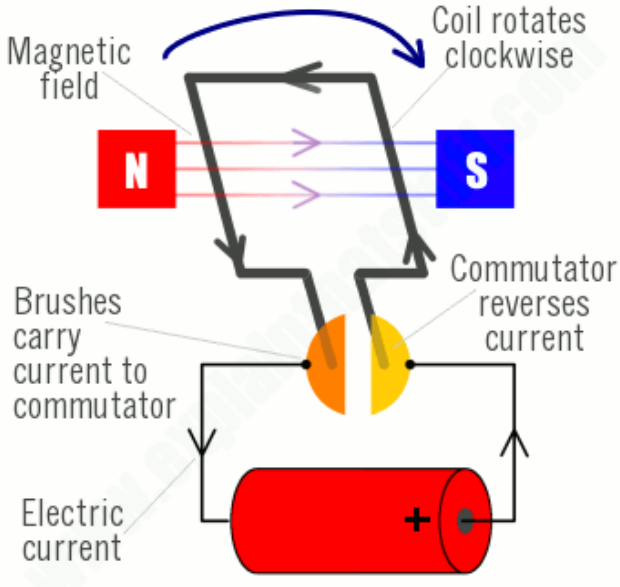
Arduino Controlled Robot - Explanations

Control a small robot with an Arduino and L298N motor driver

For this project, we will be building a small robot to demonstrate how to control motors as well as creating a moving platform for putting sensors and peripherals in the near future. For the next few meetings, we will be primarily focused on how motors work, how the L298N driver works, and how the Arduino controls them.

How does a motor work?

A motor will rotate its shaft when electricity is applied to the terminals. The direction of the rotation will change as the polarity of the voltage changes. The reason is when electric current travels through a wire, it creates a magnetic field around it. If this piece of wire is coiled up, the strength of the magnetic field significantly increases. A magnetic field can attract another magnetic field. Therefore, when a permanent magnet is placed near this coil, it responds by either attracting or repelling the permanent magnet, depending on the polarity. When the coil is placed on a shaft, this is what allows the motor to rotate. The speed that the shaft rotates can be affected by anything that affects the strength of the magnetic interaction. In other words, the power applied, strength of the permanent magnet, and number of coils can directly affect the speed.



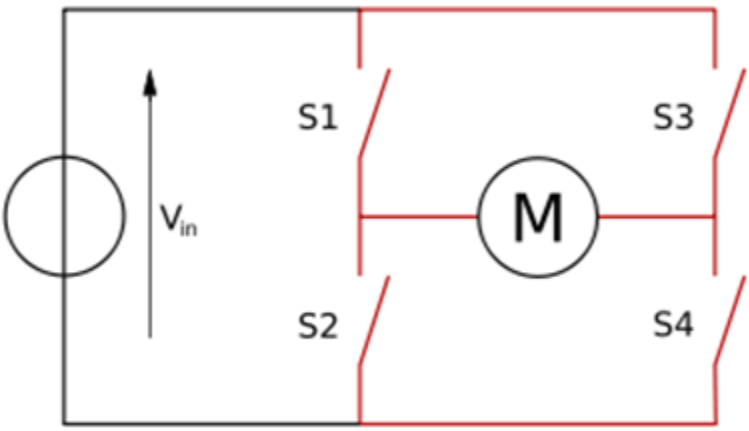
How does the L298N motor driver work?

Last week, we learned about the Arduino microcontroller and that it can control many things such as LEDs, lamps, motors, appliances, etc. However, the Arduino itself may not always have the ability to interact with components draw more than 5 volts at 40mA. Components that draw more than such ratings can damage the Arduino if not properly handled with additional circuitry.

Since we want to control motors in both directions, we need a circuit that can:

- Switch large amounts of current and voltage (more than what Arduino can handle)
- Flip the polarity of voltage given (to drive motors in both directions)
- Allow motor speed to be controlled

The L298N motor driver is an affordable Dual H-Bridge 2A motor driver that is ideal for our small robot. To better understand why, lets discuss what we mean by *Dual H-Bridge 2A*.



