**Literature review**

**Development Boards**

**Development board** is basically a printed circuit **board** with circuitry and hardware on-**board** to facilitate experimentation with certain microcontrollers. These **boards** can save you from a lot of repetitive tasks.

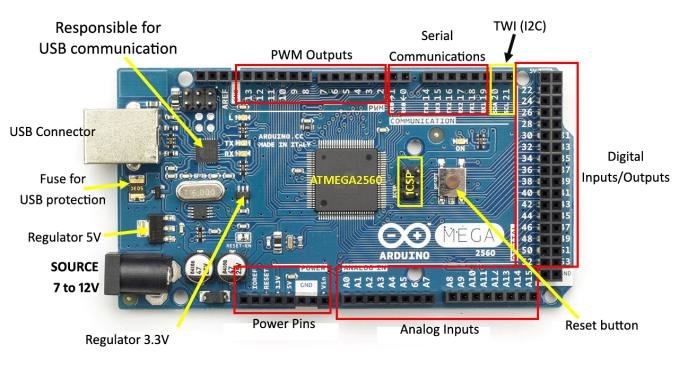
1. Arduino Development **Board**: ...
2. Raspberry Pi Development **Board**: ...
3. The BeagleBone Black Development **Board**: ...
4. The Intel Galileo Development **Board**: ...
5. The pcDuino Development **Board**:
6. The Uruk Development **Board**: ...
7. The Goldilocks Development **Board**: ...
8. The ExtraCore Development **Board**:
9. DIY Printed Circuit (PCB) Boards:

Arduino Development Board:

Arduino is the popular open-source electronics prototyping stage focused around simple to-utilize equipment and software. It's proposed for specialists, designers, and anybody intrigued by making intelligent articles or situations and is intended to be as adaptable as would be prudent to fit your venture's necessities.  The Arduino Development Board is a good example of top development boards for DIY projects. Arduino is a company that deals with open source computer hard ware and software. The company designs and does manufacturing of kits creating digital devices and many interactive objects which have the capability of sensing and making good control of the whole physical world. Since it is an example of a microcontroller, it works swiftly and steadily. It is manufactured primarily by Smart Projects in Italy and many other countries and vendors.

# **ARDUINO MEGA 2560 REV3**

The MEGA 2560 is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.

****

**Controlling of servo actuation**

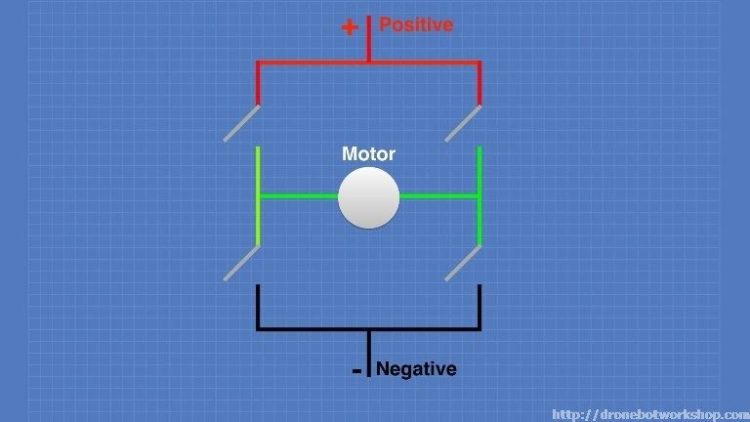
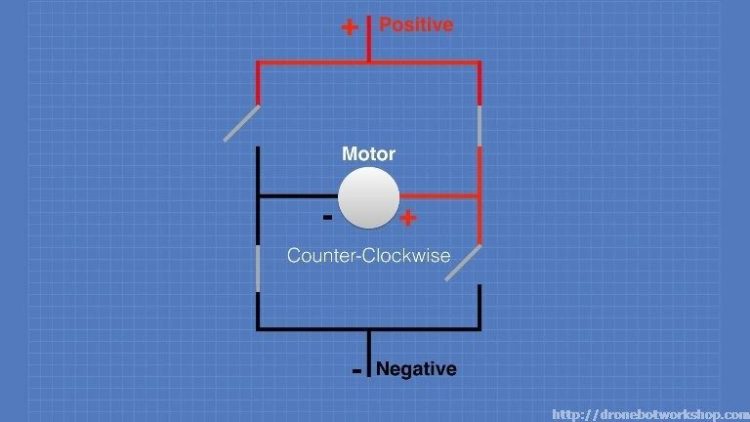
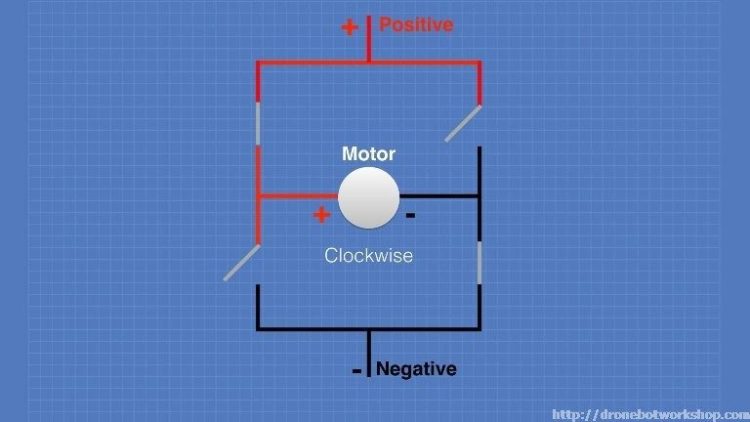
### H-Bridge

