

# ***H-Bridge Driver ICs in DC Motor Position Control: Architecture, Integration, and Course-Project Evidence***

**Yiming Liu**

*Electronics Science and Technology, Silesian University of Technology, Gliwice, Poland  
yl318556@student.polsl.pl*

**Abstract.** This study examines the H-bridge driver integrated circuit as a key hardware element in a DC motor position-control system. It is based on a course project that uses an Arduino controller, a potentiometer feedback signal, Octave-based controller tuning, and an L293D-style driver to realize bidirectional motor actuation. The driver is not treated as a transparent output block. Instead, its logic interface, output stage, voltage drop, protection behavior, and thermal limits are related to the closed-loop response observed at the shaft. The discussion combines a transfer-function model, controller-performance metrics from the Octave comparison, and a wiring-level reading of the motor circuit. It also compares the basic L293D architecture with more integrated motor-driver and motor-control IC options. The analysis suggests that a simple bipolar H-bridge is still suitable for demonstrating position control, while a driver with current limiting, fault reporting, and better diagnostic feedback is needed for repeatable dynamic evaluation. Driver selection should therefore be considered during control-system design rather than after the controller has already been chosen.

**Keywords:** DC motor, H-bridge driver, Integrated circuit, PWM, Position control

## **1. Introduction**

DC motor position control is a useful control-engineering example because it links feedback design with measurable shaft motion. It also provides students a compact platform for system modeling, controller tuning, and embedded implementation. Khalifa, Amhedb, and Al Sharqawi used a real-time LabVIEW-based position-control system to demonstrate PID control of a DC motor [1]. Pallejà Cabré et al. developed a Z-plane teaching platform that combines speed and position control with data acquisition and embedded implementation [2]. Talebi's IMC-based PID design and Kızmaz's comparison of LQR, PID, and sliding-mode strategies further show that position-control performance remains a practical design issue, not only a classroom example [3, 4].

The present study addresses a narrower but often under-discussed part of that problem: the H-bridge driver IC between the controller and the motor winding. A microcontroller can calculate the control effort, but it cannot directly supply the bidirectional current required by a brushed DC motor. Barkas et al. emphasize that brushed DC motor drives combine control technique, switching hardware, sensing, and protection in a single practical design problem [5]. That observation

motivates the central claim of this paper: the driver IC should be analyzed as part of the controlled plant interface, not as a passive connector after the control law has been chosen.

The main question is how the architecture of an H-bridge driver IC affects the conversion of a position-control command into motor motion. In the project, Octave is used to model the plant and compare controller choices, while the Arduino circuit provides the practical signal path from feedback reading to PWM and direction outputs. The aim is not to propose a new semiconductor design. It is to explain how a course-level motor-control circuit reveals the main tradeoffs between basic and smart H-bridge ICs.

## 2. Closed-loop system structure

A DC motor position-control loop can be described as a sequence of comparison, compensation, power conversion, mechanical response, and feedback measurement. The reference angle is compared with the measured position, and the controller converts the resulting error into a command. The command then enters a power interface, where logic-level signals are translated into motor terminal voltage. Classical control texts model this process by separating the controller, actuator, plant, and sensor, but the separation is only an analytical convenience [6]. In the physical circuit, the actuator interface imposes voltage, current, delay, and saturation limits that change what the controller can achieve.

The Octave model used in the project represents the plant from armature voltage to angular position. With rotor inertia  $J$ , viscous friction coefficient  $b$ , torque and back-EMF constant  $K$ , armature resistance  $R$ , and armature inductance  $L$ , the transfer function in Equation (1) provides the voltage-to-position relation used for simulation. This form is consistent with the standard electromechanical DC motor model used in undergraduate control analysis [7].

$$G_{\text{theta}}(s) = \frac{\Theta(s)}{V(s)} = \frac{K}{s[(Js + b)(Ls + R) + K^2]} \quad (1)$$

Controller comparison was then performed around the same plant model. The optimized PD controller in Equation (2) is used as the main design because it improves transient response without adding the integral term that, in this project, increased the settling time. The PID form in Equation (3) is included as a comparison case because integral action is often useful for steady-state accuracy but can also amplify practical limitations when saturation and actuator constraints are present [8].

