DC MOTOR POLARITY CONTROL

1.0 Learning Outcomes

After completing this Exercise, you will be able to:

1. Identify the main hardware components required to control a DC motor using an Arduino Uno and a Maker Drive motor driver.
2. Safely connect a DC motor circuit using an external power supply and a common ground.
3. Explain how an H-bridge changes the polarity across a DC motor to control clockwise and counterclockwise rotation.
4. Use Simulink to manually control the direction and speed of a DC motor using digital signals and PWM values.

After completing this Exercise, it is recommended that you refer back to the Learning Outcomes.

2.0 Hardware Set Up

This hardware set up uses an Arduino Uno to control a DC motor through a Maker Drive motor driver, see Figure 1. The Arduino sends control signals to the motor driver, while the motor driver supplies the higher current needed by the motor. A separate 9V battery powers the motor side of the circuit, and the Arduino, motor driver, and battery must share a common ground so the control signals work correctly.

The required hardware for this exercise is as follows:

1. Arduino Uno
2. Maker Drive motor driver board
3. DC motor
4. 9V battery and clip
5. Breadboard
6. Breadboard wires (various)

7. USB cable

Step-by-step set-up instructions

1. Connect the DC motor wires to one motor output channel on the Maker Drive, such as M1A and M1B.
2. Connect the 9V battery positive wire to the Maker Drive motor power input, usually labelled VIN, VM, or +.
3. Connect the 9V battery negative wire to the Maker Drive GND terminal.
4. Connect the Maker Drive GND to one of the Arduino GND pins. This creates a common ground.
5. Connect Arduino control pins to the Maker Drive input pins. For example, connect Arduino digital/PWM pins to the Maker Drive motor input pins for speed and direction control.
6. Check all wiring carefully before powering the circuit. Make sure the motor is connected to the motor driver, not directly to the Arduino.
7. Connect the Arduino to the computer using the USB cable.

