Expt. 5: Controlling Motors using Arduino boards

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

Department of Electrical Engineering
Indian Institute of Technology, Bombay

March 4, 2024

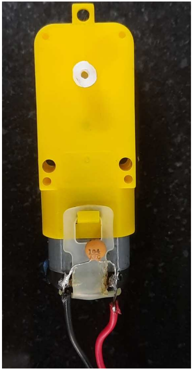
Assembling the set up

Before we can do this experiment, we have to assemble the set up. The various steps for this are:

- Soldering a capacitor and wires to the Battery Operated (BO) motor.
- Mounting the motor on a stage and fixing an encoder disk on its axis.
- Mounting a reflective sensor with an LED and photo-diode which gives digital outputs of ‘1’/‘0’ as the encoder disk rotates.
- Connecting the motor to its driver and connecting the driver to Arduino Nano card.

Soldering a Capacitor across BO Motor Terminals

- Before we connect the motor to our circuits, we need to solder a 0.1μF ceramic capacitor across its terminals to suppress noise.
- We also need to solder wires to the motor terminals, so that we can connect it to the motor driver.

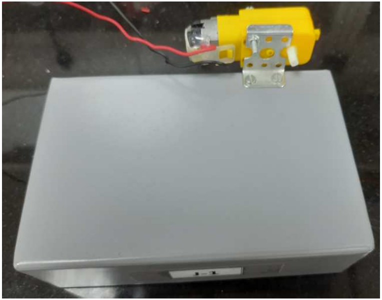


Solder the capacitor and wires to a BO motor.
Seek help from your TA/EE staff if you are unable to solder these neatly.

Mounting the motor on a platform

You are given a platform with a bracket to mount the BO motor.

Platform with mounted bracket

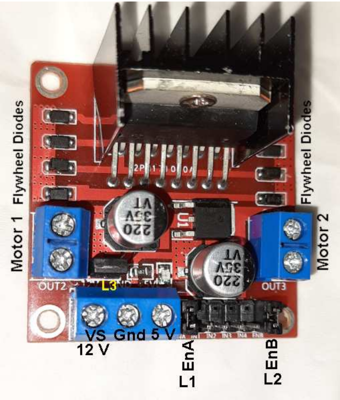


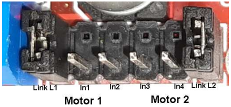
BO motor mounted on the platform

- Mount the BO motor on the bracket using nuts and bolts provided to you, as shown in the figure above.
- If your motor has a single shaft, it should be pointing outwards.
- Wires soldered to the motor will be connected to the motor driver board.

Configuring the Motor Driver Board

We shall use the motor driver L298. This board can drive two BO motors, but we are going to use only one motor in this experiment.

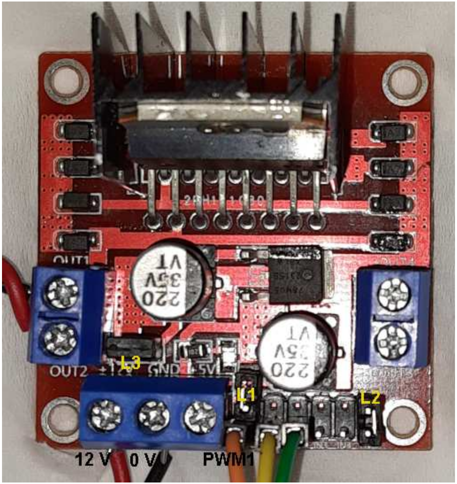




The board has 3 **"links"**. (A link is a little plastic part with a metal inside, used to connect two pins to each other). Locate the three links:

1. Link L1 is located just to the right of the power block on the bottom, between the power block and In1 connection.
2. Link L2 is at the bottom right just above the PWM2 pin.
3. Link L3 is at left, just above the power block and to the right of OUT2 motor connection.

