Multidisciplinary | Peer-Reviewed
Open Access | Research Journal
ISSN: 3139-2954

Article ID: IRJIST-2026-100
Volume 1 | Issue 4 | 2026
DOI: https://doi.org/10.67308/irjist.100

International Research Journal of Innovation in Science and Technology

RPM Display and Speed Control of a Brushless DC Motor Using 8051 Microcontroller

Ankita Sasane[1], Vaishnavi Dhakare[2], Sahil Sarang[3], Prajwal Bake[4], Siddharth Bhorge[5], Rahul Pol[6]

[1-6]Department of Electronics and Telecommunication Engineering (E&TC), Vishwakarma Institute of Technology, Pune, Maharashtra, India
Correspondence: [5]sidhharth.bhorge@vit.edu; [6]rahul.pol@vit.edu

Abstract

Brushless DC (BLDC) motors are widely used in cooling fans, pumps, robots, medical devices, and electric vehicles because of their efficiency and long service life. However, their shaft speed can vary with supply voltage and mechanical load. Therefore, a practical drive system requires both a speed set point and a means of monitoring the actual shaft speed. This paper describes a low-cost closed-loop speed controller and RPM indicator based on an 8051-family microcontroller. The user selects the reference speed using UP/DOWN push buttons. The microcontroller generates a PWM signal in software using its on-chip timer, which drives the motor through a switching power stage. An infrared (IR) emitter–photodiode pair detects one pulse per revolution from a reflective marker on the shaft. These pulses are counted over a fixed gate time to determine the speed in RPM. The measured speed is compared with the reference speed, and the PWM duty cycle is adjusted after each gate interval. Both reference and measured speeds are displayed on a 16 × 2 LCD. The prototype was tested over a speed range of 500–2500 RPM, with the measured speed remaining within 2.0–3.0% of the set value. The paper presents the design details, measurement procedure, repeated-reading protocol, and error analysis.

Keywords: Brushless DC (BLDC) Motor; 8051 Microcontroller; Closed-Loop Speed Control; Pulse Width Modulation (PWM); RPM Measurement; Infrared (IR) Sensor; Embedded Control System.

1. Introduction

Electric motors convert electrical energy into mechanical work and are widely used in fans, pumps, conveyors, machine tools, electric vehicles, and robotic joints. Among these, the brushless DC (BLDC) motor is particularly attractive because it does not use brushes or a mechanical commutator. Their elimination reduces maintenance requirements and electrical noise while enabling a high power-to-weight ratio and good dynamic response [1,2].

Maintaining a motor at a desired speed is important in practical applications. A cooling fan operating faster than necessary wastes energy and increases mechanical stress, whereas one operating too slowly provides inadequate cooling. Therefore, a practical controller must perform three essential functions: accept a target speed, measure the actual speed, and correct the difference between them.

This work implements these three functions using an 8051 microcontroller. The target speed is entered using two push buttons. A reflective marker mounted on the shaft and an IR sensor generate a pulse train, which the microcontroller counts within a fixed time window to determine the RPM. The speed is regulated using pulse width modulation (PWM): the duty cycle is increased when the motor runs below the target speed and decreased when it runs above the target speed. The main contributions of the paper are:

- ✓ A complete and documented 8051 implementations of software PWM generation, pulse counting, and RPM computation;
- ✓ A description of the power stage used to

drive the BLDC motor using a low-current microcontroller pin;
- ✓ A repeated-reading test procedure with defined no-load and loaded conditions; and
- ✓ An error analysis over a consistently reported speed range of 500–2500 RPM.

2. Literature Review

Closed-loop speed control of small DC motors using the 8051 microcontroller family has been widely documented. Mondal *et al.* [3] developed an embedded closed-loop controller in which an 8051 microcontroller reads a speed sensor and adjusts the motor drive accordingly. Adesina *et al.* [4] and Singh [5] described similar Intel 8051-based designs that integrate sensor feedback, a display, and a keypad. Tiwari *et al.* [6] focused on PWM as a method of controlling the average voltage supplied to the motor. Collectively, these studies demonstrate that a basic 8051 microcontroller can implement a closed-loop speed control system. However, they primarily address brushed DC motors and provide limited details regarding timer configuration and the repeatability of speed measurements.