$$C_{\text{PD}}(s) = K_d(s + z) \quad (2)$$

$$C_{\text{PID}}(s) = K_p + \frac{K_i}{s} + K_d s \quad (3)$$

Table 1. Octave control performance metrics comparison

| Controller   | Rise time (s) | Settling time (s) | Overshoot | Steady-state error |
|--------------|---------------|-------------------|-----------|--------------------|
| P baseline   | 1.399         | 4.388             | 7.67%     | 1.68e-04           |
| Initial PD   | 0.311         | 1.228             | 12.68%    | 4.44e-15           |
| Optimized PD | 0.301         | 0.452             | 1.67%     | 8.80e-09           |
| PID trial    | 0.297         | 5.258             | 3.57%     | 5.44e-03           |

Table 1 shows why the optimized PD design is a reasonable project choice. Its rise time remains close to the PID trial, while its settling time is much shorter and the overshoot is lower than the initial PD case. The steady-state error is also small enough for the simulated position target. These figures do not prove that the real motor will behave identically, but they define the response that the hardware path must try to reproduce.

The Arduino implementation gives this loop a concrete signal path. The potentiometer wiper is connected to A0 and mapped from 0-1023 to 0-180 degrees. The target angle is 90 degrees. Pins D7 and D8 select motor direction, while D9 supplies PWM to the driver enable input. The control error and PWM command are therefore generated in software, but the motor only receives energy after the driver IC interprets these logic states. Equation (4) summarizes the measurement and command-limiting relation used in the implementation.

$$e(t) = \theta_{ref}(t) - \theta(t), \quad PWM = \text{clip}(|u(t)|, 0, 255) \quad (4)$$

This signal path explains why the driver IC cannot be ignored. The controller updates a command in discrete time, but the motor current changes through an inductive winding and the shaft position changes through a mechanical load. A driver with a large voltage drop, a narrow current margin, or unreported thermal shutdown can make the physical response depart from the simulated one even when the control law is unchanged. Figure 1 places the H-bridge driver between the controller output and the motor terminals. It also shows the feedback path: the potentiometer returns the measured shaft position to the Arduino, and the driver converts PWM and direction signals into bidirectional motor voltage.