Using Links to configure the motor driver

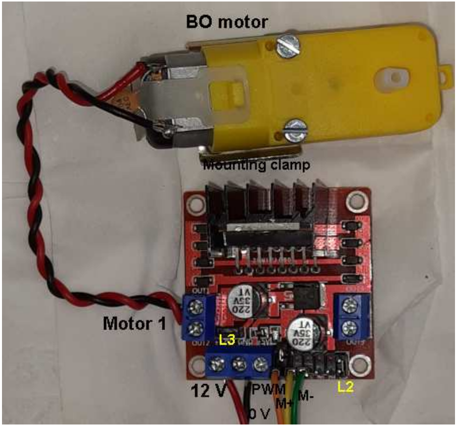


While removing a link, if you just take it off, you are likely to lose it.

To remove a link, pull it out and plug it back such that it covers only one of the two pins over which it was originally placed.

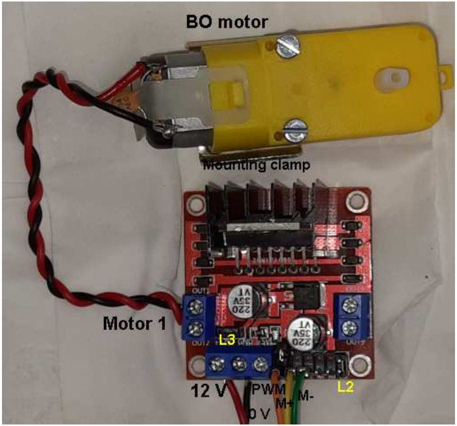
(See link L1 which has been moved up by one position.
This exposes the pin previously shorted by the link and now has an orange wire labelled PWM1 connected to it.)

Connecting Motor and Power to the Driver Board



- The motor leads (soldered earlier) should be connected to Motor 1 terminals (OUT1 and OUT2) on the L298 motor driver board.
- Connect a DC power supply (programmed to 12 V) to the V_S and Ground terminal blocks on the L298 card. The third terminal (meant for 5 V DC) **should be left open.**
- Connect the ground terminal of L298 card to the ground pin of Arduino.

Connections between Motor Driver and Arduino



- Remove Link L1. (The link connects the enable input for motor A to 5 V generated by on-board 5 V regulator).
- The pin (which was connected to on-board 5 V through the link earlier) should now be connected to pin 6 of Arduino for PWM control.
- Connect the next two pins (IN1 and IN2) on the motor driver to pins 10 and 11 of Arduino. (Yellow and Green wires in the picture).

Female to male hook up wires should be used for these connections.

Links: L3 in; L1 removed.

Power:

Driver card	Power supply
V_S (10-12 V)	Supply+
Ground	Supply$-$
5 V Terminal	**Open**

Motor to Driver card

OUT1, OUT2 - Motor wires

Motor card to Arduino

Driver Card	Arduino UNO/nano
Ground	Gnd
Enable A	Pin 6
IN1	Pin 10
IN2	Pin 11

After making all the connections, have these checked by your TA.

Sketch for Rotating the Motor

We are all set to rotate the motor.

We shall use the provided sketch (sketch-expt5A.ino) to do this. The sketch defines a PWM value in the code (at the end of the sketch) and runs the motor with this PWM value.

If the defined PWM value is negative, it rotates the motor in the opposite direction.

- Connect your Arduino to the laptop **without powering on the motor driver.**
  (Keithley Power supply output should be OFF).
- Enter, compile and upload the provided sketch (sketch-expt5A.ino).

Rotating the Motor

We'll use 12 V power from the Keithley power supply (Channel 1 or 2).

- Set the voltage on the selected channel to 12 V.
- Program Iset to 1A on this channel and turn ON the output.

The power LED on motor driver should glow.
If it does not, turn everything off and re-check your connections.

If all is well, **your motor will rotate!**

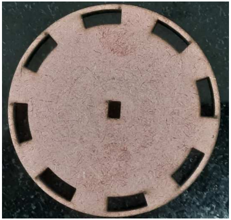
Effect of PWM Value on Motor Speed

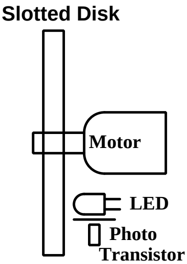
When you have stopped jumping for joy, call the TA and show your success to him/her. Have your lab book signed that your motor connections are right and the motor is rotating.