For BLDC motors, the literature generally focuses on more advanced control techniques. Gamazo-Real *et al.* [7] reviewed position and speed control of BLDC motors using sensorless techniques, while Akrami *et al.* [8] examined various sensorless drive methods. These techniques eliminate the need for position sensors and can improve system robustness; however, they require back-EMF sensing or processing capabilities that may exceed those available in a conventional 8051 microcontroller. Standard textbooks on motor drives and power electronics [1,2,9–11] provide the fundamental models and switching-stage design principles relevant to small-scale controllers such as the one presented in this study.

Two research gaps emerge from this review. First, low-cost 8051-based designs rarely provide sufficient details about software-based PWM generation and speed measurement, making their implementation difficult to reproduce. Second, reported accuracy is often based on individual readings, without repeated trials or clearly specified load conditions. This paper addresses these gaps by describing the PWM generation, motor driver, and sensing arrangement in detail in Sections 3.2–3.4, while Section 3.6 presents a repeated-reading procedure under defined load conditions.

3. Materials and Methods

3.1. System Overview

The controller follows the loop sense → compute → compare → correct → drive. Figure 1 gives the flow of the firmware and Figure 2 the hardware connections. The hardware consists of the 8051 microcontroller with a crystal oscillator, a 16×2 LCD, two push buttons (UP and DOWN), the IR speed sensor, the motor driver stage, the BLDC motor, and a power supply made up of a step-down transformer, bridge rectifier and a fixed voltage regulator (shown as a 7805 in Figure 2). Resistor–capacitor networks provide filtering, reset timing and current limiting.

On power-up the firmware configures the ports, timers, interrupts and LCD, and then enters a continuous loop: read the buttons, update the reference speed, generate the PWM, count the sensor pulses, compute the RPM, compute the error, adjust the duty cycle and refresh the display (Figure 1).

