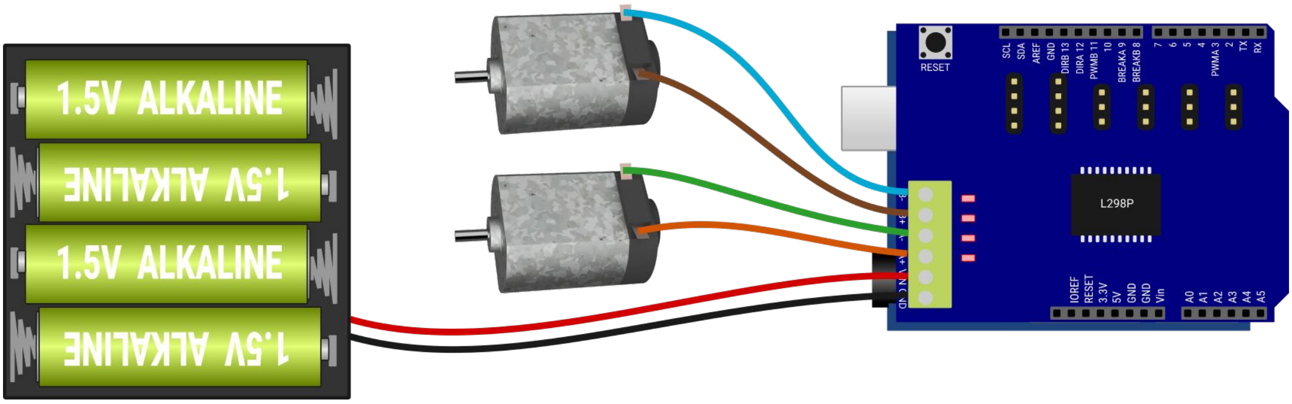
# Arduino-project 008 – Control motors with the motor shield

Jun 1, 2018 | Example code, Guides | 0 comments

A robot needs to be able to move and for that you need motors. But motors require alot of current so you can't connect them directly to a Arduino board. Therefore you need some sort of motor driver. One of the easiest to use is the Arduino Motor Shield R3. As this is a so called  shield it can be mounted directly on top of (or under) a standard Arduino board and you don't need to connect a lot of wires on a bread board. The chip on this card is a L298P which can drive two DC motors of up to 2A each which is enough for most smaller robots. To control the the motors with this card you use three signals per motor called PWM, DIR and BREAK. PWM is used for setting the speed of the motor, DIR controls the direction and BREAK short circuits the motor (safely) which acts as a breaking mechanism. As the name implies the speed is controlled with the pulse-width modulation, i.e. you use the analogWrite() function.

## The circuit

Mount the motor shield on top of the Arduino board and connect the motors and batteries (or other power source) for the motors according to the diagram below. The batteries must have the same total voltage that the motors are made for (but no more than 12 V). The Arduino board can take power directly from the batteries, but for safer operation it is better to power the Arduino separately. Otherwise there is the risk of the voltage dropping when the motors draw a lot of current causing the Arduino to reboot. This circuit and the code below assumes that the robot is using so called tank drive where the wheels on each side of the robot are driven separately and the robot turns by changing the speed/direction of the motors. This is by far the most common way for controlling smaller robots and this project is suitable for most of our robot chasis. For robots that have four motors (two on each side) you connect both motors on each side together to the same outputs on the motor shield.



## Description of the code

The code is simple and doesn't really do anything useful, but works well for testing the circuit and forms a good ground for programming your own robots. First we define which connections on the Arduino board are used to what. These are of course determined by the motor sield and you shouldn't change them, but we add them as constants so that in the rest of the code we can use the names instead of having to type numbers. It's much easier to remember `dir_a` thn tht the function is

on pin 12. The motor shield has a few additional connections, but these are not necessary for basic usage.
After this we create a couple of functions for setting the speed of the motors and to run them forward, backward and for turning and stopping. The stop function is strictly not required as the only thing it does is setting the speed to 0, but it does make the rest of the code easier to read and understand.
In setup()we specify which connections on the Arduino that should be outputs, i.e. th four control signals that the motor shielduses. In `loop()` we find the main algorithm. This is very simple and uses the functions we created earlier for driving and turning in a simple sequence. This is of course just a basic example to make sure everything is working. You will replace this code with something more exciting, e.g. letting the robot accept commands via a bluetooth module, or you could connect an ultra-sonic range sensor so that the robot can drive around and avoid obstacles on its own.

## If the robot runs the wrong way

If the robot doesn't move quite the way you expect it to do there are several possible reasons. The direction a DC motor rotates depends on which way the current through it flows. So if a motor rotates the wrong way it is enough to switch around the two wires to it. I.e. if the robot rotates left when it should go forward then switch the wires on the left motor; and vice verse, switch the wires to the right motor if it rotates right. If the robot moves backwards when it should go forward you switch the wires for both motors. Each motor should still be connected to the same pair of contacts, i.e. one should use A+ and A1 and the other B+ and B-.
If the robot moves forward and backward as it should, but turns left when it should turn right and vice versa it means that the motors are on the wrong side of the robot. You would think that it would be enough to switch places of the motors, but if you do that the robot will go backwards when it should go forward. Instead you switch the wires to the motors around so that the motor connected to A+ and A- is instead connected to B+ and B- (without changing places of + and -), and the motor connected to B+ ans B- is moved to A+ and B- (again keeping track of + and -).
If your robot has four motors, two on each side, and one motor on one side rotates the wrong way you switch around the wires to that motor. You of course also hav to make sure that both motors on one side are connected to the same connectors, e.g. both left motors connected to A+ and A- and both right motors connected to B+ and B-.