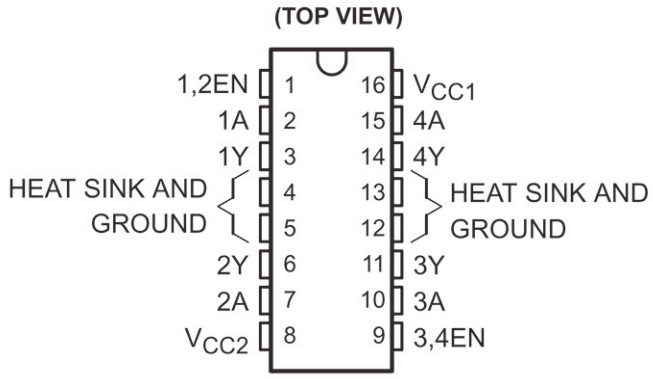
DC motor control with a SN754410 motor driver IC and the Arduino board
Prof. Fabian Winkler

The SN754410 is a handy IC that allows you to control the speed and direction of a DC motor with only one PWM output and two digital outputs from your Arduino board.

SN754410
QUADRUPLE HALF-H DRIVER



FUNCTION TABLE
(each driver)

INPUTS†		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low-level
X = irrelevant
Z = high-impedance (off)
† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

For more information on this part read pp.255 – 260 in O'Sullivan/Igoe: *Physical Computing* for how to use the SN754410 motor driver IC with a microcontroller.

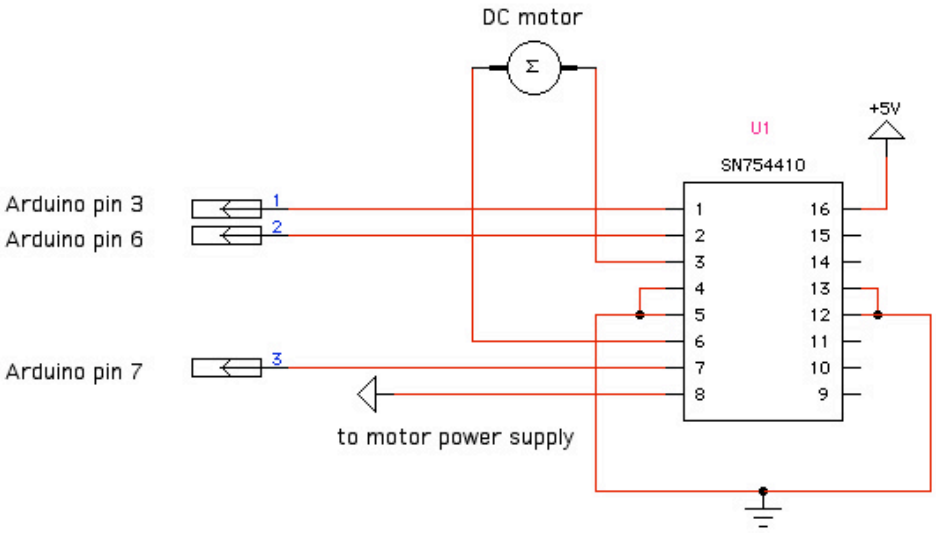
In this example, we set up a simple code that controls the direction and speed of a DC motor.

Here is the Arduino code:

