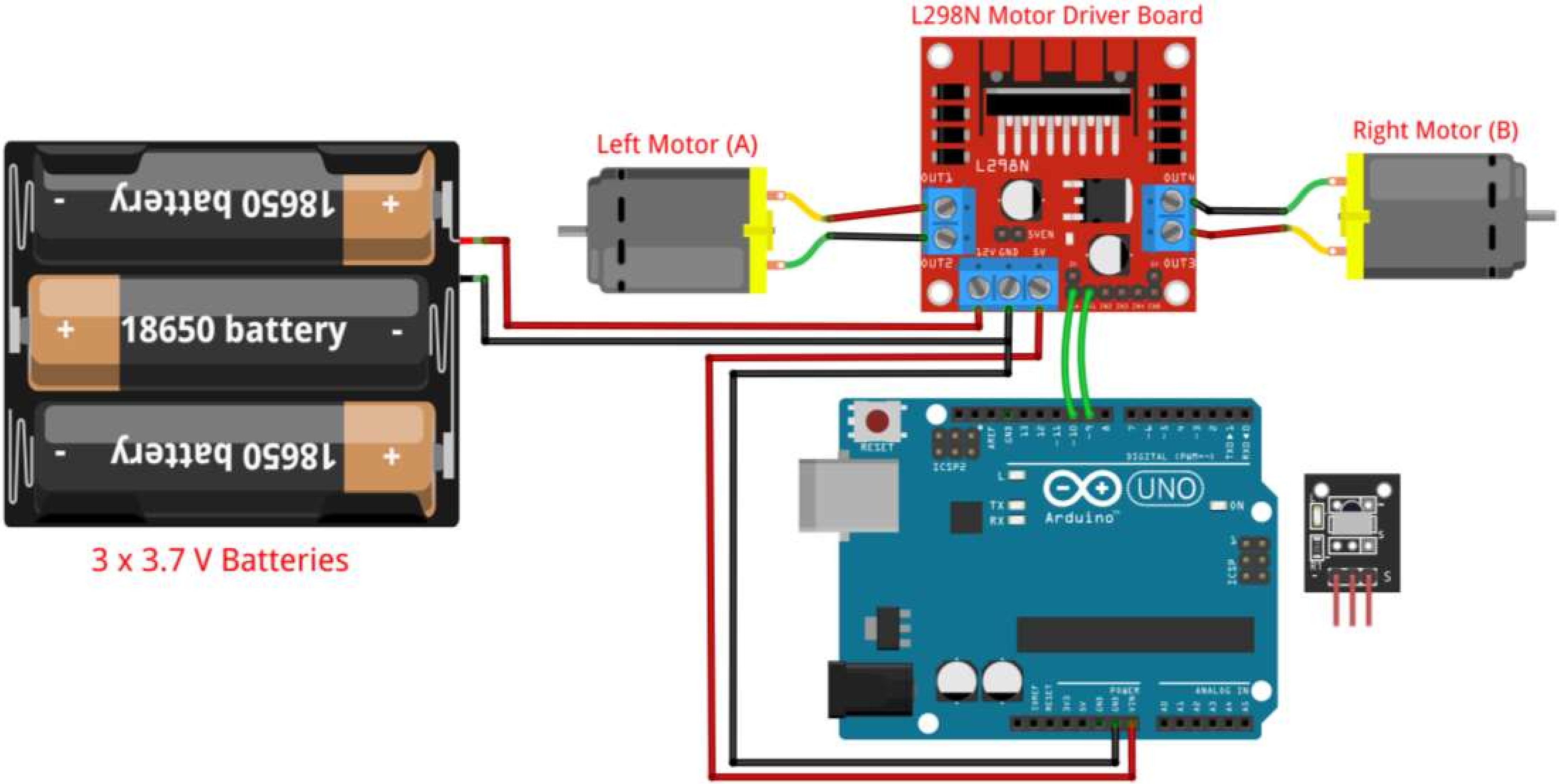
Remote Car: Steps

7. Connect the ENA of L298N to pin 10 on Arduino.



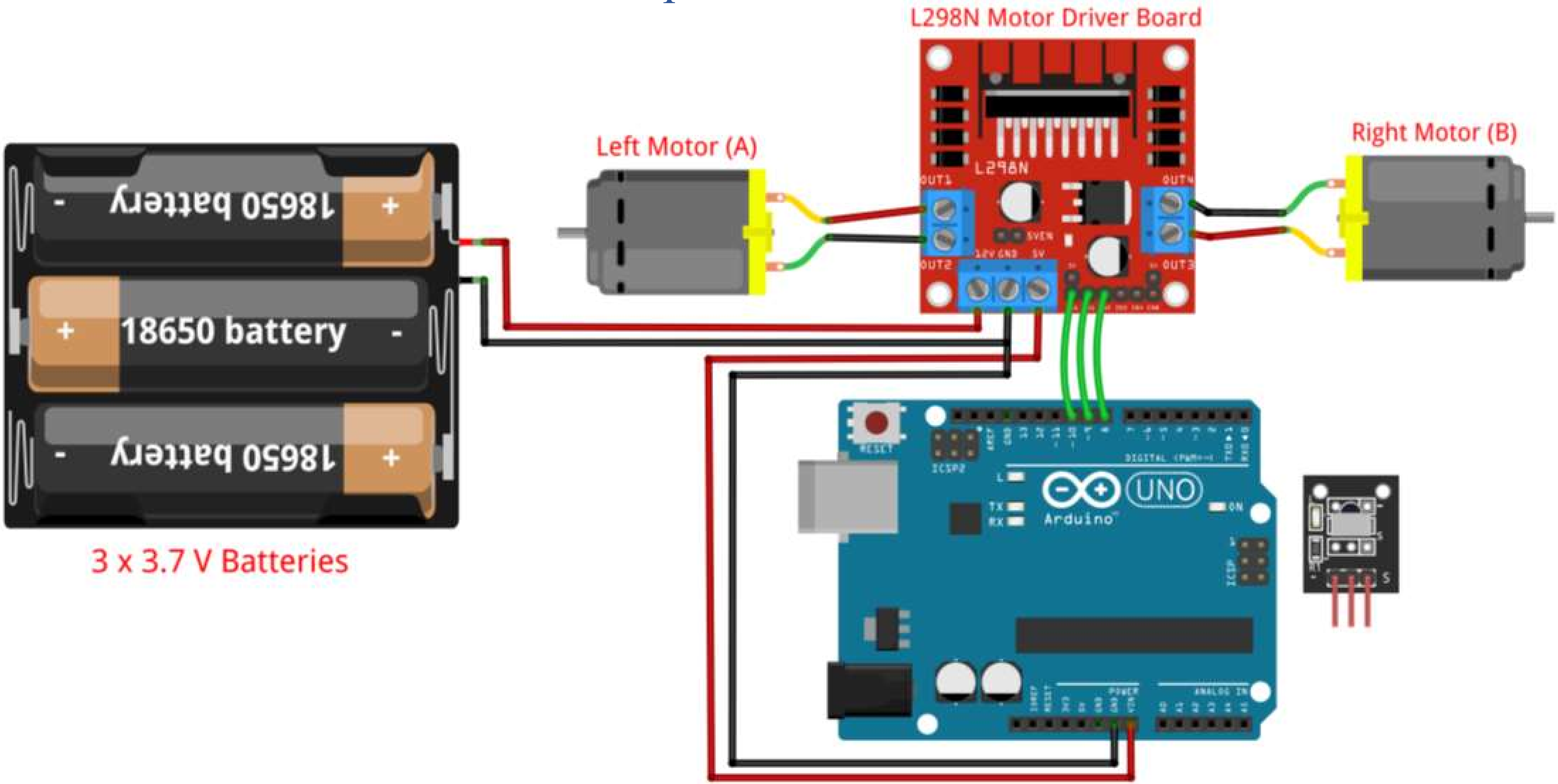
Remote Car: Steps

8. Connect the IN1 of L298N to pin 9 on Arduino.



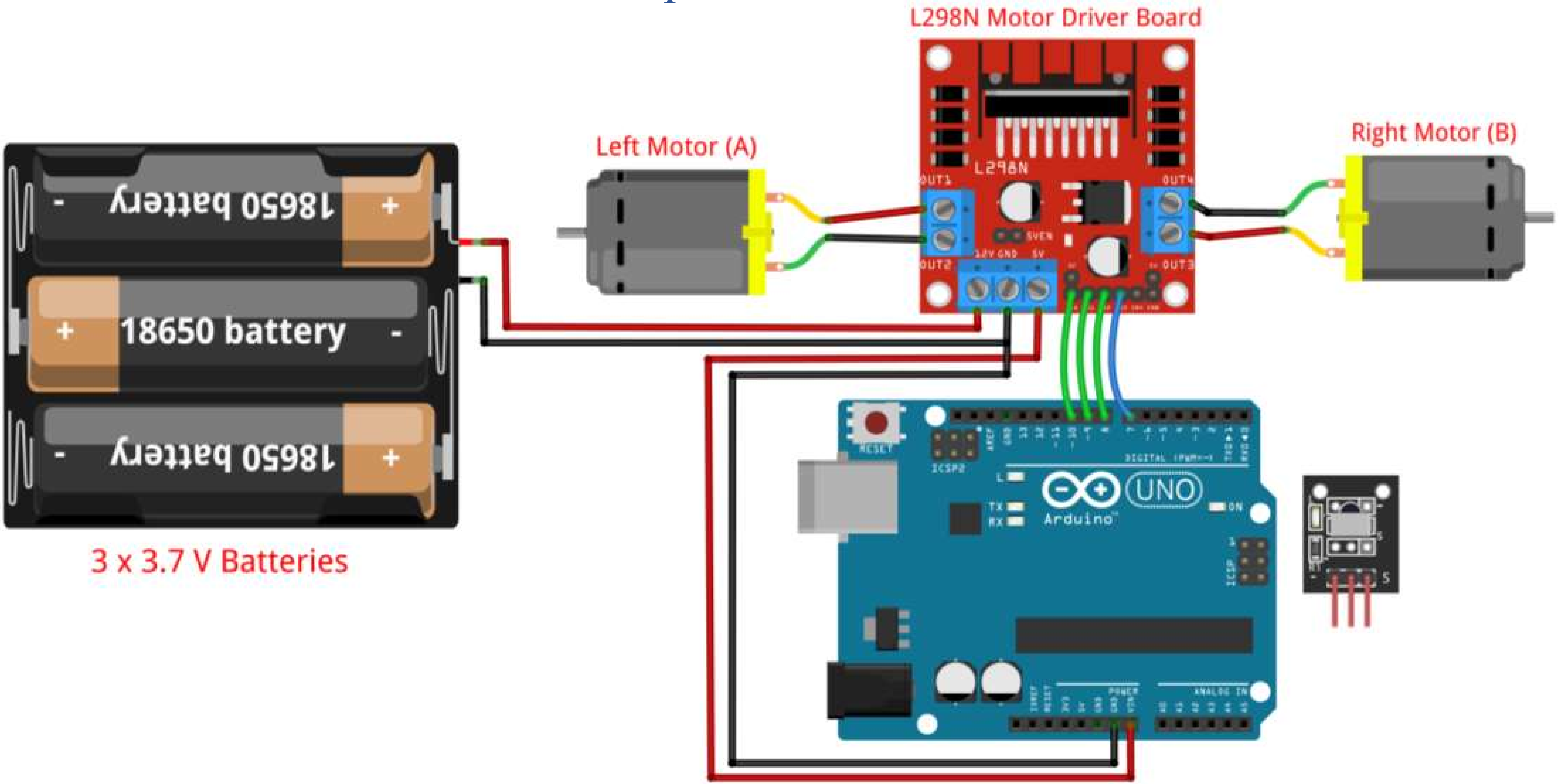
Remote Car: Steps

9. Connect the IN2 of L298N to pin 8 on Arduino.



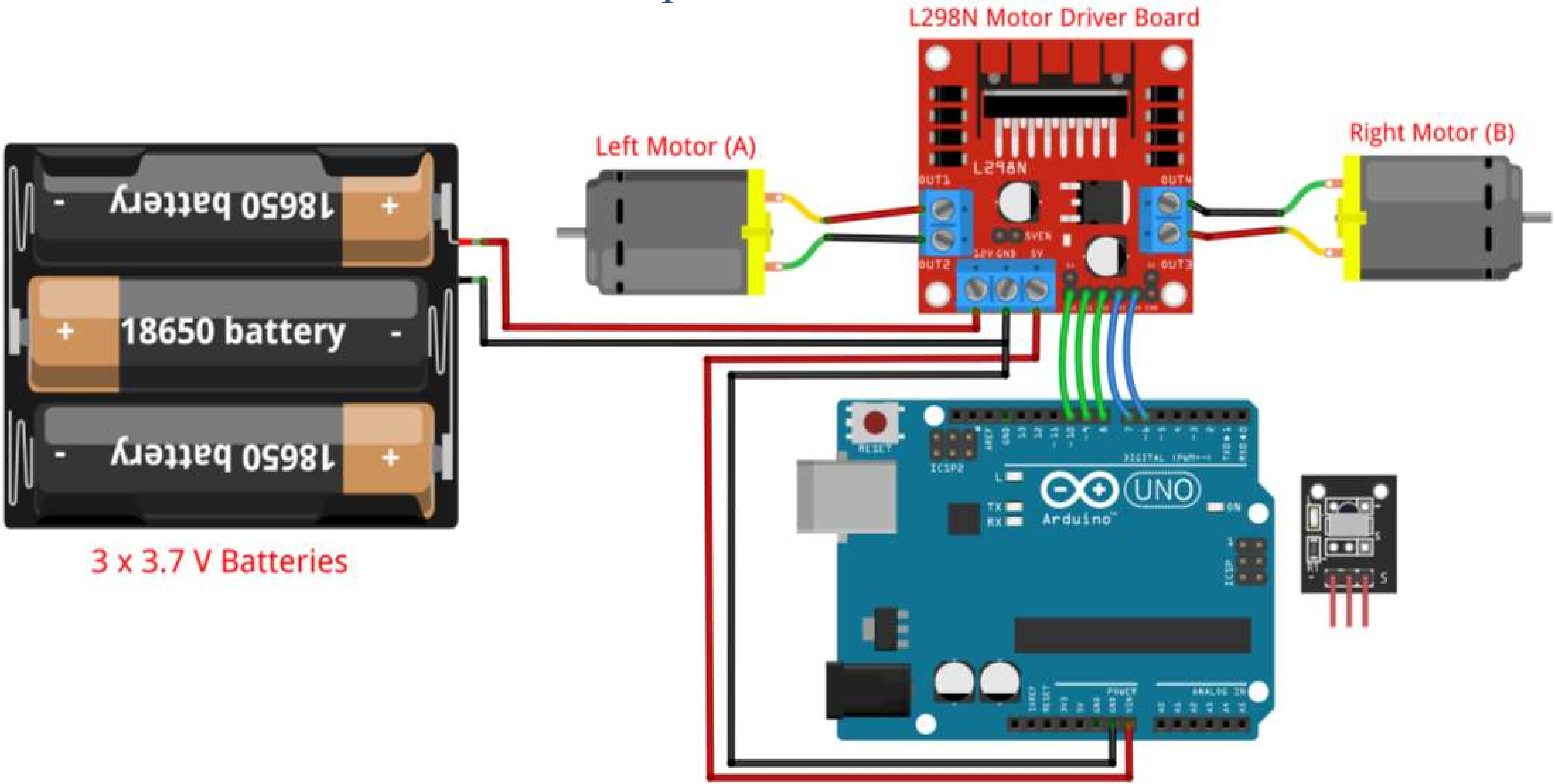
Remote Car: Steps

10. Connect the IN3 of L298N to pin 7 on Arduino.



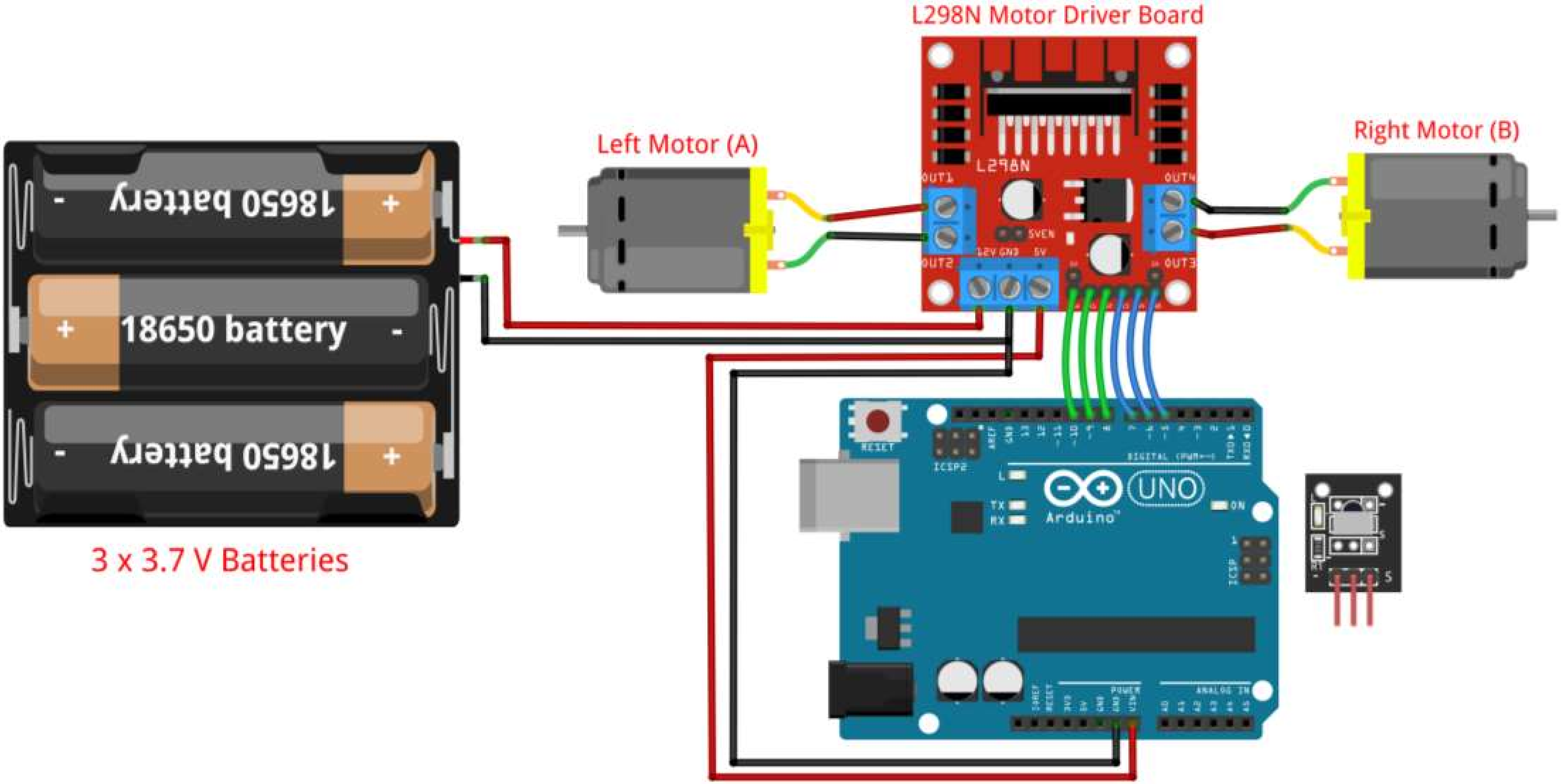
Remote Car: Steps

11. Connect the IN4 of L298N to pin 6 on Arduino.



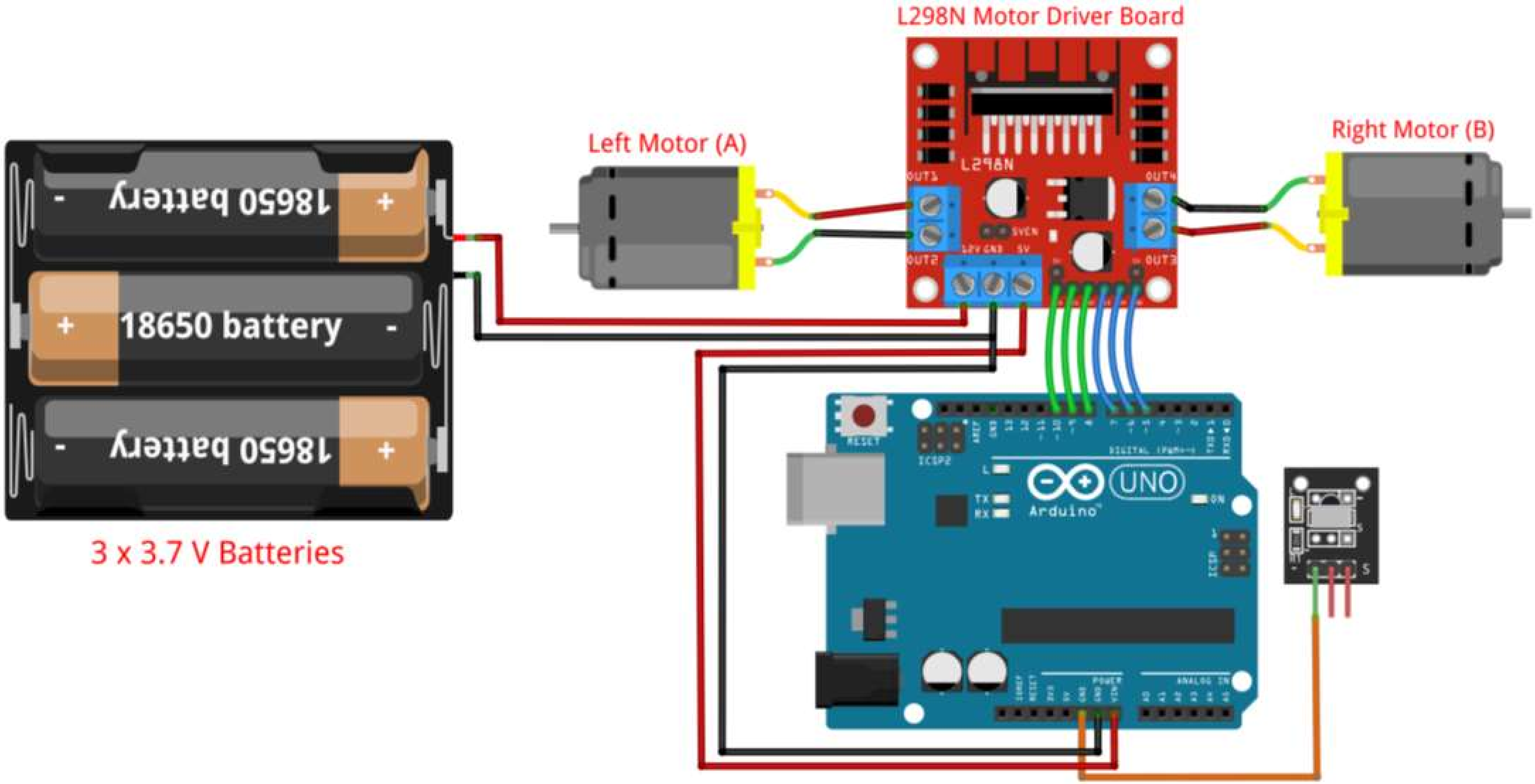
Remote Car: Steps

12. Connect the ENB of L298N to pin 5 on Arduino.



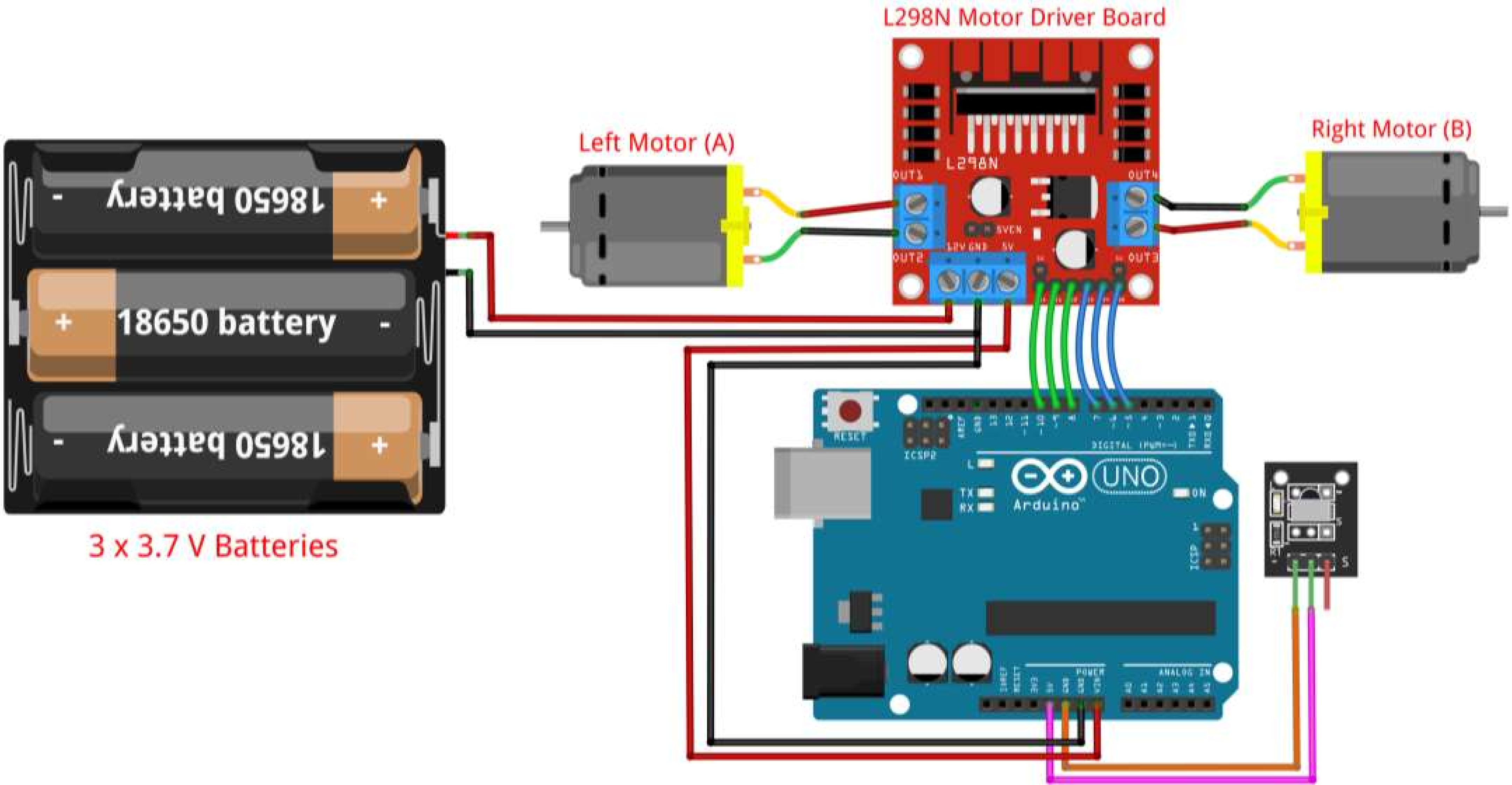
Remote Car: Steps