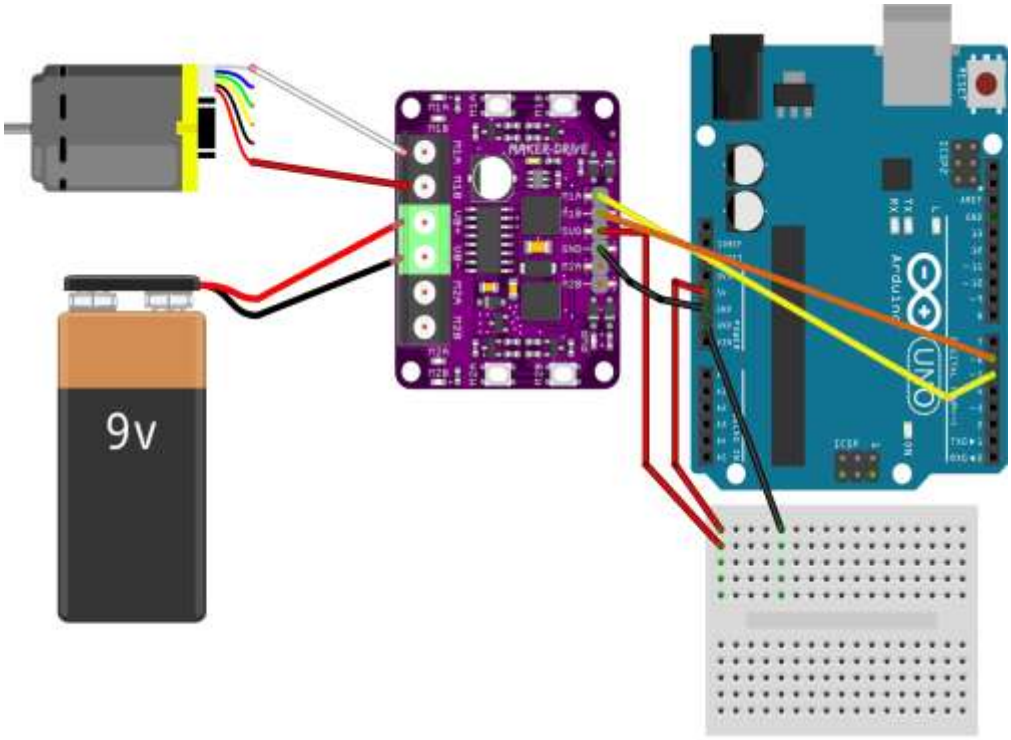

Figure 1: *Hardware Set Up for DC Motor Polarity Control*

3.0 Simulink Set Up and Results

The Simulink block diagram for the H-bridge algorithm design is given in Figure 2, with the logic used for controlling the 'forward' and 'reverse' motion of the DC motor. Please note that to implement this in real-time on a physical system, the switches would need to be external, or logic built into Simulink (e.g., use of saturation blocks that will be later detailed). Based on the electronic circuit set-up provided above, the operation of the H-bridge polarity 'control' operates as follows:

1. Clockwise Motion
    - Set IN1 to High
    - Set IN2 to Low
2. Counterclockwise Motion
    - Set IN1 to Low
    - Set IN2 to High
3. Stop Motion
    - Set both IN1 and IN2 to Low
    - Set both IN1 and IN2 to High

Once, set-up and operating within Simulink, perform the following:

1. Change the switches as detailed above and view the scopes to explore how the signals are changing
2. As the PWM has values between 0 and 255 (150 and 255 are initially given below), investigate changing the numbers within the gain blocks to alter the speed of the DC motor.

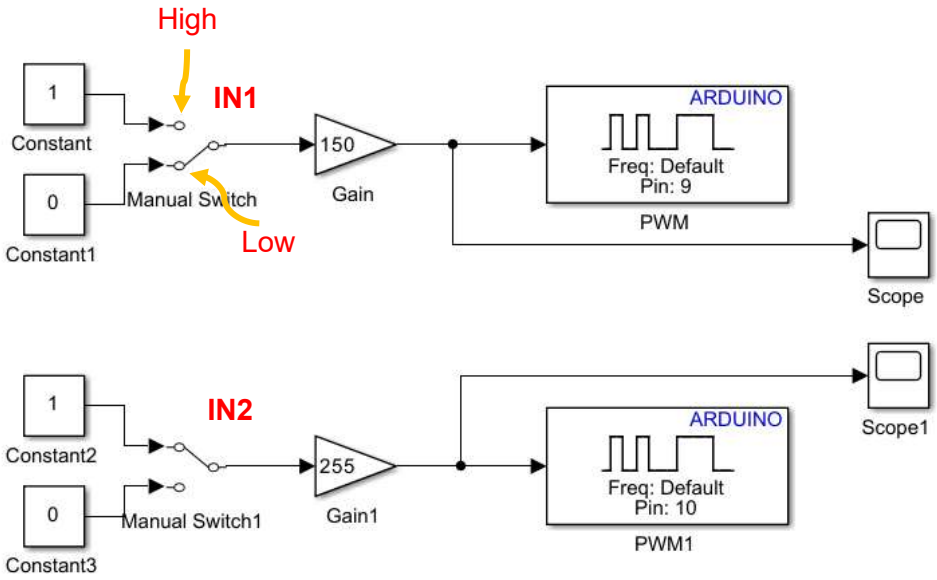

Figure 2: *Simulink Diagram for Manual Polarity Control of a DC Motor using a H-bridge*

4.0 Concluding Remarks

This exercise demonstrated how an Arduino Uno and Maker Drive motor driver can be used to control the direction and speed of a DC motor. By changing the H-bridge input signals, the polarity across the motor can be reversed to produce clockwise or counterclockwise rotation. The use of PWM also allows the motor speed to be adjusted. This provides a practical introduction to motor driver circuits, polarity control, and real-time control implementation using Simulink.

Recommended citation:
Pickering, J.E. (2026). DC Motor Polarity Control. ACE-Lab Teaching Notes. Available at: www.ace-lab.co.uk.