EE3422- Embedded and Realtime Systems
LAB-3

Section 1: How to drive a motor

This section will provide an overview of how to drive and control the speed of DC motor using Arduino and L298N motor driving IC. As discussed in earlier labs, Arduino Uno comes with the six Pulse Width Modulation (PWM) pins (3, 5, 6, 9, 10 and 11) that are used to control the motor rotation in the forward and backward direction. PWM signal from Arduino is provided to L298N IC, which than provides current to motors depending on the external applied voltage (4.5V to 36V).

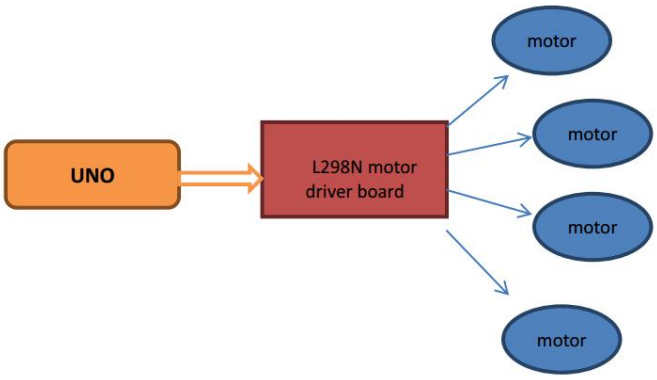
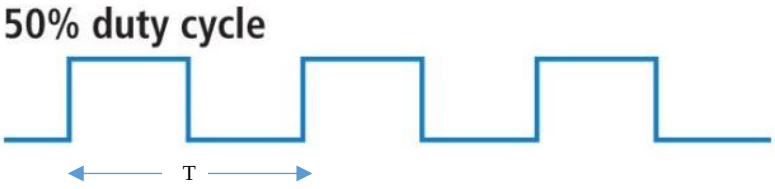


Figure 1: Motor connection overview

What is PWM signal?

Pulse Width Modulation or PWM signal is a technique of using digital means to get the analog signal. As the Arduino is a digital controller that gives only 0 or 5 volts at the output pins. However, using ON and OFF switching technique, a square wave can be generated that has 50% ON state and 50% OFF state on a constant time period as shown below.



This ON-OFF pattern allows to generate the output voltages between 0 to 5 volts. Like in case of 50% duty cycle the average output voltage is 2.5 volts. The duration of "ON time" is called as Pulse Width. To get varying analog voltage values, you have to change, or modulate the pulse width.

If you repeat this ON-OFF pattern fast enough using an LED for example, the result is as if the signal is a steady voltage between 0 and 5V controlling the brightness of the LED.

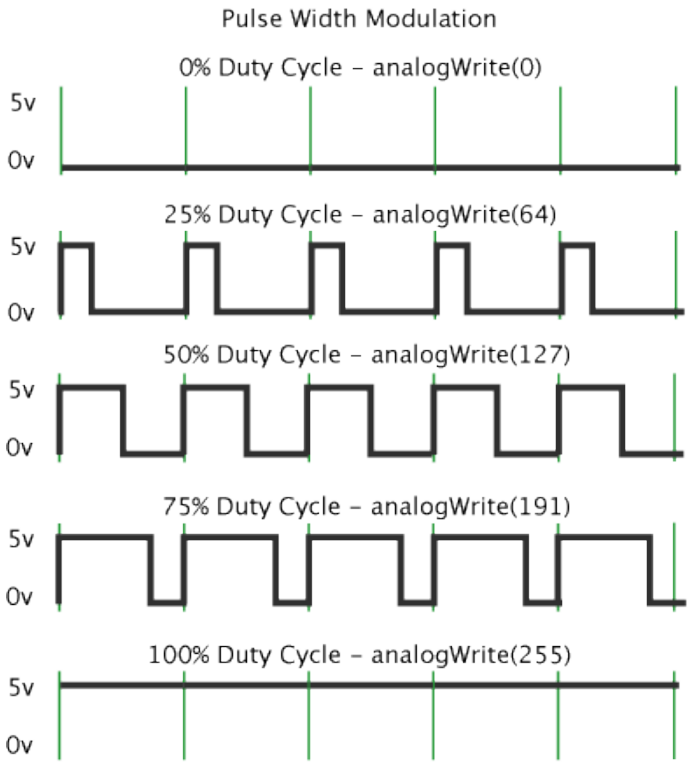


Figure 2: Different duty cycles in PWM signal

Figure 2, shows the different pulse width durations and commands used to generate varying duty cycle. The green lines show the regular time period. For example, if the PWM has 500HZ frequency, then the time period between green lines is 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.

