Driving Motors using Arduino

Dinesh Sharma
D. Chakraborty, K. Chatterjee, B.G. Fernandes,
J. John, P.C. Pandey, N.S. Shiradkar, K.R. Tuckley

EE department, IIT Bombay

January 28, 2024

1 Drive Requirements

For most applications using Battery Operated DC motors (BO motors) we need to be able to control the rotation speed as well as the direction of rotation of the motor shaft.

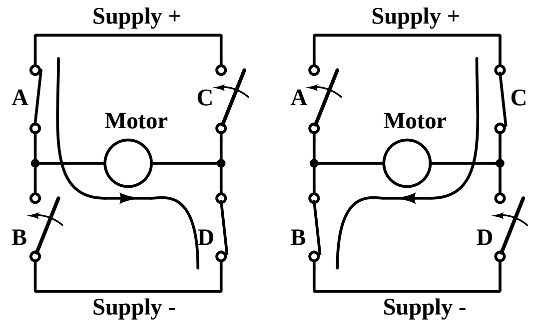
- A Battery Operated motor typically requires hundreds of mA of current. This is much higher than what the port pins of most micro-controllers can drive directly. For example the absolute limit for current drawn from an Atmega 328P micro-controller pin (used in Arduino cards) is 40 mA.
- Also, to generate sufficient torque, BO motors often need to operate at voltages higher than 5 V (which is used as the operating voltage for a large number of digital circuits and micro-controllers). Port pins provide digital outputs limited by the supply voltage of the micro-controller.
- Motor windings are inductive and will generate large voltage spikes when current through them is switched. We need to provide flywheel diodes to prevent these spikes from damaging connected components.
- It is not efficient to use separate batteries to provide different operating voltages required for the motor, the micro-controller card and other circuits in the application. Ideally, all the required voltages should be derived from the same battery.

To meet all these requirements, we need a well-designed interface circuit between the micro-controller and the motor.

2 H bridge

The high current needed for driving the motor is often provided through an “H bridge” circuit.

There are four switches in the H bridge – labelled as A, B, C and D in the figure below. The load (in this case, a motor) forms the horizontal element of the H shaped circuit.



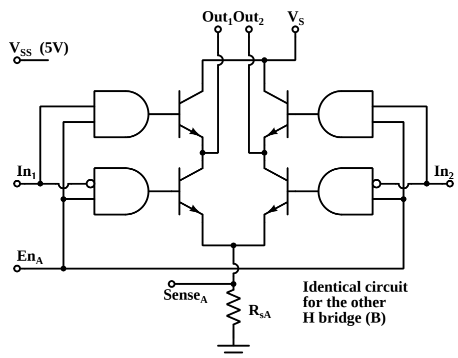
- Current flows from left to right through the motor terminals when switches A and D are closed, while B and C are open.
- Current flows from right to left through the motor terminals when switches A and D are open, while B and C are closed.

Thus we can reverse the direction of rotation of the motor by closing the appropriate pair of switches in the H-bridge. Of course we have to ensure that switches A and B are never closed simultaneously. Similarly, switches C and D must never be closed simultaneously – as this will lead to a short across the power supply.

3 IC LM298 with dual H bridge circuits

The mechanical switches shown in the H bridge circuit discussed above can be replaced by electronic switches in an integrated circuit which can also ensure through in-built logic that two switches in the same arm are never simultaneously closed. LM298 chip from ST micro provides this functionality.

The internal circuit of LM298 IC is shown on the right. The actual chip has two identical H bridges labelled A and B with individual enable inputs En_A and En_B. Only half the circuit (for A side) is shown here. The other half is identical. The load (for example a motor) is connected between Out_1 and Out_2. The output transistors are connected to the digital output transistors of AND gates in a Darlington configuration to provide high current drive.



The resistor to ground is used to monitor the current being driven through the load. If this is not required, the Sense_A pin should be connected to ground. V_{SS} provides the logic supply voltage of 5 V. V_S provides the higher voltage to be applied to the motor and up to 46 V can be used for this purpose. When $\mathrm{En}_A = 0$, outputs of all AND gates is ‘0’ and all npn transistors are OFF. Thus, the outputs are floating and no current is driven through the load.

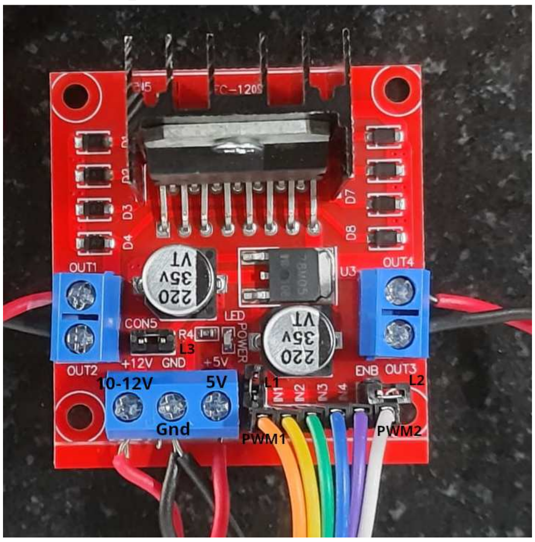
With $\text{En}_A = 1$ and In_1 at logic High, Out_1 will be $\approx V_s - 1.2$ V. (This is because the output is pulled up through the npn emitter which is in a Darlington configuration with the output driver of the AND gate – roughly two diode drops below the base drive which can at max be V_s).