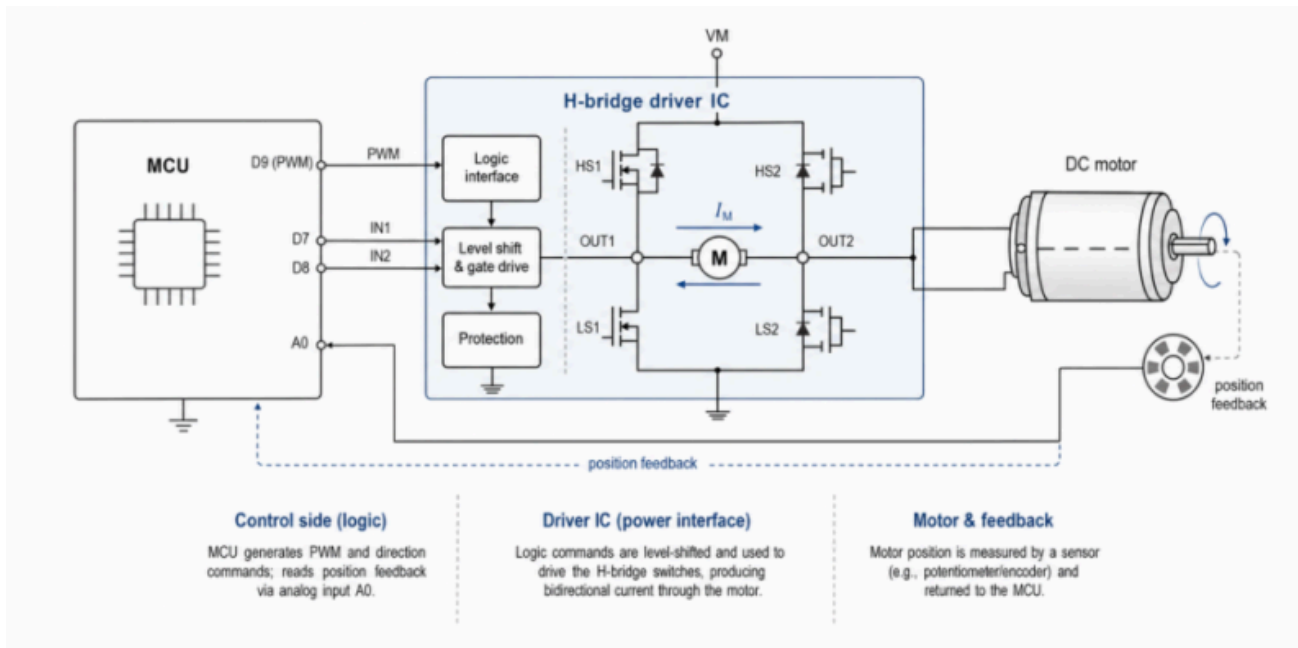


Figure 1. Closed-loop DC motor position-control system

### 3. Basic architecture of H-bridge driver ICs

An H-bridge is the standard power-stage arrangement for bidirectional brushed DC motor control. Four switching devices are arranged around the motor load so that one diagonal pair applies one polarity, while the other diagonal pair applies the opposite polarity. Brake, coast, and PWM

operating modes are obtained by changing which devices are switched and when. At the IC level, these functions are integrated with logic inputs, level shifting, output-drive circuitry, protection circuits, and supporting components that simplify motor-control system design.

The driver is also a thermal component. Zhou et al. show that power semiconductors in an H-bridge should be evaluated not only for switching behavior but also for temperature distribution and loss paths [9]. For a course project, that point matters because the controller can demand a duty cycle that appears valid in software while the package is already close to a thermal or current limit. The power stage therefore defines part of the feasible operating region of the closed-loop control system.

The L293D is a useful baseline device because it makes the internal H-bridge concept visible without requiring a discrete transistor design. Texas Instruments describes the L293x family as quadruple high-current half-H drivers, and the L293D variant includes clamp diodes for inductive transient suppression [10]. In the project circuit, Arduino pins D7, D8, and D9 are converted into bidirectional motor voltage through this IC. That conversion is simple enough for a laboratory demonstration, but it hides several nonideal effects inside the package.

One practical limitation of this architecture is not that it fails to drive the motor. It is that it supplies little diagnostic information about the drive process. The serial monitor can report target angle, measured angle, control error, and PWM value, but it cannot directly report winding current, voltage loss across the output stage, or chip temperature. If the motor stalls or the load changes, the driver IC may become the dominant limiting factor without providing explicit diagnostic feedback. Figure 2 shows the internal bridge structure in simplified form, including the logic inputs, switching devices, clamp or protection elements, and motor outputs. These blocks are part of the control experiment because their electrical limits affect the response, even when the Arduino does not measure those limits directly.