What is H-Bridge (L298N)?

In this tutorial, you will learn how to use L298N H-bridge Dual Motor Controller Module 2A with Arduino. This allows you to control the speed and direction of **two** DC motors, or control **one** bipolar stepper motor with ease. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC. In general, an H-bridge is a simple circuit, containing four switching elements, with the load at the center, in an H-like configuration:

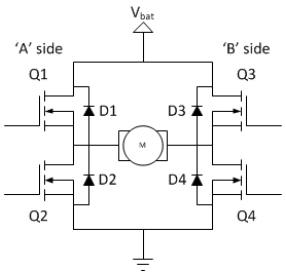
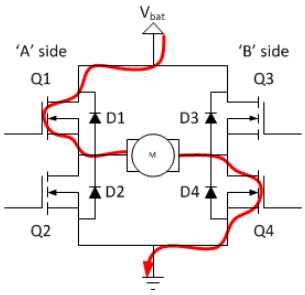


Figure 3: H-bridge schematic

The switching elements (Q1 .. Q4) are usually bi-polar or FET transistors, in high-voltage applications IGBTs. Integrated solutions also exist but whether the switching elements are integrated with their control circuits or not is not relevant for the most part for this discussion. The diodes (D1 .. D4) are called catch diodes and are usually of a Schottky type. The top-end of the bridge is connected to a power supply (battery for example) and the bottom-end is connected to ground.

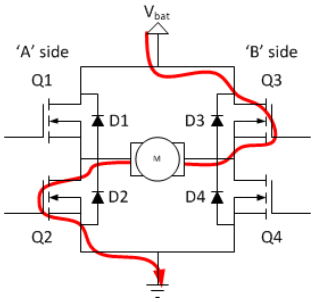
Forward Motion

The basic operating mode of an H-bridge is simple: if Q1 and Q4 are turned on, the left lead of the motor will be connected to the power supply, while the right lead is connected to ground. Current starts flowing through the motor which energizes the motor in (let's say) the forward direction and the motor shaft starts spinning.



Backward Motion

If Q2 and Q3 are turned on, the reverse will happen, the motor gets energized in the reverse direction, and the shaft will start spinning backwards.



Caution: *In a bridge, you should never ever close both Q1 and Q2 (or Q3 and Q4) at the same time. If you did that, you just have created a low-resistance path between power and GND, effectively short-circuiting your power supply. This condition is called 'shoot-through' and is an almost guaranteed way to quickly destroy your bridge, or something else in your circuit.*

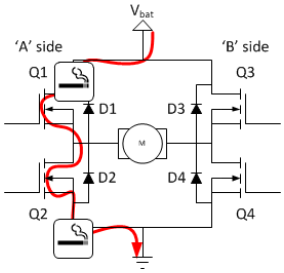


Figure.4: Shoot-through in H-bridge

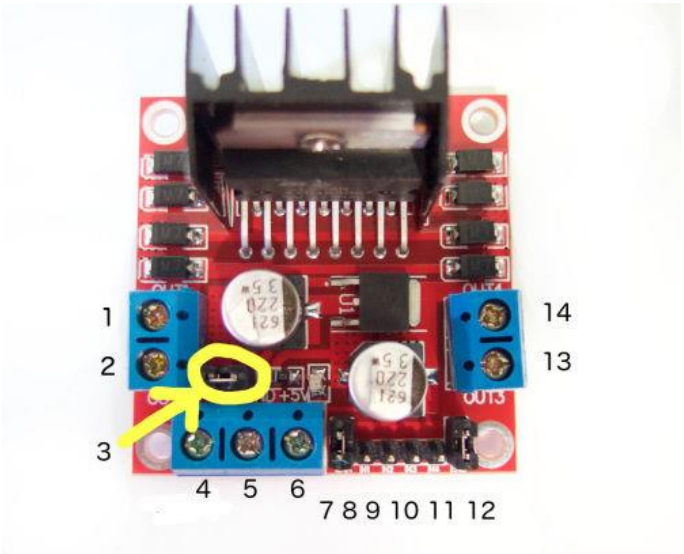
How to Connect L298N with Arduino?

Specifications:

- Double H Bridge Drive Chip: L298N
- Logical voltage: 5V Drive voltage: 5V-35V
- Logical current: 0-36mA Drive current: 2A (MAX single bridge)
- Max power: 25W

L298N Pins:

1. DC motor 1 “+” or stepper motor A+
2. DC motor 1 “-” or stepper motor A-
3. 12V jumper – remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control
13. DC motor 2 “+” or stepper motor B+
14. DC motor 2 “-” or stepper motor B-



Two things to mention;

- Make sure you have all your grounds tied together; Arduino, Power source, and the Motor controller.
- The PWM Pins are unnecessary if you do not want to control the speed of the motors and want a fix speed.