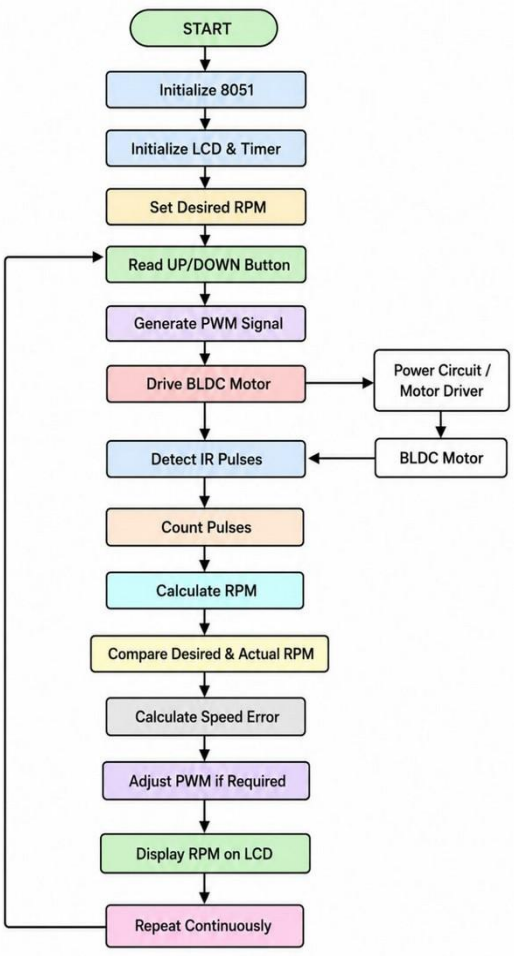


Figure 1: *Firmware flow chart.*

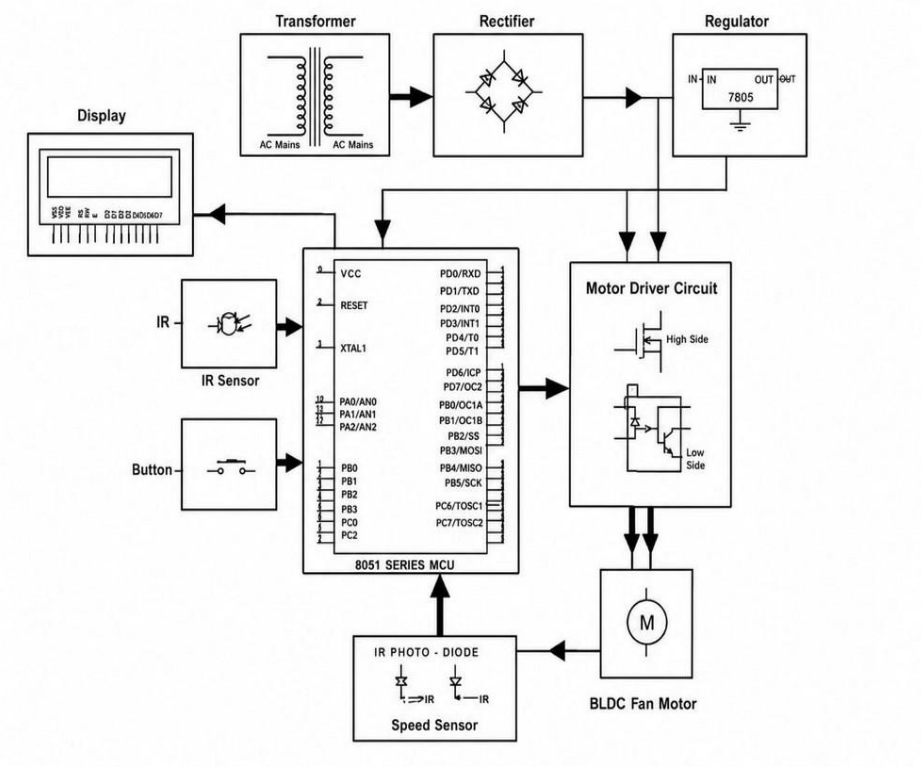


Figure 2: *Block diagram of the speed controller.*

3.2. PWM Generation on the 8051

3.2.1. Why software PWM

The standard 8051 has no PWM peripheral, so the signal is produced in firmware by toggling one port pin under timer control. Timer 0 is used in 16-bit mode (Mode 1) with its overflow interrupt enabled. The interrupt service routine alternates between a HIGH interval and a LOW interval whose lengths set the duty cycle.

3.2.2. Timing and duty cycle

With a crystal of 11.0592 MHz the 8051-machine cycle is 12/fosc, and the timer counts one step per machine cycle. For a PWM period T made of a HIGH time TON and a LOW time T – TON, the duty cycle is

$$D = \frac{T_{ON}}{T} \times 100\ \%$$

The reload values loaded into TH0:TL0 at each interrupt are 65536 – (TON/Tmc) for the HIGH phase and 65536 – ((T – TON)/Tmc) for the LOW phase, where Tmc is the machine-cycle time. The PWM frequency used is 1 kHz and the duty cycle can be changed in steps of 1 %, which fixes the PWM resolution. Raising D increases the mean voltage across the motor and therefore its speed; lowering D has the opposite effect.

3.2.3. Resource allocation

The three time-critical tasks are shared among the on-chip resources as listed in Table 1. The PWM interrupt is kept short (a reload and a pin toggle) so that it does not disturb the pulse count.

Table 1: *Allocation of on-chip resources.*

Resource	Function
Timer 0	1 kHz software PWM generation
Timer 1	1-second RPM measurement gate
INT0 (P3.2)	Counts IR sensor pulses
P1.0	PWM output to MOSFET driver
P1.1	UP button; 100 RPM per press
P1.2	DOWN button; 100 RPM per press
Port 0/2	LCD data and control lines

3.3. BLDC Motor Driver and Control Mechanism

A port pin of the 8051 can source only a few milliamperes, far below what the motor winding needs, so the PWM pin never drives the motor directly. It drives the gate of a switching power stage built from power MOSFETs through a series resistor. The stage connects the motor

supply to the motor for the HIGH interval of each PWM period and disconnects it for the LOW interval, as in the motor-driver block of Figure 2. A flyback diode across the motor and a decoupling capacitor on the supply protect the switch from inductive spikes and supply dips.

Commutation of the three motor phases is carried out by an external three-phase electronic commutation circuit. The microcontroller therefore regulates only the average voltage, and with it the speed, and does not generate the phase sequence. In the control law, the speed error is e = Nref − Nmeas (Equation 3, Section 3.4.2); after every gate interval the duty cycle is changed by one step: increased when e is positive and larger than a dead band of ±20 RPM, decreased when it is negative and beyond the band, and left unchanged inside the band. D is limited to the range 0–100 % so that the output never saturates unpredictably.

3.4. RPM Measurement and IR Sensor Configuration

3.4.1. Sensor arrangement

A small reflective marker is fixed to the rotating part of the motor. An IR emitter–photodiode sensor is mounted 5 mm from the marker. The LM393 comparator converts the reflected IR signal into a clean logic-level pulse wired to INT0. One marker produces one pulse per revolution.