With $\text{En}_A = 1$ and In_1 at logic Low, Out_1 will be ≈ 0.2V. This is the saturation voltage of the lower npn transistor.
Similarly, Out_2 is controlled by In_2.

4 Motor Driver Cards

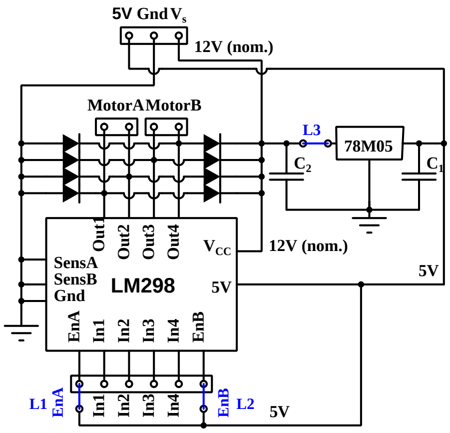
While it is certainly possible to design a special purpose interface circuit using such ICs to drive motors, it is much faster and more convenient to use one of several ready-made cards available for this purpose.

For example, we can use an L298N card which makes use of the LM298 IC described above.



- The L298N motor driver card is widely used for driving BO motors. (The card and the IC have similar sounding names – do not confuse the two! We are talking of the **card** now, which uses the IC LM298 as a component.)
- The card can drive two BO motors independently, providing drive current up to 2A.
- Apart from the LM298 IC, the card contains flywheel diodes for spike suppression and a 5 V regulator chip (78M05) to generate the logic power supply of 5 V from the motor drive supply which can be a voltage between 7 V and 12 V.

The LM298 IC can go up to 46V for the motor voltage supply, but the external 78M05 voltage regulator used in the card limits the maximum voltage to 12 V.



- Sense_A and Sense_B pins are connected to ground.
- The circuit is configured to different usage styles, using links L1, L2 and L3 (shown in blue on the left).
  (Links are small connectors which short adjacent pins).
- Links L1 and L2 short the enable inputs A and B to +5 V. Thus both motor drives are permanently enabled and will run at full speed. If we wish to control the rotation speed of the motor using PWM signals, Links L1 and L2 should be removed and the enable pins should be connected to PWM capable port pins on Arduino.

Link L3 controls whether the 5 V regulator will be activated or not. If the link is present, higher voltage supply is connected to the 3 terminal regulator input. The 5 V output generated by the regulator can be used by digital circuits on board as well as for powering an Uno or a Nano card.

5 Supply connections

Several options exist for powering Arduino and the motor driver card. The following table summarises the various configurations which can be used:

Arduino Powered by	L298N Logic Powered by	Link L3 status	Connection between L298N 5V and Arduino 5 V
USB/Wall plug supply	on-board 78M05	Present	Absent
USB/Wall plug supply	Arduino	Removed	Present
Motor driver 78M05	on-board 78M05	Present	Present†

† In this case a preferred configuration is to connect V_s to V_{in} pin of Arduino.
Only if that is too noisy, 5 V(driver) to 5 V(Arduino) should be connected.

In all cases, grounds of the motor driver card and Arduino should be connected. Notice that Arduino **must not be powered by USB or wall plug supply** if both Link L3 and the connection between 5 V terminal on L298N card and Arduino 5 V pin are present.

When the board is transferred to a project (say a robot), the USB connection is no more available. In this case, the last configuration will be used.

1. We connect the ground of L298N card to ground of Arduino.
2. We retain the link L3.
3. This will power the logic circuits in L298N with the 5 V output of the on-board 78M05.

4. We connect the V_S supply to V_{in} input of the Arduino card. This will provide power to the Arduino card from the main 10-12 V battery.

5. At this time, the Arduino card must **not** be powered using the USB cable.

The BO motor creates a lot of noise due to sparking in the brush contacts. This noise can couple through the L298N Board into the micro-controller on the Arduino and cause it to malfunction. To avoid it, it is advisable to solder a 0.1μF ceramic capacitor across the motor terminals.
Additionally, if V_S is connected to V_{in} of Arduino, a 0.1μF ceramic capacitor can be connected on the Arduino board between V_{in} and Ground to reduce noise. If problems with noise persist, Arduino board can be powered by connecting 5 V terminal of the L298N card to the 5 V pin on Arduino.