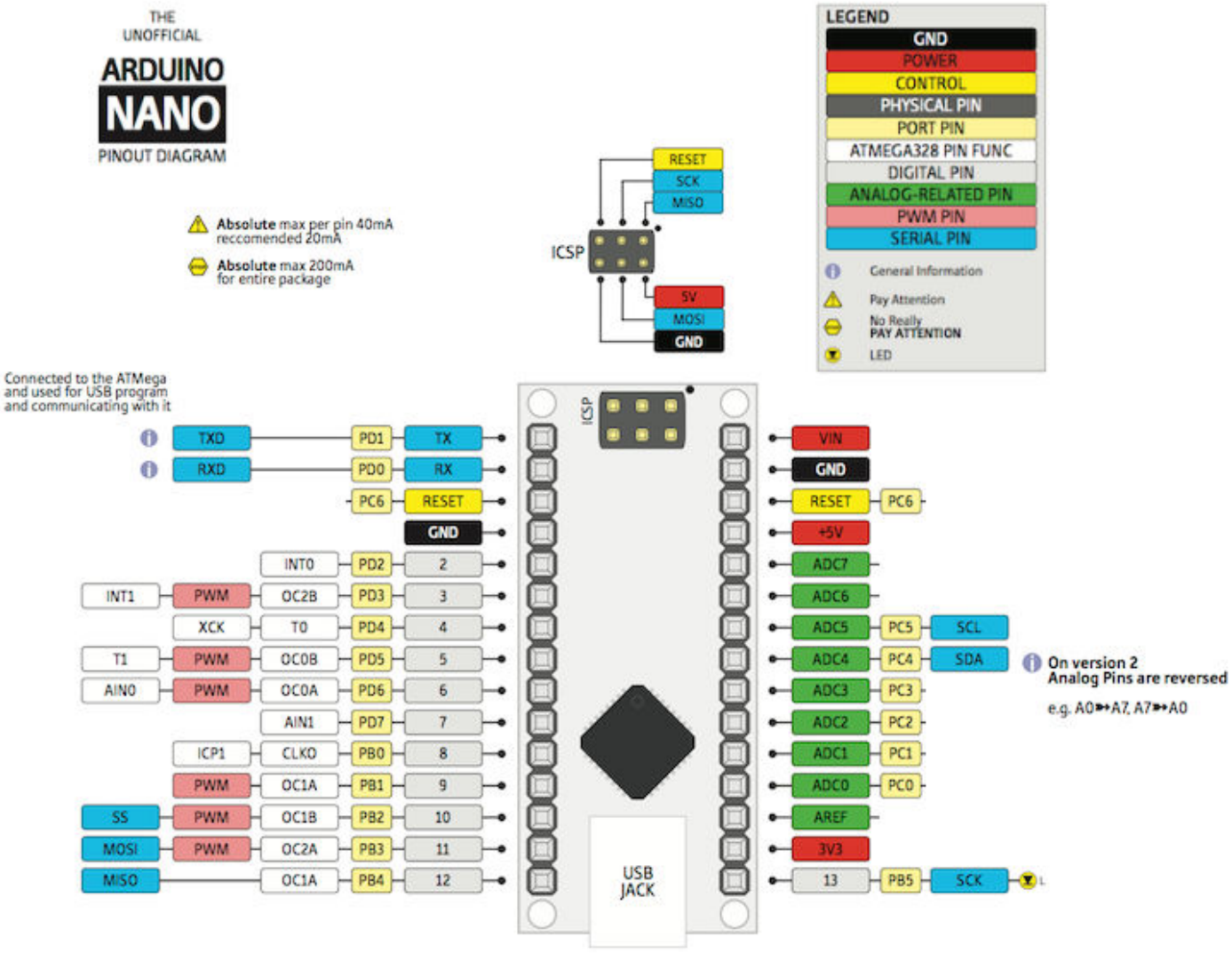
An H-Bridge is an electronic circuit that enables a voltage to be applied across a load in either direction. It is named for how it looks in a circuit. Most H-Bridge circuits involve 4 MOSFETs per load to be controlled. In our case, we have a Dual H-Bridge, meaning that we would theoretically have 8 MOSFETs on our board. Additionally, the title says 2A which specifies that the board will be able to safely handle up to 2 amps of current on the load. The datasheet specifies that the board can handle up to 46V operating voltage, but we will only be using 6V or less for our needs.

The L298N has 8 MOSFETs wired internally. However, it appears as a nice compact chip that is attached to the L298N board. The other components are primarily regulators and resistors for protection.

