International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, sep-Oct-2026, ISSN (Online): 2395-566X

Temperature-Based Automatic Fan Speed Control Using Embedded Microcontrollers: A Review of Sensing, Pwm Drive and Closed-Loop Control Strategies

Chavan Prathmesh Sambhaji[1] Dhakne Vaibhav Balasaheb[2] Giram Devandra Rajandra[3]
Prof. S. D. Chougule[4] Prof. M. S. Nalawade[5] Dr. V. M. Dhede[6]
[1,2,3]Department Of Electronics And Telecommunication Jaihind College Of Engineering Kuran, Pune, Maharashtra, India
[4]Seminar Guide Department Of Electronics And Telecommunication Jaihind College Of Engineering Kuran, Pune, Maharashtra, India;
[5]Seminar Coordinator Department Of Electronics And Telecommunication Jaihind College Of Engineering Kuran, Pune, Maharashtra, India;
[6] Head Of Department Department Of Electronics And Telecommunication Jaihind College Of Engineering Kuran, Pune, Maharashtra, India

Abstract—Electric fans are used for personal comfort and for cooling equipment such as power supplies, boilers, incubators and ovens. A fan that always runs at full speed wastes energy and produces unnecessary noise, while manual speed selection depends on the user noticing a change in temperature. Automatic fan speed control solves this by measuring temperature and adjusting fan speed continuously. This paper reviews the design of such controllers built around low-cost microcontrollers of the AVR family, such as the ATmega8, ATmega168 and ATmega328 used on Arduino boards. The review covers temperature sensing with the LM35 and alternative sensors, analog-to-digital conversion, pulse width modulation (PWM) drive through a transistor stage, display of set and measured values on a 16×2 LCD, and the power supply. Control strategies ranging from on–off and stepped control to proportional, PID and fuzzy logic control are explained and compared. A reference architecture and control flow are presented for a stand-alone controller using a 12 V, 2 A supply, together with design calculations. Applications in air-conditioning, water heaters, incubators and industrial process equipment, as well as limitations and possible improvements such as wireless monitoring, are discussed.

Index Terms—Fan speed control, temperature sensor, LM35, Arduino, ATmega328, pulse width modulation, closed-loop control, embedded system.

I. INTRODUCTION

Temperature control is one of the oldest and most common tasks in engineering. In homes, fans provide comfort; in electronic equipment, they remove heat from processors and power devices; in industry, they help regulate the temperature of ovens, furnaces, heat exchangers and boilers. In each case the cooling required changes with time. Running a fan at constant full speed handles the worst case but wastes electrical energy, shortens fan life and produces noise even when little cooling is required.

An automatic fan speed controller measures the temperature of the space or equipment and sets the fan speed accordingly. Because the measured temperature is fed back to the controller, the arrangement forms a closed-loop feedback control system [1]. Early controllers used analog comparators or thermistor bridges. Modern designs use a small microcontroller, which makes the logic flexible, allows set-points to be changed easily, and supports user-friendly displays [2].

Development boards such as Arduino, built around the ATmega328 microcontroller, have made such systems simple and inexpensive to build [3].

This paper reviews the components and control methods used in microcontroller-based temperature-controlled fans and presents a reference design. Its contributions are:

- A structured review of temperature sensors, microcontroller features, drive circuits and display interfaces used in fan controllers.
- A classification and comparison of fan control strategies.
- A reference hardware architecture, control algorithm and design calculations for a stand-alone controller.
- A discussion of applications, limitations and future improvements.

Section II explains the fundamentals, Section III reviews the components, Section IV reviews control strategies, Section V presents the reference design, Section VI gives a comparison, Section VII discusses applications and challenges, and Section VIII concludes.

© 2026 IJSRET

Ii. Fundamentals

A. Closed-Loop Temperature

Control In a closed-loop system, the controlled variable is measured and compared with a desired value called the set-point. The difference, or error, determines the corrective action [1], [4]. For a cooling fan, the error is the amount by which the measured temperature exceeds the set temperature, and the corrective action is an increase in fan speed. The thermal system responds slowly because of heat capacity, so the controller must avoid frequent switching and large oscillations.

1. Speed Control Of Dc Fans

Small DC fans run from 5 V or 12 V supplies. Their speed depends on the average voltage applied. Changing this average with a linear resistor or regulator wastes power as heat. Pulse width modulation (PWM) instead switches the supply fully on and off at a fixed frequency and changes the fraction of time it is on, called the duty cycle *D*. The average voltage seen by the fan is approximately

$$V_{avg} = D \cdot V_{supply} \quad (1)$$

Because the switching transistor is either fully on or fully off, its power loss is small, which makes PWM efficient [5]. Four-wire PC fans accept a separate PWM control signal and provide a tachometer output for speed measurement [6].

