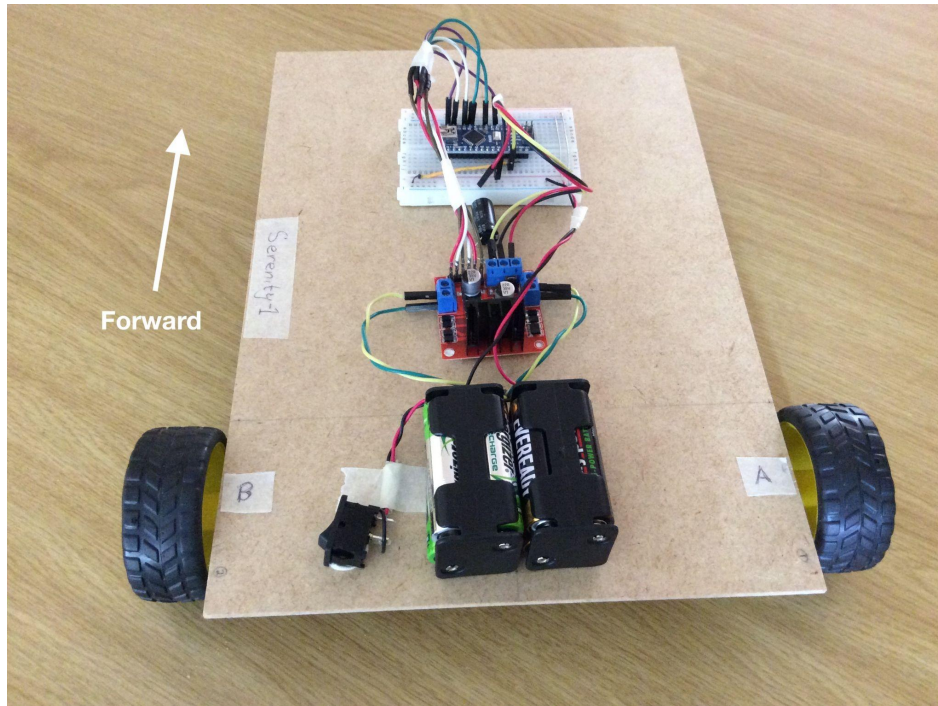


## Serenity-1 - Movement Basics



### Motor A

Pins on the Arduino are connected to ENA, IN1 and IN2 on the H-Bridge.

A HIGH on IN1 (and a LOW on IN2) sets Motor A rotation direction to: forward.

A HIGH on IN2 (and a LOW on IN1) sets Motor A rotation direction to: reverse.

A PWM signal, `analogWrite(pin_num, pwm_value)`, to ENA causes the motor to start, and the PWM value (0-255) sets the motor rotation speed. Alternatively, a `digitalWrite(pin_num, HIGH)` to ENA causes the motor to start and rotate at full speed.

## Motor B

Pins on the Arduino are connected to ENB, IN3 and IN4 on the H-Bridge.

A HIGH on IN4 (and a LOW on IN3) sets Motor B rotation direction to: forward.

A HIGH on IN3 (and a LOW on IN4) sets Motor A rotation direction to: reverse.

A PWM signal, `analogWrite(pin_num, pwm_value)`, to ENB causes the motor to start, and the PWM value (0-255) sets the motor rotation speed. Alternatively, a `digitalWrite(pin_num, HIGH)` to ENB causes the motor to start and rotate at full speed.

## Straight Line Movement Calibration

If both Motor A and Motor B are turning forwards but Motor A is turning faster than Motor B then the robot will turn left. If Motor B is turning faster than Motor A then the robot will turn right.

The speeds (PWM values) of the two motors need to be adjusted to make the robot move forward in a straight line. We get the PWM values by trial and error.

When calibrating it helps to determine the time that the robot takes to rotate 360 degrees. This can be used to roughly set specific turning angles.

## Basic Movements

### 1. Move forward

- Move Motor A forwards
- Move Motor B forwards

### 2. Reverse

- Move Motor A backwards
- Move Motor B backwards

3. Turn right and pivot on Wheel A
  - Move Motor B forwards
  - Stop Motor A
4. Turn left and pivot on Wheel B
  - Move Motor A forwards
  - Stop Motor B
5. Rotate the robot clockwise. The pivot point will be half-way between Wheel A and Wheel B.
  - Move Motor B forwards
  - Move Motor A backwards
  - Run both motors at the same rotational speed

### Arduino Code Example

```
void loop() {  
  
    /*  
    For each loop iteration, we are specifying the sequence of actions  
    we want to happen AND how long we want each action to take.  
    If we don't specify how long we want the action to take then  
    the robot won't perform that action.  
    We specify "how long" by adding a delay after that action.  
    */  
  
    move_forward();  
    delay(2000);  
  
    reverse();  
    delay(1000);  
  
    rotate_left();  
    delay(1400);  
  
    move_forward();  
    delay(1000);  
}
```