PURDUE Mechatronics

Arduino Nano pin connections

Last week, we learned about using Digital Inputs/Outputs and Analog Inputs/Outputs with the Arduino. Now we are really going to learn about the microcontroller by being familiar with the Arduino pins so that we understand which pins can be assigned to control the L298N motor driver.



Controlling L298N motor driver with an Arduino

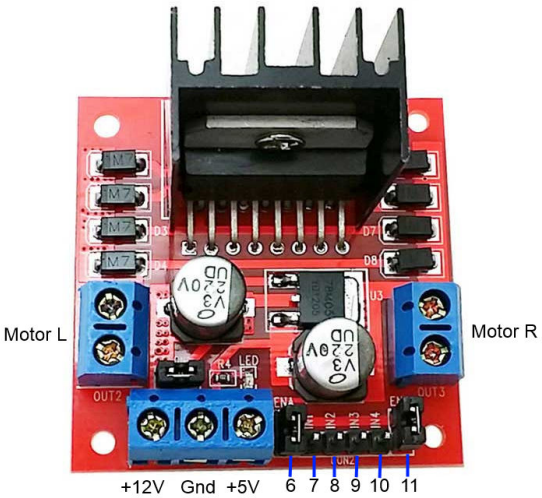
The L298N motor driver requires a total of 4 digital pins (2 pins specify the direction of each motor) and 2 analog PWM pins (1 pin specifies the speed of each motor). To make the process of choosing pins easier for everyone, we have decided to choose digital pins 6-11 on the Arduino to control the L298N motor driver. Even though other pins could have been used, we feel this is the best consecutive pinout for connecting to the Arduino.

If you notice on the Arduino pinouts, pins 6 and 11 are both PWM pins. Almost every pin on Arduino (except VCC/RST/GND) can be assigned as a digital input or output pin.

1. Connect pins labeled **6, 7, 8, 9, 10, 11** from the L298N motor driver to the Arduino Nano. On the motor driver board, these pins are respectively labeled as ENA, IN1, IN2, IN3, IN4, ENB.

2. Connect the terminal labeled **+5V** from the L298N motor driver to the Arduino Nano. Connect the terminal labeled **Gnd** to the same terminal on the Arduino Nano.

3. Connect the terminals on Motor L and motor R to the respective motors on the robot chassis. Remember to be consistent with the polarity as it may affect the direction of the robot. Finally, connect the **+12V** terminal and **Gnd** terminal to the respective terminals on the battery pack (red goes to **+12V**, black goes to **Gnd**).



Writing Arduino code to test the robot

After you have correctly wired the motor driver you can now program the Arduino and use it to run the robot. You can configure the robot to move in any way you want it to. In the near future, we will discuss about different sensors that can be placed on the robot so that it can be configured to avoid walls, receive signals from a smartphone, follow a line, detect light, etc.

For now let's understand how the Arduino can interact with the robot with code. The L298N motor driver works in a very simple way. Each motor is controlled by 3 pins. The left motor is controlled by the left 3 set of pins, and the right motor is controlled by the right 3 set of pins.

The left 3 set of pins are **6, 7, 8** which will control the left motor. Pin 6 controls the speed.
The right 3 set of pins are **9, 10, 11** which will control the right motor. Pin 11 controls the speed.

First, declare all 6 pins to be output in *void setup()* function.

When pin **7** is set HIGH and pin **8** is set LOW, the left motor will go forward.
When pin **9** is set HIGH and pin **10** is set LOW, the right motor will go forward.

When pin **7** is set LOW and pin **8** is set HIGH, the left motor will go backward.
When pin **9** is set LOW and pin **10** is set HIGH, the right motor will go backward.

When ***analogWrite(6, 255)*** is declared, the left motor speed is set to full speed.
When ***analogWrite(11, 255)*** is declared, the right motor speed is set to full speed.

In order for the robot to move, the direction pins must be set **and** the speed pin must be set.

- AnalogWrite value must be set at a high value to move the robot
- Pins **7** and **8** cannot both be HIGH or LOW at the same time
- Pins **9** and **10** cannot both be HIGH or LOW at the same time

Practice Exercise: Make the robot move forward at 50% speed and turn right every 2 seconds.

For your convenience, the example code has been provided for you and can be accessed at:

www.PurdueMechatronics.com/resources