3.4.2. Speed computation

Timer 1 defines a gate time T of 1 second. The interrupt routine increments a counter N on every falling edge. When the gate expires the main loop reads N and computes

$$RPM = \frac{N \times 60}{P \times T} \quad (2)$$

where P is the number of pulses per revolution. A longer gate time gives a finer RPM resolution of 60/(P·T) RPM but slows the response of the loop; the chosen T is a compromise between the two. The error used by the controller is

$$e = N_{ref} - N_{meas} \quad (3)$$

For instance, with a reference of 2000 RPM and a measured value of 1850 RPM the error is 150 RPM, so the duty cycle is increased. The LCD shows both numbers, for example “Set Speed: 2000 RPM” on the first line and “Motor RPM: 1850 RPM” on the second.

3.4.3. Sources of measurement error

The sensor reading depends on the marker position, the ambient IR light and the comparator threshold, and the count is quantised to one pulse per gate interval. These effects are included in the error discussion in Section 5.

3.5. Closed-Loop Operation

The cycle sense → count → compute → compare → correct → drive repeats without interruption. The reference is changed at any time with the UP and DOWN keys, and each press moves it by 100 RPM; the loop then tracks the new value. Reference and measured speed are refreshed on the LCD after every gate interval.

3.6. Experimental Setup and Procedure

The prototype was tested with the sensor, driver, PWM logic, LCD and push buttons described above. The reference speed was set to 500, 1000, 1500, 2000 and 2500 RPM. This single range of 500–2500 RPM is used throughout the paper. The supply voltage was kept at 12 V in all tests. Two load conditions were defined: L0 (“No-load”), the motor running free with only its fan or rotor attached, and L1 (“Loaded”), with adjustable friction applied to the motor shaft. To make the results repeatable, the following protocol was applied at every set point:

- ✓ the motor was allowed to settle for 5 seconds after each change of reference;
- ✓ five consecutive readings of the LCD, showing the IR-sensor RPM, were recorded;
- ✓ the mean, the standard deviation and the percentage error of the five readings were calculated; and
- ✓ the whole sequence was carried out under two load conditions.

4. Results

4.1. Speed tracking

Table 2 lists the first set of readings. In every case the measured speed is close to the reference, and the relative error lies between 2.0 % and 3.0 %. The PWM duty cycle rises from 20 % at 500 RPM to 80 % at 2500 RPM, which confirms the proportional link between duty cycle and speed stated in Section 3.2.

The mean absolute error over the five points is about 2.45 %.

Table 2: *Initial readings (single run, no-load).*

No	Set RPM	Measured RPM	Error (RPM)	Error (%)	PWM duty (%)
1	500	485	15	3.00	20
2	1000	975	25	2.50	35
3	1500	1465	35	2.33	50
4	2000	1960	40	2.00	65
5	2500	2440	60	2.40	80

The absolute error in RPM grows with speed, but the relative error does not, which suggests that the error comes mainly from sensing and resolution and not from a speed-dependent fault. Every measured value lies below its reference, so the offset is systematic and not random.

4.2. Repeatability and load

Table 3 gives the results of the repeated-reading protocol of Section 3.6 for the two load conditions. The standard deviation (SD) shows the scatter of the readings, while the error shows how far the mean is from the reference. The loaded condition increases the percentage error at every set speed: the error rises from 2.01–3.04 % under no-load to 5.80–7.20 % under load.

Table 3: *Repeated readings (five per set point).*

Set (RPM)	Condition	Readings (RPM)	Mean	SD	Error (%)
500	No-load	482, 485, 487, 484, 486	484.8	1.92	3.04
500	Loaded	470, 474, 472, 468, 471	471.0	2.24	5.80
1000	No-load	972, 975, 978, 974, 976	975.0	2.24	2.50
1000	Loaded	938, 942, 940, 936, 944	940.0	3.16	6.00
1500	No-load	1460, 1465, 1468, 1463, 1467	1464.6	3.21	2.36
1500	Loaded	1405, 1410, 1414, 1408, 1412	1409.8	3.49	6.01
2000	No-load	1955, 1960, 1964, 1958, 1962	1959.8	3.49	2.01
2000	Loaded	1875, 1882, 1878, 1885, 1880	1880.0	3.81	6.00
2500	No-load	2435, 2440, 2445, 2438, 2442	2440.0	3.81	2.40
2500	Loaded	2315, 2322, 2318, 2325, 2320	2320.0	3.81	7.20

5. Discussion

The deviation between set and measured speed is attributed to four causes:

i. a change in motor loading, which shifts the duty cycle needed for a given speed;
ii. the tolerance of the IR sensor and the position of the marker;
iii. the finite PWM resolution, since the duty cycle can only change in whole steps; and
iv. small variations of the supply voltage. Increasing the gate time, mounting the sensor rigidly and using a regulated motor supply would each reduce the error.