13. Connect the GND pin (-) of IR Receiver to GND on Arduino.



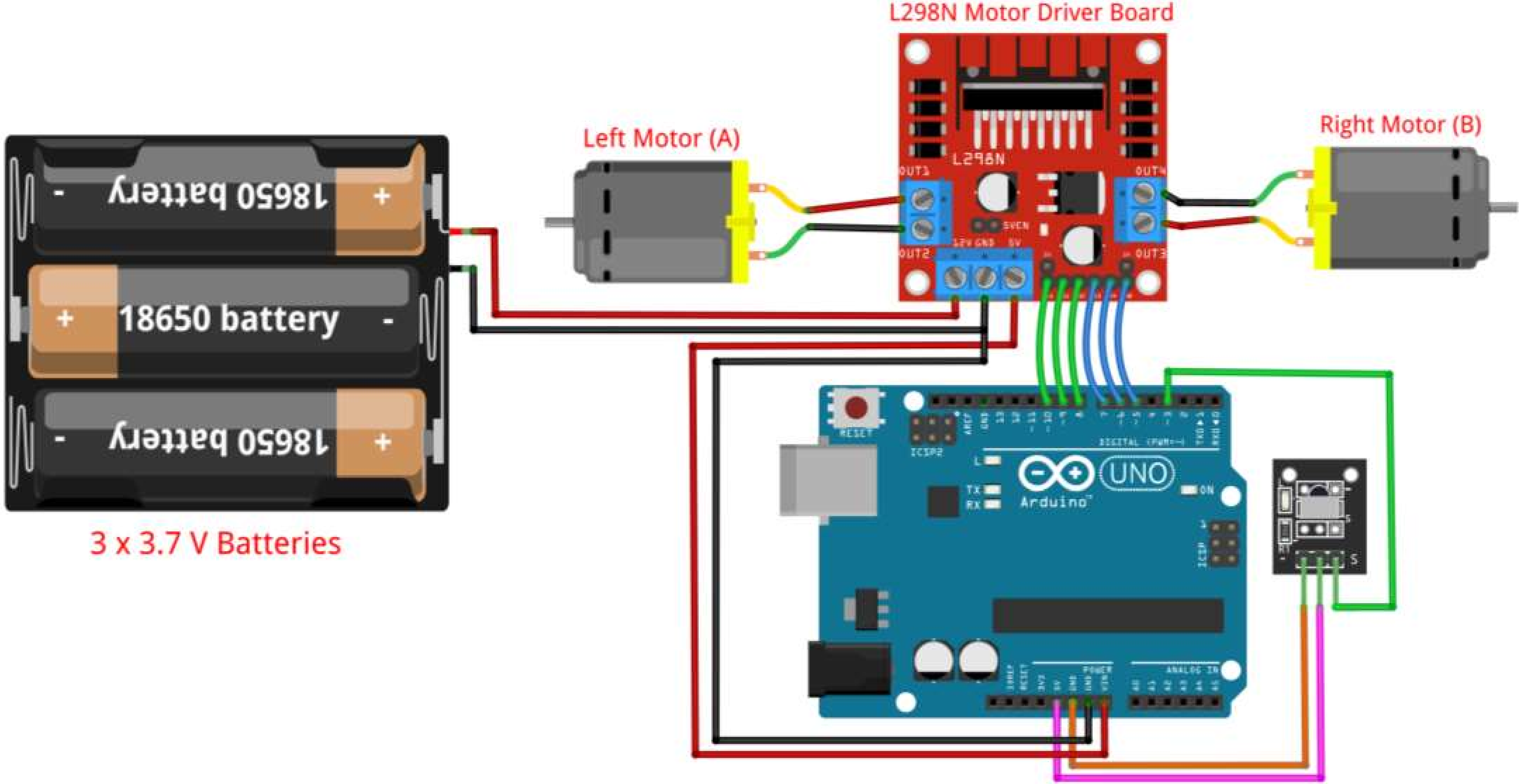
Remote Car: Steps

14. Connect the VCC pin of IR Receiver to 5V OR 3.3V on Arduino.



Remote Car: Steps

15. Connect the Signal pin (S) of IR Receiver to pin 3 on Arduino.



Remote Car: Code – Pins

```
// IRremote Library Pins
#include <IRremote.h>                       // Include IRremote library
#define RECV_PIN 3                          // Connect IR receiver pin 3

// Motor A Pins                             // Left Motor Pins
#define ENA 10                              // Connect ENA to pin 10
#define IN1 9                               // Connect IN1 to pin 9
#define IN2 8                               // Connect IN2 to pin 8