Now that you know how DC motors work, how you can reverse their direction by changing polarity and how you can change their speed using pulse width modulation, let’s examine an easy way to do this using a very common circuit configuration called an “H-Bridge”.

An “H-Bridge” is simply an arrangement of switching the polarity of the voltage applied to a DC motor, thus controlling its direction of rotation. To visualize how this all works I’ll use some switches, although in real life an H-Bridge is usually built using transistors. Using transistors also allows you to control the motor speed with PWM, as described above.

In the first diagram we can see four switches which are all in the open or “off” position. In the center of the circuit is a DC motor. If you look at the circuit as it is drawn here you can distinctly see a letter “H”, with the motor attached in the center or “bridge” section – thus the term “H-Bridge”.

If we close (i.e. turn on) two of the switches you can see how the voltage is applied to the motor, causing it to turn clockwise.

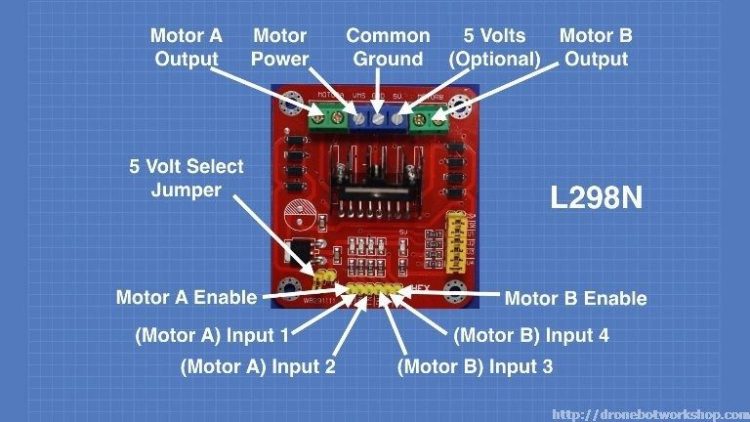
Now we’ll open those switches and close the other two. As you can see this causes the polarity of the voltage applied to the motor to be reversed, resulting in our motor spinning counterclockwise.

This is pretty simple but effective. In fact if all you need to do is design a circuit to drive the motor full-speed in either direction you could actually build this as shown here, using a 4PDT (4 Pole Double-Throw) center-off switch. But of course we want to control the motor using an Arduino, so an electronic circuit where the switches are replaced by transistors is what we need.