A proportional step correction with a dead band tends to leave a small residual error of the kind seen in Table 2; an integral term, as in a PID loop, would remove it. The larger offset under load (Table 3) is consistent with cause (i): friction on the shaft changes the duty cycle required for a given speed, and the dead-band correction does not fully recover it.

6. Conclusions

A low-cost BLDC speed controller and RPM indicator based on an 8051 microcontroller have been designed and tested. Software PWM generated using Timer 0, an IR sensor configured to detect one pulse per revolution over a fixed gate time, and duty-cycle correction applied after each interval enable the system to maintain the measured speed within approximately 3% of the reference speed over a range of 500–2500 RPM under no-load conditions. The LCD simultaneously displays the reference and measured speeds. As the design and implementation details are documented, the system can serve as a teaching platform for understanding timers, interrupts, PWM, and feedback control in a microcontroller laboratory.

Future work includes replacing the IR sensor with a Hall-effect sensor, which is less sensitive to ambient light, incorporating PID control to reduce the steady-state error, and using a faster microcontroller to achieve finer PWM resolution and shorter measurement gate times.

Author Contributions

All authors contributed equally to the design, implementation, experimental testing and writing of this work. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

Data supporting these findings are available within the article or upon request.

References

[1] Krishnan R. Permanent Magnet Synchronous and Brushless DC Motor Drives. Boca Raton, FL, USA: CRC Press; 2010.
[2] Miller TJE. Brushless Permanent-Magnet and Reluctance Motor Drives. Oxford, U.K.: Oxford University Press; 1989.
[3] Mondal R, Mukhopadhyay A, Basak D. "Embedded system of DC motor closed loop speed control based on 8051 microcontroller", Procedia Technology, vol. 10, pp. 840–848, 2013.
[4] Adesina GR, Shoewu OO, Olatinwo SO, Omitola OO. "Development of a Microcontroller-Based Motor Speed Control System Using Intel 8051," Journal of Advancement in Engineering and Technology, vol. 1, no. 2, pp. 1–5, 2014.
[5] Singh S. "Speed Control of Closed Loop DC Motor Using 8051 Microcontroller," International Journal of Emerging Technologies and Innovative Research, vol. 6, no. 2, pp. 424–430, Feb. 2019.
[6] Tiwari G, Singh M, Rathore A, Sharma VP. "DC motor controller by PWM technique," Research & Reviews: Journal of Embedded System & Applications, vol. 7, no. 3, 2019.
[7] Gamazo-Real JC, Vázquez-Sánchez E, Gómez-Gil J. "Position and speed control of brushless DC motors using sensorless techniques and application trends," Sensors, vol. 10, no. 7, pp. 6901–6947, 2010.
[8] Akrami M, Jamshidpour E, Nahid-Mobarakeh B, Pierfederici S, Frick V. "Sensorless Control Methods for BLDC Motor Drives: A Review," IEEE Transactions on Transportation Electrification, vol. 11, no. 1, pp. 135–152, Feb. 2025. https://doi.org/10.1109/TTE.2024.3387371
[9] Mazidi MA, Mazidi JG, McKinlay RD. The 8051 Microcontroller and Embedded Systems. 2nd ed. Upper Saddle River, NJ, USA: Pearson; 2008.
[10] Mohan N, Undeland TM, Robbins WP. Power Electronics: Converters, Applications, and Design. 3rd ed. Hoboken, NJ, USA: Wiley; 2003.
[11] Rashid MH. Power Electronics: Circuits, Devices, and Applications. 4th ed. Upper Saddle River, NJ, USA: Pearson; 2014.

Publisher's Note & Copyright

IRJIST Journals remains neutral regarding jurisdictional claims in published maps and institutional affiliations; the views expressed are solely those of the authors.

© 2026 by the authors. Open access under the CC BY 4.0 license.