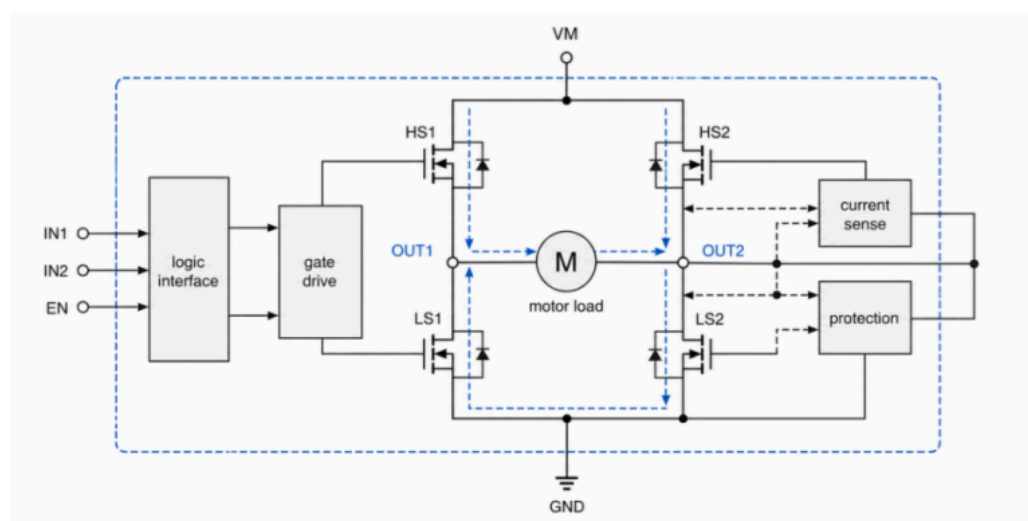


Figure 2. Generic internal architecture of an H-bridge motor driver IC

#### 4. Integration trend of smart motor driver ICs

Modern motor-driver ICs extend the basic H-bridge by integrating features that help the controller observe and constrain the power stage. The DRV8876, for example, is specified by Texas Instruments as an H-bridge motor driver with integrated current sense and current regulation [11]. Similarly, the STSPIN840 is described by STMicroelectronics as a compact dual brushed DC motor

driver featuring a protected low-resistance dual full-bridge power stage [12]. Compared with traditional H-bridge drivers, these devices are not merely smaller replacements for older bridge drivers; they change what can be measured and protected during operation.

A motor driver and a motor-control IC should also be distinguished. The TMC4671 is a dedicated motion controller that provides field-oriented control support for PMSM, BLDC, and stepper motors rather than serving as a brushed DC H-bridge power driver [13]. Including it in the comparison clarifies the boundary between the control-side IC and the power-stage driver. In a position-control system, both layers may be present, but they solve different problems. Table 2 compares these device roles from the viewpoint of a course-level DC motor position-control experiment. The comparison emphasizes output-current capability, current information, protection behavior, and diagnostic feedback.

Table 2. Functional comparison between basic and modern motor-control IC options [10-13]

| Feature             | Basic L293D-style driver                                 | Modern driver or control-side IC                                                |
|---------------------|----------------------------------------------------------|---------------------------------------------------------------------------------|
| Power stage         | Bipolar half-H outputs with higher voltage drop          | Low-resistance MOSFET bridges; external stages for some control-side ICs        |
| Current information | Usually unavailable to the controller                    | Current sense, current regulation, or status feedback                           |
| Protection          | Clamp diodes and package-level limits dominate           | Overcurrent, undervoltage, short-circuit, thermal shutdown, and fault reporting |
| System role         | Translates logic inputs into bidirectional motor voltage | Power or diagnostic interface; control-side motion component                    |

For position control, current sensing and protection are especially important when the shaft is blocked or the load changes abruptly. A controller that sees only position error may keep increasing PWM until the driver or motor reaches an electrical limit. A driver with a current-sense output, current regulation, or fault indication allows the embedded controller to detect that the actuator has entered a constrained state. The same control law can therefore be made safer and more repeatable by choosing a driver that exposes the relevant physical quantity.

Higher integration also changes the experimental workflow. With a basic driver, external sensing circuits must be added before current can be recorded. With a smart driver, some of that measurement path can be part of the IC or the recommended application circuit. The difference is not just convenience. It affects whether a student project can compare simulated response, measured position response, and electrical load response in the same experiment.

A practical selection rule follows from this comparison. If the assignment only requires visible direction reversal and approximate position regulation, the older L293D-style bridge is adequate. If the assignment asks why the response changes under load or why the controller output rises without corresponding shaft motion, the driver must provide more evidence than two motor output pins can supply. Current sense, current limiting, and fault reporting make the actuator interface observable enough for engineering analysis, not only for circuit construction [10-13].

## 5. Project-based analysis and improvement path

The existing project can be directly mapped onto the driver IC architecture. The Arduino serves as the digital controller, the potentiometer provides position feedback, the L293D acts as the

bidirectional power interface, and the DC motor with its mechanical load represents the plant. Rather than designing an H-bridge from discrete transistors, the project examines the driver IC as a functional component within the control loop. Figure 3 illustrates this signal path, showing the controller inputs, driver interface, motor connections, feedback sensor, and common ground. The diagram also separates software-visible variables, such as position error and PWM command, from physical quantities inside the driver path, including winding current, voltage drop, switching transitions, and heat.