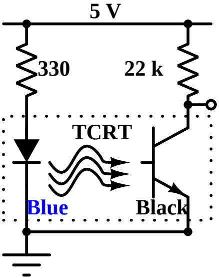
Try various positive and negative values of PWM.
(Absolute value of PWM has to be ≤ 255).
For this you will have to edit the sketch, altering the value of pwmValue near the end of the sketch every time, and re-uploading the binary to Arduino.

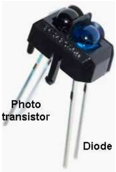
After that, Disconnect the power from the motor driver –
We have more work to do ...

Measuring Rotation Speed



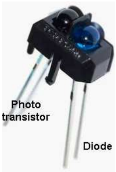


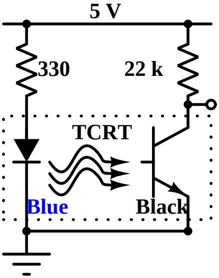


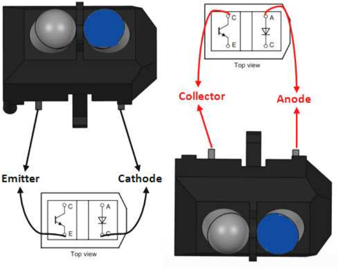


- Mount the encoder disk with cut out sectors on your motor.
- We shall use an infrared LED/photo-transistor pair (TCRT sensor) to detect the rotation of this disk.
- The circuit for the TCRT sensor is shown on the left. Make this circuit on the breadboard where you have mounted Arduino-Nano.
- The output from TCRT should be connected to pin 7 on Arduino Nano.

Measuring Rotation Speed







- Connect the power terminals of the TCRT sensor circuit to 5V and Gnd terminals of Arduino Nano.
- Connect the output from TCRT to pin 7 on Arduino Nano.
- Run the sketch for experiment 5A and Power the motor on as before so that the encoder disk rotates.
- Place the TCRT sensor next to the slots on the rotating encoder disk.
- Observe the waveform at the TCRT output on the oscilloscope.

Measuring Rotation Speed

- Due to light beam interruption by rotating sectors on the encoder disk, you will see a (nearly) square waveform on the oscilloscope.
- The time period of this waveform is the time taken for one slot to move into the location of the previous slot.
- Since there are 8 slots on the disk, the time for one full rotation is 8 times the time period of the waveform.
- The reciprocal of this is the speed in revolutions per second.
- Speed is often specified in revolutions per minute (RPM). That is just 60 times the revolutions per second that we had computed.

Rotation Speed vs PWM Plot

- Modify the PWM value in the given sketch (sketch-expt5A.ino) from 32 to 224 in steps of 32.
- Notice that at low PWM values, the motor may not rotate at all. This is normal.
- Compute Rotation Speed for each value of PWM. Enter these values in your note book as a table and have these signed by your TA.
- Plot Rotation Speed vs the applied PWM.

After this, enter, compile and upload the second sketch given to you. (sketch-expt5B.ino)
See the motor accelerate in one direction, decelerate, then accelerate in the other direction and decelerate to idle condition.

PID control for motor speed

We have learnt about PID control in a lecture. We are going to apply that technique to control the speed of a rotating motor.

- Read through the sketch – sketch-expt5C.ino
- The sketch monitors the output of TCRT sensor. It measures the elapsed time between two rising edges of the TCRT output. This is the sector time we had seen on the oscilloscope. From this, the sketch computes the speed in revolutions per second.
- Based on the desired speed, it computes the error value between the set speed and the measured speed.
- It computes the proportional, integral and differential error and sets the PWM value to a weighted sum of these.

PID control for motor speed

- Uncomment the debug statements in sketch-expt5C.ino
  (These print out the values of pwm and speed on the monitor).
- Recompile and upload the sketch.
  With the current choice of proportionality constants, PID control takes several minutes to converge. This allows you to see how the pwm value is continually adjusted to approach the set speed.
- After the speed has stabilized, change the Keithley supply voltage to 11 V.
- The speed will drop initially, but will eventually settle down to the same desired speed with this changed voltage.