// Motor B Pins                             // Right Motor Pins
#define ENB 5                               // Connect ENB to pin 5
#define IN3 7                               // Connect IN3 to pin 7
#define IN4 6                               // Connect IN4 to pin 6

IRrecv irrecv(RECV_PIN);                    // Create IRrecv object
decode_results results;                     // Variable to hold the button code

int Speed = 120;                            // Variable to hold the car speed
```

Remote Car: Code – Pins Mode

```
void setup() {
  Serial.begin(9600);                       // Begin serial communication
  irrecv.enableIRIn();                      // Start the receiver

  pinMode(ENA, OUTPUT);                     // Set ENA pin as output
  pinMode(ENB, OUTPUT);                     // Set ENB pin as output
  pinMode(IN1, OUTPUT);                     // Set IN1 pin as output
  pinMode(IN2, OUTPUT);                     // Set IN2 pin as output
  pinMode(IN3, OUTPUT);                     // Set IN3 pin as output
  pinMode(IN4, OUTPUT);                     // Set IN4 pin as output
}
```

Remote Car: Code – Control

```
void loop() {
  if (irrecv.decode(&results)) {                  // If a button is pressed
    Serial.println(results.value, HEX);           // Print button hex code

    if(results.value == 0x2FD48B7)                // Stop car
      stop_car();
    else if(results.value == 0x2FDD827)           // Move forward
      move_forward();
    else if(results.value == 0x2FDF807)           // Move backward
