



LESSON D-11

MOTOR PWM & CALIBRATION

In the previous lesson, you connected the drive motors on your robot to the L293DNE motor drive IC and tested both motors in both directions. This test was conducted with the motors running at full speed, but there are often times when you might want the motors to run at lower speeds:

- The robot is near an object.
- The robot is trying to follow a line using its sensors.
- The robot is turning too quickly to be easily controlled remotely.

The input lines of the motor drive IC are digital, meaning they will only recognize high or low signals. This means that the only options for driving those inputs are high or low, which will result in the connected motor either being off or driving at full speed. Since the speed of the motor can't be controlled by the motor input lines, the only way to control the speed is to use the enable lines of the motor drive IC.

PWM Control of DC Motors

You may remember from previous lessons that a PWM signal allows you to rapidly turn a signal on and off. We previously used a PWM output to control the brightness of an LED. With standard high and low signals, the LED will either be on or off. Brightness values in between can be attained by rapidly turning the LED on and off. If this high/low switching is done fast enough, the LED will appear to be on, but dim. This brightness value can be adjusted by manipulating the duty cycle of the signal which can be adjusted to any value between 0%, or never high, to 100%, or always high.

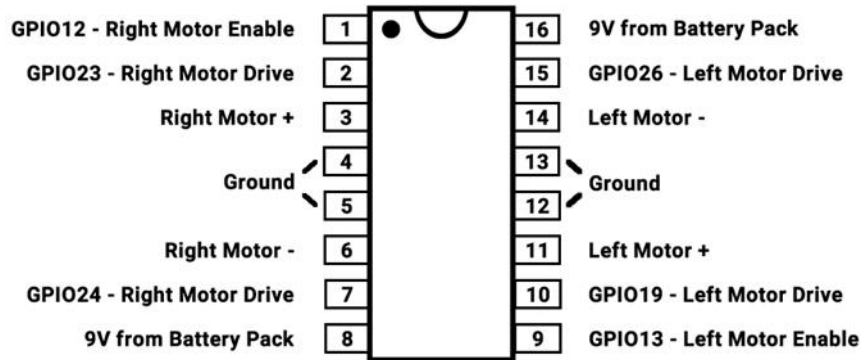
- A higher duty cycle means the signal is high more often, and the LED will be brighter.
- A lower duty cycle means the signal is high less often, and the LED will not be as bright.

These principles for controlling the brightness of an LED can be directly applied to speed control of a motor. By using a PWM signal to control the enable line of the motor drive IC, you can control how often that enable line will be high, which will control how often the enable line will allow the motors to drive.

Motor Drive IC Pinout

Here is the pinout of the L293DNE motor drive IC as it will be used in our robot:

L293DNE Robot Wiring



GPIO12 will be controlling the enable line for the right motor, and GPIO13 will be controlling the enable line for the left motor. Sending out PWM signals from these GPIO pins on the Pi will allow the speed of each motor to be controlled independently.

Pushing an enable line high will only cause a motor to move if one of the motor drive inputs to that channel is also high. Let's say for example, the right motor drive lines on GPIO23 and GPIO24 are both low. You can drive GPIO12 high permanently, or intermittently using PWM, but the right motor will still never move. The motor drive IC will pass the signals from GPIO23 and GPIO24 to the motors, but since both are low, the motor will not move. A high would also be required on either GPIO23 or GPIO24 in order to make that motor move:

GPIO12 (Enable)	GPIO23	GPIO24	Motor
Low	High	Low	No movement – Enable line is low
Low	Low	High	No movement – Enable line is low
High	High	Low	Fast movement
High	Low	High	Fast movement
PWM	High	Low	Speed controlled movement
PWM	Low	High	Speed controlled movement

Low Speed Motor Characteristics

PWM control of the enable lines on the motor drive IC is a great way to control the speed of motors, and high speeds are no problem. As the value of the PWM signal approaches 100% duty cycle, the motor will just run faster and faster, until it's running full speed.

Caution must be exercised when reducing the PWM duty cycle value to levels that cause the motor to move too slowly, or not at all. Low speed operation is not exactly the same as high speed. A PWM duty cycle value of 99% will run the motor at almost full speed, or 100%. This is not the case for low speeds. At 0% duty cycle, the motor will be off. At 1% duty cycle, the motor will be turning on 1% of the time, which will not be enough to get the motor to move. Trying to drive the motor with a PWM duty cycle value that's too low will cause a condition called a stall.

A motor stall occurs when there is not enough current running through the motor windings to create movement. This stall condition can cause unwanted heat in the motor, motor drive IC, or associated wiring if allowed to continue too long. A stall can often be identified by a high-pitch whine coming from a motor when it should be moving. This is an indicator that the PWM duty cycle that you're sending the motor is too low.

The minimum PWM duty cycle value that can be used to drive a motor will be based on many variables:

- Construction of the motor – Variables from one motor to another can affect the lowest speed that a motor can run at without stalling.
- Gear ratio of the gearbox, if one is present – A gearbox will generally lessen the load on the motor, allowing the motor to turn at higher speeds, while the wheel/tire turns more slowly.
- Size of the wheel/tire combination attached to the motor – Larger diameter wheel/tire combinations will be more difficult for the motor to rotate. A larger the wheel/tire combination will cause the motor to stall more easily.
- Voltage being supplied to the motor drive IC – It's easier to stall a motor when the motor drive IC is being run by a lower voltage supply. A motor drive IC running on 5-volts will cause motors to stall easier than one running at 9-volts or more.

- Resistance of the driving surface – A smooth driving surface like hardwood or concrete will allow for lower PWM values before stalling. Surfaces like carpet will cause more resistance for the robot to drive over and will require a higher PWM value to avoid stalling.
- Weight of the robot – A light robot will be easier for the motors to propel, so lower PWM values can be used. As the robot increases in weight, driving the motors will require more force, so a higher PWM value will be required.

With so many factors involved, the only way to figure out the minimum PWM duty cycle that can be used on your robot before stalling the motors is experimentation. Every robot and driving surface will be a little different.

Left and Right Motor Variability

In a perfect world, when commanded to drive straight, your robot would drive in a perfectly straight line. As we found in the last section, many variables will come into play when determining exactly how fast a motor will move. This is especially problematic when the enable lines of the motor drive IC are not controlled using PWM.

Without PWM control, each motor will run as fast as possible when commanded. This could result in one motor running at 195 revolutions-per-minute (RPM), and the other motor running at 205 RPM. This may not sound like much, but that means that wheel will rotate 10 more times than the other over the course of a minute. Breaking that down even further, one of the wheels will make one extra revolution every 6 seconds. Even over short distances, this will result in a slow turn when the robot is supposed to be driving straight.

PWM control of the left and right enable lines on the motor drive IC will allow you to “calibrate” the left and right motor speeds independently. If you command both motors to drive forward at a PWM value of 95% and the robot is turning slightly left, that means the right motor is driving a little faster than the left when using 95% PWM values. Slightly decreasing the PWM value of the right motor will slow down that motor and should cause the robot to drive more accurately in a straight line. Reducing the right motor PWM value too much could cause the robot to start turning to the right. Finding the right value might take a couple of attempts, but the driving accuracy that’s gained will be well worth the effort.