II. REVIEW OF SYSTEM COMPONENTS

1. Temperature Sensors

The LM35 is a precision integrated-circuit sensor whose output voltage is linearly proportional to temperature in degrees Celsius, with a scale factor of 10 mV/°C, so no conversion from Kelvin is needed [7]. It is calibrated at manufacture, draws little current and is simple to connect to a microcontroller analog input. Digital sensors such as the DS18B20 communicate over a single-wire bus and provide a programmable resolution of up to 12 bits, which avoids analog noise and allows several sensors on one line [8]. Thermistors are low in cost but non-linear, and thermocouples cover high temperatures but need amplification and cold-junction compensation [9]. Table I compares these options.

Table I Comparison Of Temperature Sensors For Fan Controllers

Sensor	Output	Advantage	Limitation
LM35 [7]	Analog, 10 mV/°C	Linear, calibrated in °C	Needs ADC; noise on long wires
DS18B20 [8]	Digital 1-Wire	Up to 12-bit, multi-drop	Slower conversion time
Thermistor (NTC)	Resistance	Very low cost, sensitive	Non-linear; needs calibration
Thermocouple	Microvolts	Wide range, robust	Amplifier and compensation needed

2. Microcontroller

The ATmega8, ATmega168 and ATmega328 are 8-bit AVR microcontrollers with on-chip flash memory, a multi-channel 10-bit analog-to-digital converter (ADC) and hardware timers that can generate PWM signals [10]. With a 10-bit ADC and a 5 V reference, the resolution is about 4.9 mV, which corresponds to roughly 0.5 °C for the LM35. The temperature can be computed from the ADC reading *N* as

$$T\,(°C) = (N \times 5000 / 1024) / 10 \quad (2)$$

The Arduino platform provides a boot-loader and libraries that make reading the ADC and writing a PWM value a matter of a few function calls, which shortens development time [3].

3. Driver Stage

A microcontroller pin can supply only a few tens of milliamperes, which is not enough for a 12 V fan. A transistor, either a bipolar junction transistor such as a TIP122 Darlington or a logic-level MOSFET, is therefore used as a switch between the fan and ground, driven by the PWM pin through a base or gate resistor [11]. Because a fan motor is inductive, a freewheeling diode is connected across it to absorb the voltage spike produced at turn-off [5].

4. Display And User Interface

A 16×2 character LCD based on the HD44780 controller is widely used to show the measured temperature, the set temperature and the fan speed level [12]. It can be driven in 4-bit mode to save microcontroller pins. Push buttons or a potentiometer allow the user to change the set-point.

5. Power Supply

© 2026 IJSRET

A regulated 12 V, 2 A adapter or transformer-rectifier circuit powers the fan, while a voltage regulator derives 5 V for the microcontroller, sensor and display. Decoupling capacitors close to the microcontroller reduce noise from fan switching, which otherwise disturbs ADC readings [11].

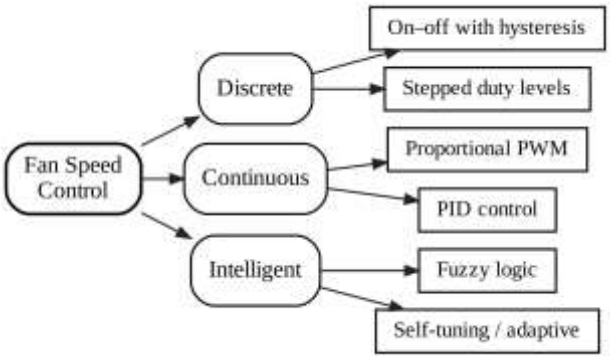


Fig. 1. Classification of fan speed control strategies reviewed in this paper.

III. REVIEW OF CONTROL STRATEGIES

1. On–Off Control

The simplest controller switches the fan fully on when the temperature exceeds the set-point and off when it falls below. To prevent rapid switching near the set-point, a band called hysteresis is added, so the fan switches on at, for example, T_{set} + 1 °C and off at T_{set} − 1 °C [1]. On–off control is easy to implement but causes the temperature to oscillate within the band and gives noisy fan operation.