### L298N Module Pinouts

You’ll find a few different styles of L298N boards but they all operate in the same fashion. The board contains an L298N mounted on a heatsink, a 5 volt voltage regulator to “optionally) provide power for logic circuits, supporting diodes and capacitors and connectors as follows:

* Logic inputs for each H-Bridge circuit
* Power supply inputs for the motor power supply
* An optional 5 Volt power input for the logic circuits.
* Outputs for each DC motor

A typical L298N Board is shown here.

You’ll notice that the board also has a number of jumpers. Most of the time you will leave them in place, with the exception of one. They are as follows:

* **CSA** – This is the “current sensing” function for Motor A. If the jumper is in this function is ignored. Most of the time you’ll leave this jumper in place.
* **CSB** – The “current sensing” function for Motor B. Again you’ll usually just leave this in place to disable this function.
* **U1** – Input 1 pull-up resistor. You will usually leave this in place, which enables a 10k pull-up resistor for the input.
* **U2** – Input 2 pull-up resistor.
* **U3** – Input 3 pull-up resistor.
* **U4** – Input 4 pull-up resistor.
* **5v-EN** – This is the only jumper that you need to really pay attention to. When this jumper is in place it enables the boards internal 78M05 5 Volt regulator, supplying logic power from the motor power supply. When this jumper is enabled you will NOT supply 5 volts to the 5 Volt input terminal. When the jumper is removed you will need to supply 5 Volts to the 5 Volt input terminal.

If you do use the internal voltage regulator you’ll have to supply the motor power supply with at least 7.5 volts.

Speaking of the motor power supply it needs to be a bit higher voltage than the actual motor requirements. This is due to the internal voltage drop in the transistors that form the H-Bridge circuit. The combined voltage drop is 1.4 volts, so if you are using 6 Volt motors you’ll need to give the board 7.4 volts, if you have 12 volt motors then your motor supply voltage will need to be 13.4 volts.

The board has four input terminals plus two enable terminals. You will use these terminals to control both direction and speed or each motor. They are as follows:

* **IN1** – Input 1 for Motor A
* **IN2** – Input 2 for Motor A
* **IN3** – Input 3 for Motor B
* **IN4** – Input 4 for Motor B
* **EN1** – Enable line for Motor A
* **EN2** – Enable Line for Motor B

In order to simplify things a bit I’ll just discuss the inputs and enable for Motor A, Motor B functions identically.

The two Input lines control the direction that the motor rotates. I will call one direction “forward” and the other one “reverse”, if it makes more sense to you just substitute “clockwise” and “counterclockwise”.

You control motor direction by applying either a Logic 1 (5 Volts) or Logic 0 (Ground) to the inputs. This chart illustrates how this is done.

|  |  |  |
| --- | --- | --- |
| **INPUT 1** | **INPUT 2** | **DIRECTION** |
| Ground (0) | Ground (0) | Motor Off |
| 5 Volts (1) | Ground (0) | Forward |
| Ground (0) | 5 Volts (1) | Reverse |
| 5 Volts (1) | 5 Volts (1) | Not Used |

As you can see only two combinations are actually used to control the direction of the motors rotation.

The Enable line can be used to turn the motor on, to turn it off and to control its speed. When the Enable line is at 5 Volts (1) the motor will be on. Grounding the Enable line (0) will turn the motor off.

To control the speed of the motor you apply a Pulse Width Modulation (PWM) signal to the Enable line. The shorter the pulse width, the slower the motor will spin.

1. **MD10C**

**1.0 Introduction:**

[Enhanced 10Amp DC Motor Driver (MD10C)](http://cytron.com.my/p-md10c)is an enhanced version of the MD10B which is designed to drive high current brushed DC motor up to 13A continuously (for Rev2.0). It offers several enhancements over the MD10B such as support for both locked anti-phase and sign-magnitude PWM signal as well as using full solid state components which result in faster response time and eliminate the wear and tear of the mechanical relay.

Arduino main boards such as [Arduino Uno](http://cytron.com.my/p-arduino-uno) are quite popular these days due to its easy-to-use programming environment. Therefore, by interfacing Arduino main board with MD10C, we gain easier control of DC motors. In addition, further adding an Arduino LCD Keypad Shield can help us to control the DC motor that connected to MD10C with the 6 momentary push buttons (built-in push buttons on LCD keypad shield including 1 Reset button) and also display some useful message or information.