      move_backward();
    else if(results.value == 0x2FD58A7)           // Turn right
      turn_right();
    else if(results.value == 0x2FD7887)           // Turn left
      turn_left();

    irrecv.resume();                              // Receive the next value
  }

  delay(100);                                     // Short delay
}
```

Remote Car: Code – Stop Car

```
// Stop the car by setting INs to LOW

void stop_car(){
  // Motor A                                  // Stop Left Motor
  digitalWrite(IN1, LOW);                     // Set IN1 to LOW
  digitalWrite(IN2, LOW);                     // Set IN2 to LOW
  analogWrite(ENA, 0);                        // Set motor A speed to 0

  // Motor B                                  // Stop Right Motor
  digitalWrite(IN3, LOW);                     // Set IN3 to LOW
  digitalWrite(IN4, LOW);                     // Set IN4 to LOW
  analogWrite(ENB, 0);                        // Set motor B speed to 0
}
```

Remote Car: Code – Move Forward

```
// Move the car forward by moving both motors

void move_forward(){
  // Motor A                                // Move Left Motor
  digitalWrite(IN1, HIGH);                  // Set IN1 to HIGH
  digitalWrite(IN2, LOW);                   // Set IN2 to LOW
  analogWrite(ENA, Speed);                  // Set motor A speed

  // Motor B                                // Move Right Motor
  digitalWrite(IN3, HIGH);                  // Set IN3 to HIGH
  digitalWrite(IN4, LOW);                   // Set IN4 to LOW
  analogWrite(ENB, Speed);                  // Set motor B speed
}
```

Remote Car: Code – Move Backward

```
// Move the car backward by moving both motors in reverse

void move_backward(){
  // Motor A                                  // Move Left Motor in Reverse
  digitalWrite(IN1, LOW);                     // Set IN1 to LOW
  digitalWrite(IN2, HIGH);                    // Set IN2 to HIGH
  analogWrite(ENA, Speed);                    // Set motor A speed

  // Motor B                                  // Move Right Motor in Reverse
  digitalWrite(IN3, LOW);                     // Set IN3 to LOW
  digitalWrite(IN4, HIGH);                    // Set IN4 to HIGH
  analogWrite(ENB, Speed);                    // Set motor B speed
}
```