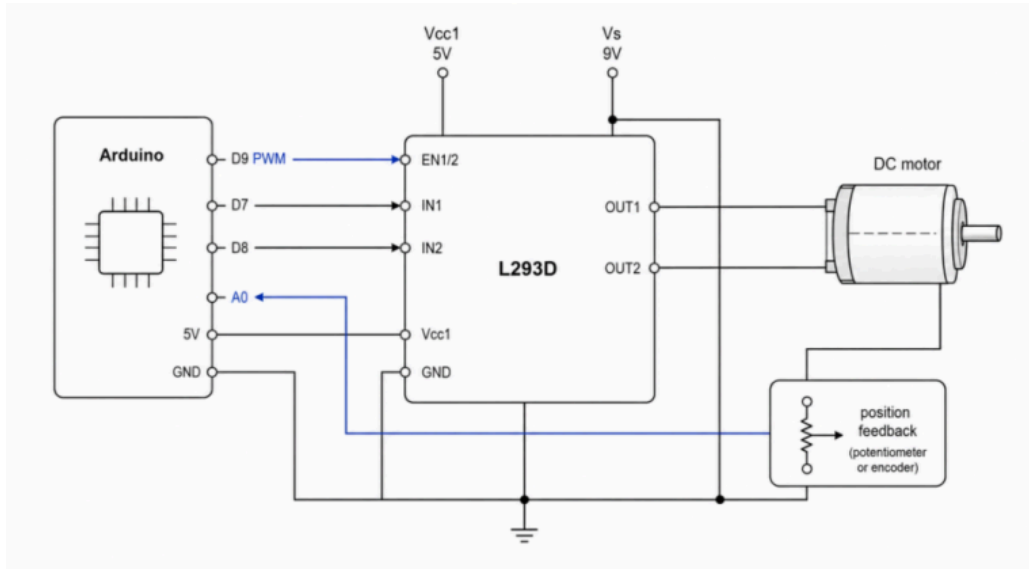


Figure 3. Detailed Arduino-L293D wiring diagram for the DC motor position-control experiment

A similar project-level design philosophy appears in the economical shake-table work of Parajuli, Bhatt, and Kawan, where a brushed DC motor and PD control are used to build a practical test platform rather than an abstract controller alone [14]. Sensorless and intelligent-controller studies also show that the motor-drive problem often shifts from selecting a controller to deciding what the controller can infer from limited electrical signals [15]. The course circuit is simpler than those systems, but it exposes the same engineering question: what information is available to the controller when the motor does not move as expected?

A realistic improvement path is to replace the basic driver with a modern H-bridge driver that provides current sensing and fault reporting. Erdogan and Cindemir's recent H-bridge driver study reinforces the continuing relevance of bridge-driver design for brushed DC motor control [16]. The upgraded experiment would keep the same target angle, feedback mapping, and controller comparison, but it would add current measurement and fault status as logged variables. That change would make it possible to compare a fast simulated response with the electrical effort required to obtain it.

The PWM implementation should also be interpreted through the current path rather than only through duty cycle. ROHM's PWM driving note explains that voltage application and current regeneration alternate during brushed DC motor PWM operation [17]. Microchip's AN893 application note similarly treats bidirectional brushed DC control as a problem of coordinating PWM generation with bridge-state selection [18]. For this project, the important point is that changing the driver can change the current waveform even when the high-level position controller is unchanged.

The proposed improvement should remain proportional to the assignment. Semiconductor device modeling, layout extraction, and chip fabrication are outside the project. A more useful extension is to add observability to the existing loop: record position, PWM, current, and fault state under the same target command. That extension would connect the course simulation to a more complete motor-drive experiment without changing the basic educational purpose.

## 6. Conclusion

The H-bridge driver IC is not a minor peripheral in this DC motor position-control loop. It is the hardware interface that turns low-power control commands into bidirectional motor energy. The Octave model and controller comparison define the desired dynamic response, and the optimized PD controller gives a reasonable balance between rise time, settling time, overshoot, and steady-state error. In the real circuit, however, that response is limited by the driver's voltage drop, current margin, switching behavior, and thermal condition. This means that the driver should be selected as part of the control-system design, not only after the controller has been chosen.

For the present course project, the L293D-style driver is still a reasonable choice. It makes direction control and PWM actuation easy to implement, and it lets the student observe the basic relationship among a 90-degree target, position feedback, controller output, and motor reversal. Its simplicity also makes the closed-loop signal path easy to explain in a laboratory report. The limitation is that the same simple circuit hides much of the electrical behavior. The driver provides little direct information about winding current, voltage loss, temperature, or fault state. When the measured position does not follow the simulated response, the software variables alone cannot fully show whether the problem comes from the controller, the load, or the actuator interface.