Section 2: Tasks to Perform

2.1 Example – LED Fading using PWM

In this example, you will learn how to use PWM to control the brightness of LED. Connect a resistor 330Ω and LED in series with the PWM Pin 9 of the Arduino Uno. After the connection please download the Sketch (**FadingLED.ino**) from the Moodle in Lab-3 folder. Go through the program and run the program and observe how it works.

Write your observations that how code is working in the area below.

2.2 Example – Driving a Motor using H-bridge

In this example, we will learn how to drive a motor. You need to use the robot base and only one set of tyres will move from the code given. As shown in the Figure 5 below, Motor 1 and Motor 2 are attached with the OUT1, OUT2 and OUT3, OUT4 of the L298N IC respectively.

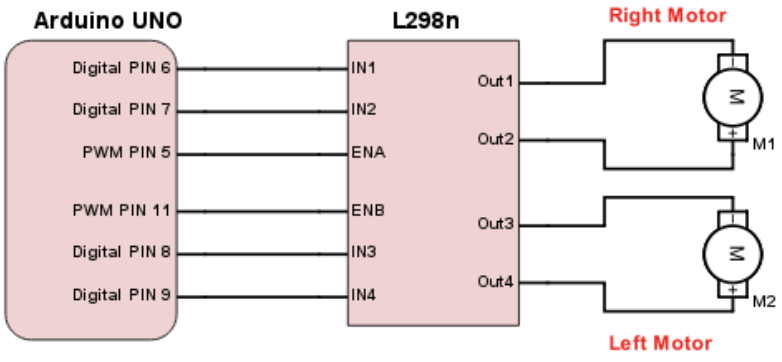


Figure 5: Connection from Arduino to L298N and Motors

IN1 and IN2 of L298N receive the direction of the motor rotation from Arduino Uno digital PINs 6 (high) and 7 (low) or for other direction PIN 7 (high) and PIN 8 (low). While the PWM PIN 5 of Arduino which is connected to ENA of L298N is used to control the speed of Motor 1.

- int speedPinA = 5;
  Variable name assignment to Pin 5
- analogWrite(speedPinA, 255);
  Value of speedPinA=255 that correspond to maximum voltage across motor which is 12V. The value can be changed from 0-255 depending upon the required speed.

In this example, you don't need to do any connections rather used the main robotic kit and upload the provided sketch **(Motor_Rotation_Example.ino)** from the Moodle in Lab-3 folder. Go through the program and run the program and observe how it works.

Write your observations that how code is working in the area below.

Section 3: Solve the Challenges

You need to write program and build circuit for the given challenges and show the result to Lab Demonstrators and get signed your Lab report.

NOTE: You can perform subsection 3.1 and 3.2 in one program.

3.1 Forward and Backward motion with variable speed

In this task, you have to write a code that moves the robotic car forward and backward direction in a given sequence,

Time	Forward	Backward
0-2 sec	Slow Speed	
2-5 sec	Medium speed	
5-7 sec	High Speed	
7-10 sec	Stop	Stop
10-12 sec		High Speed
12-14 sec		Medium speed
14-16 sec		Slow speed

Slow speed= 35% duty cycle (analogWrite(speedPinA, 90);)

Medium speed= 65% duty cycle (analogWrite(speedPinA, 167);)

High speed= 100% duty cycle (analogWrite(speedPinA, 255);)

3.2 Left and Right Turn

In addition to above task, write five functions in your sketch. Functions should have name shown below,

```
void forward(fspeed)
{
   // Left motors will have same speed as variable 'fspeed'
}
void backward(bspeed)
{
  // Left motors will have same speed as variable 'bspeed'
}
void leftmove(lspeed)
{
  // Left motors will move with 'lspeed' variable and right motors will be stoped
}
void rightmove(rspeed)
{
  // Right motors will move with 'rspeed' variable and left motors will be stoped
}
void stopall()
{
  // pass the zero PWM value to all motors
}
```

Call these functions in your void loop () in below sequence,

Time	Forward	Backward	Left	Right
0-2 sec	Medium Speed			
2-4 sec			Turn Left	
4-6 sec				Turn Right
7-10 sec	Stop	Stop	Stop	Stop
10-12 sec		High Speed		
12-14 sec		Medium speed		
14-16 sec		Slow speed		

LAB-3 END