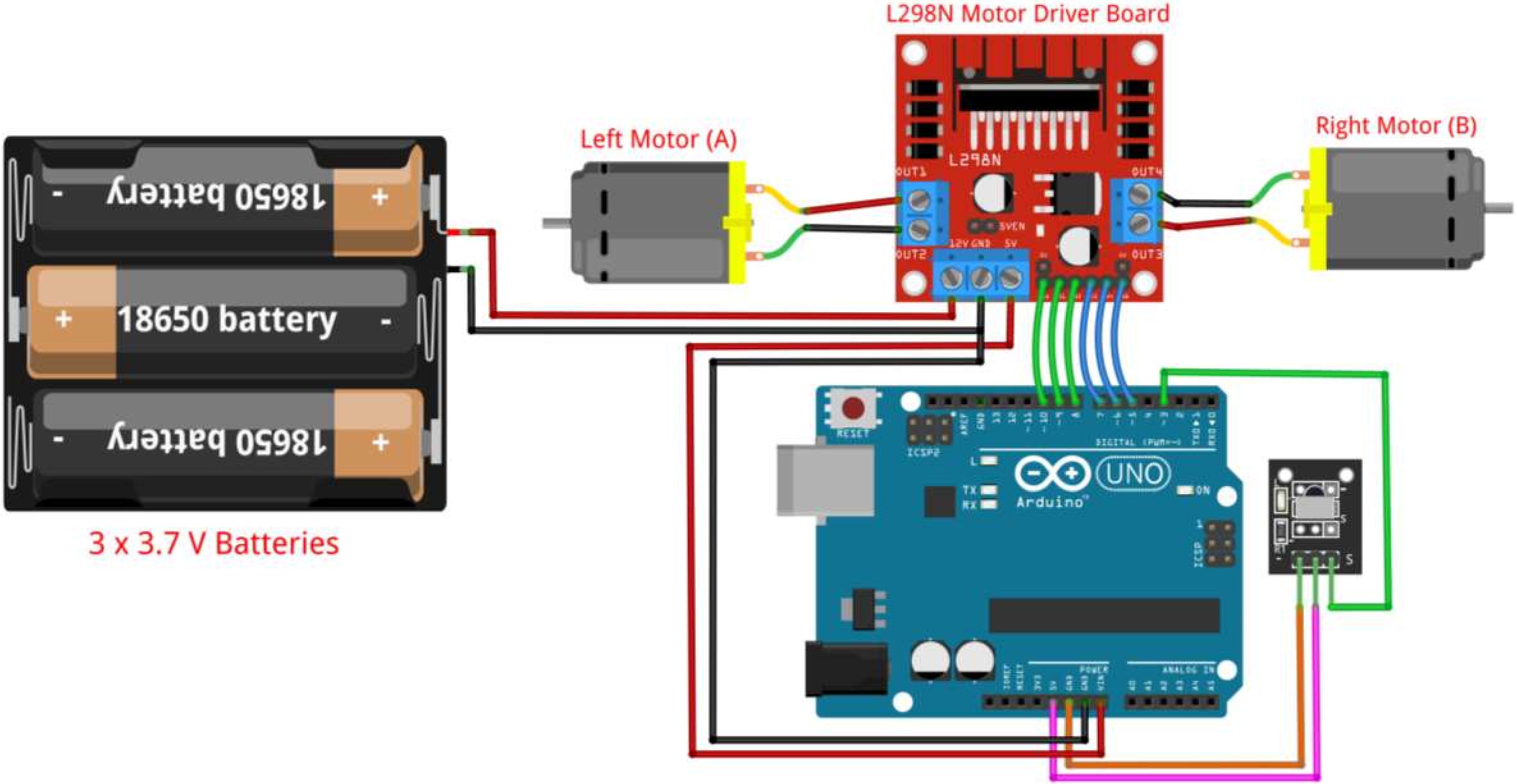
Remote Car: Code – Turn Right

```
// Turn the car right by moving left motor, and stopping right motor

void turn_right(){
  // Motor A                                  // Move Left Motor
  digitalWrite(IN1, HIGH);                    // Set IN1 to HIGH
  digitalWrite(IN2, LOW);                     // Set IN2 to LOW
  analogWrite(ENA, Speed);                    // Set motor A speed

  // Motor B                                  // Stop Right Motor
  digitalWrite(IN3, LOW);                     // Set IN3 to LOW
  digitalWrite(IN4, LOW);                     // Set IN4 to LOW
  analogWrite(ENB, 0);                        // Set motor B speed to 0
}
```

Remote Car: Code – Turn Left

```
// Turn the car left by stopping left motor, and moving right motor

void turn_left(){
  // Motor A                                // Stop Left Motor
  digitalWrite(IN1, LOW);                   // Set IN1 to LOW
  digitalWrite(IN2, LOW);                   // Set IN2 to LOW
  analogWrite(ENA, 0);                      // Set motor A speed to 0

  // Motor B                                // Move Right Motor
  digitalWrite(IN3, HIGH);                  // Set IN3 to HIGH
  digitalWrite(IN4, LOW);                   // Set IN4 to LOW
  analogWrite(ENB, Speed);                  // Set motor B speed
}
```

Assignment 05: Remote Car



References

- Interfacing IR Sensor Module with Arduino
- Interface L298N DC Motor Driver Module with Arduino
- L298N Motor Driver – How It Works
- How to Use the L298 Motor Driver Module
- How to use the L298N Motor Driver
- How to Make Line Follower Robot Using Arduino
- Line Follower Robot using Arduino and L298N Module
- How to Make a Line Follower Robot Using Arduino