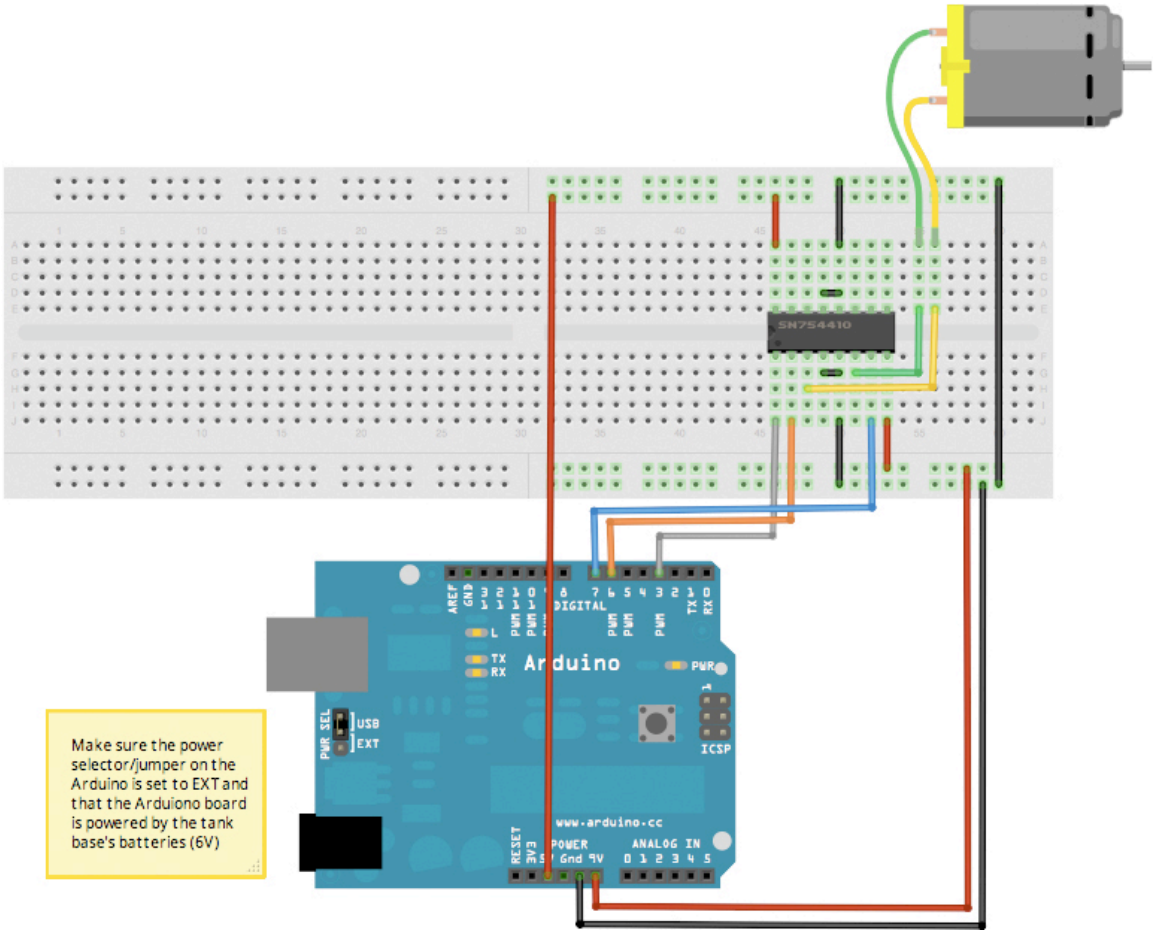
```
/*
 * Arduino code for SN754410 H-bridge
 * motor driver control.
 * copyleft Feb. 2010, Fabian Winkler
 *
 */

int speedPin = 3;       // H-bridge enable pin for speed control
int motor1APin = 6;     // H-bridge leg 1
int motor2APin = 7;     // H-bridge leg 2
int ledPin = 13;        // status LED
int speed_value_motor1; // value for motor speed

void setup() {

  // set digital i/o pins as outputs:
  pinMode(speedPin, OUTPUT);
  pinMode(motor1APin, OUTPUT);
  pinMode(motor2APin, OUTPUT);
  pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  digitalWrite(ledPin, HIGH);  // status LED is always on

     // put motor in forward motion
    digitalWrite(motor1APin, LOW);   // set leg 1 of the H-bridge low
    digitalWrite(motor2APin, HIGH);  // set leg 2 of the H-bridge high

    // just invert the above values for reverse motion,
    // i.e. motor1APin = HIGH and motor2APin = LOW


    // control the speed 0- 255
    speed_value_motor1 = 127; // half speed
    analogWrite(speedPin, speed_value_motor1);  // output speed as
                                                // PWM value

}
```

The following is a circuit diagram for the project



And this is one possibility how it can look like on your breadboard (Fritzing screenshot):



Regular DC Motors vs. Gearbox Motors

In most of your application when a rotary movement is necessary you will need force (torque) over speed. In this case, use a gearbox motor instead of a regular DC motor. The gearbox attached to a motors output shaft amplifies its torque and slows down its speed. If you are using a regular DC motor and adjust its speed with the Arduino's PWM output you will slow down its speed AND reduce its torque!

A good source for inexpensive gear motors is Solarbotics.com with their *Gear Motor 2 – 1:224 offset shaft* (see picture on previous page).

You can also consider hacking a servo motor for an inexpensive continuously rotating gear motor (see: http://www.seattlerobotics.org/guide/servohack.html) rather than purchasing a regular gearbox motor. In most cases this technique will save you half the cost.

We will continue with the next workshop that uses the Arduino board, the SN754410 motor driver IC and the motors and the base of a hacked RC tank as a starting point for a caterpillar track – based robot.