The most practical improvement is to make the actuator path more observable without changing the whole assignment. A modern H-bridge driver with current sensing, current regulation, and fault reporting would keep the same reference angle, feedback sensor, and controller comparison, while adding electrical evidence about how the motor is being driven. With that upgrade, the experiment could compare simulated position response, measured shaft motion, PWM command, and driver current under the same operating conditions. It would also show why a control law that works well in simulation can behave differently in hardware. The H-bridge driver IC is therefore both a power component and a measurement boundary in the experiment. Recognizing that boundary makes the project more technically complete while keeping it within the scope of a course-level DC motor position-control system.

## References

- [1] Khalifa, M., Amhedb, A. H., & Al Sharqawi, M. (2021). Real time DC motor position control using PID controller in LabVIEW. *Journal of Robotics and Control (JRC)*, 2, 342–348.
- [2] Pallejà Cabré, T., Saiz Vela, A., Tresanchez Ribes, M., et al. (2021). Didactic platform for DC motor speed and position control in Z-plane. *ISA Transactions*, 118, 116–132.
- [3] Talebi, N. (2021). Position control of permanent magnet DC motor using IMC-based PID controller. *Transactions on Machine Intelligence*, 4, 90–99.
- [4] Kızmaz, H. (2023). Comparative analysis of optimal control strategies: LQR, PID, and sliding mode control for DC motor position performance. *Gazi University Journal of Science Part A: Engineering and Innovation*, 10, 571–592.
- [5] Barkas, D. A., Ioannidis, G. C., Psomopoulos, C. S., et al. (2020). Brushed DC motor drives for industrial and automobile applications with emphasis on control techniques: A comprehensive review. *Electronics*, 9, 887.
- [6] Ogata, K. (2010). *Modern control engineering* (5th ed.). Pearson Prentice Hall, Upper Saddle River.
- [7] Nise, N. S. (2019). *Control systems engineering* (8th ed.). Wiley, Hoboken.

- [8] Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design, and tuning* (2nd ed.). ISA, Research Triangle Park.
- [9] Zhou, D., Peng, Y., Iannuzzo, F., et al. (2020). Thermal mapping of power semiconductors in H-bridge circuit. *Applied Sciences*, 10, 4340.
- [10] Texas Instruments. (2016). *L293x quadruple half-H drivers*. <https://www.ti.com/lit/ds/symlink/l293.pdf>.
- [11] Texas Instruments. (2019). *DRV8876 H-bridge motor driver with integrated current sense and regulation*. <https://www.ti.com/lit/ds/symlink/drv8876.pdf>.
- [12] STMicroelectronics. (2018). *STSPIN840 compact dual brushed DC motor driver*. <https://www.st.com/resource/en/datasheet/stspin840.pdf>.
- [13] Analog Devices. (2024). TMC4671 data sheet: Dedicated motion controller for 2-phase/3-phase PMSM. <https://www.analog.com/media/en/technical-documentation/data-sheets/TMC4671.pdf>.
- [14] Parajuli, B., Bhatt, B., & Kawan, C. K. (2025). Development of economical shake table with brushed DC motor and proportional-derivative control. *Experimental Techniques*, 49, 635–654.
- [15] Salmaninejad, A., & Mayorga, R. V. (2021). Sensor-less brushed DC motor speed control with intelligent controllers. *WSEAS Transactions on Systems*, 20, 140–148.
- [16] Erdoğan, S., & Çindemir, S. (2025). H-bridge motor driver circuit for control of brushed DC motors. *Proceedings of Automotive Science and Technology*, 1, 77–84.
- [17] ROHM Semiconductor. (2022). Driving brushed DC motors using PWM output: Principles of PWM driving. *ROHM Semiconductor*. <https://techweb.rohm.com/product/motor/motor-types/328/>.
- [18] Microchip Technology Inc. (2015). Low-cost bidirectional brushed DC motor control using the PIC16F684. *Microchip Technology Inc.* <https://www.microchip.com/en-us/application-notes/an893>.