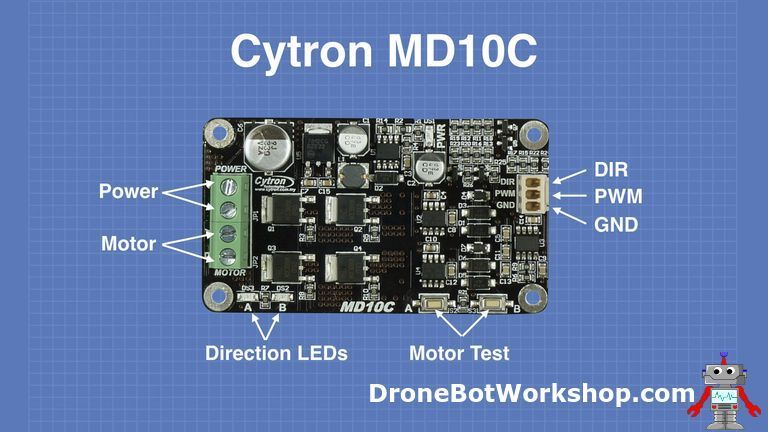
In this tutorial, I am going to show several simple ways to drive DC brush motor using MD10C and Arduino.

**How does MD10C work?**

One of the features of MD10C is it supports both sign-magnitude and locked anti-phase PWM signal, means you can control motor in 2 different ways!

* **Sign-magnitude mode**You require **2 separate signals** to control the motor, one is for direction (counterclockwise or clockwise) and another is for the speed. To control motor direction, DIR pin is connected to HIGH or LOW for different direction, whereas PWM pin is fed with PWM signal to control the motor speed.
* **Locked anti-phase mode**In this mode, only **1 signal** is required to control both speed and direction of motor. PWM pin of MD10C is always connected to HIGH (5V), while DIR pin is fed with d to PWM signal. The direction of motor depends on whether the duty cycle of PWM signal is less than or more than 50%. The motor will run in one direction if the duty cycle is less than 50% and another direction if more than 50%. The motor stops if duty cycle is 50% (approximately). The speed depends on the percentage of duty cycle.

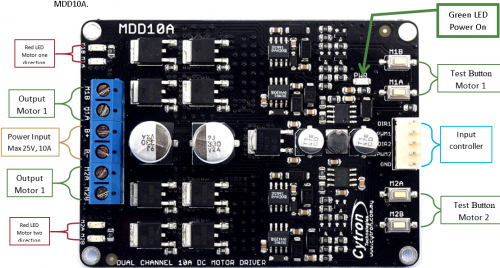
**Sign-magnitude VS Locked anti-phase**

****Main advantage of using locked anti-phase mode compare to sign-magnitude mode in motor controlling is it reduces number of I/O pins used since it uses only 1 signal for controlling both the speed and direction of motor. However, if locked anti-phase is used, in the beginning motor will run at maximum speed (as you can see in the video) when there is no PWM signal. Users will have to provide PWM signal at 50% duty cycle to DIR pin or set PWM pin (MD10C) to LOW state in the beginning. In terms of programming, the algorithm will be more complex because in locked anti-phase mode, motor speed changes with increment or decrement of PWM value depends on the direction which the motor is at. Motor becomes slower with increment of duty cycle if duty cycle of PWM signal is less than 50% or otherwise if duty cycle is more than 50%.

1. **Md10a**

**Introduction**

[MDD10A](http://cytron.com.my/p-md10c) stands for Motor Driver Dual Channel 10(Amps). This DRIVER is the dual channel version of [MD10C](http://cytron.com.my/p-md10c) which is designed to drive 2 brushed DC motors at high current s up to 10A each, continuously. Just like [MD10C](http://cytron.com.my/p-md10c), [MDD10A](http://cytron.com.my/p-md10c) also supports locked-antiphase and sign-magnitude PWM control. It uses all solid state components resulting in faster response times and eliminates the wear and tear of mechanical relays.

****