B. Stepped (Multi-Level) Control

Many practical designs divide the temperature range into bands and assign a fixed duty cycle to each, for example 25%, 50%, 75% and 100%. This gives smoother operation than on–off control and is easy to display as a speed level on the LCD. Its performance depends on how the bands are chosen.

2. Proportional Control

In proportional control, the duty cycle increases linearly with the error between measured and set temperatures, up to a maximum:

$$D = \min(1, \max(0, (T - T_{set}) / (T_{max} - T_{set}))) \quad (3)$$

This provides continuous adjustment without abrupt changes. A small steady-state error remains, which is acceptable for most fan applications.

3.3 Pid Control

A proportional–integral–derivative (PID) controller adds an integral term that removes steady-state error and a derivative term that reacts to the rate of change [13]. PID control is common in industrial temperature regulation, but tuning the three gains requires care, and the slow thermal response makes the derivative term sensitive to sensor noise.

4. Fuzzy Logic Control

Fuzzy logic, introduced by Zadeh, represents variables with linguistic sets such as *cool*, *warm* and *hot* [14]. Mamdani and Assilian showed that such rules can be used to control a physical plant without an exact mathematical model [15]. A fuzzy fan controller may use rules such as "if temperature is hot and rising, then speed is high". It handles non-linear behaviour well but needs more memory and design effort than simple methods.

Reference Design

6. Hardware Architecture

Fig. 2 shows the reference architecture. The LM35 output is connected to an ADC input of the ATmega328. The microcontroller compares the measured temperature with the set-point, computes the duty cycle and outputs a PWM signal to the transistor driver, which switches the 12 V fan. The LCD displays the set and measured temperatures and the speed level. Set-point buttons provide user input.

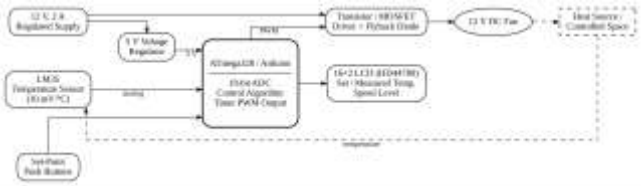


Fig. 2. Block diagram of the microcontroller-based temperature-controlled fan speed controller.

7. Control Algorithm

The control flow is shown in Fig. 3. After initialising the ADC, timer and LCD, the program reads the sensor several times and averages the readings to reduce noise, converts the result to degrees Celsius using (2), and computes the duty cycle with a stepped or proportional rule (3). The PWM output is updated, the display refreshed, and the loop repeats about every 500 ms, which is fast compared with the thermal time constant of most systems.

© 2026 IJSRET

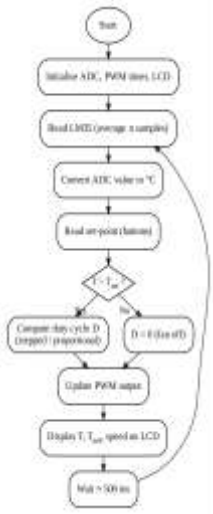


Fig. 3. Control flowchart of the automatic fan speed controller.

8. Design Considerations

The PWM frequency should be chosen with the fan in mind. Low frequencies near the default Arduino value of about 490 Hz can produce audible noise, while four-wire fans are designed for a control frequency of around 25 kHz [6]. The transistor must handle the fan current with margin; a 12 V fan rated at 0.2 A, for instance, is easily driven by a logic-level MOSFET. The sensor should be placed where it represents the temperature to be controlled, and shielded cable should be used for long connections because the LM35 output is a low-level analog signal [7].

IV. COMPARATIVE ANALYSIS

Table II compares the control strategies for a microcontroller-based fan controller.

Table Ii Comparison Of Fan Speed Control Strategies

Strategy	Complexity	Temperature Stability	Noise and Wear	Energy Saving	Typical Use
On–off with hysteresis	Very low	Oscillates in band	High (frequent switching)	Moderate	Simple ventilation
Stepped levels	Low	Moderate	Moderate	Good	Domestic fans, cabinets
Proportional	Low	Good, small offset	Low	Good	Electronics cooling
PID [13]	Medium	Very good	Low	Very good	Process equipment, incubators
Fuzzy logic [15]	High	Very good	Low	Very good	Non-linear thermal systems

V. APPLICATIONS AND CHALLENGES

Temperature-controlled fans are applicable to air-conditioning units, water heaters, ovens, mixers, heat exchangers, furnaces, incubators, thermal baths and veterinary operating tables, as well as to cooling of power supplies, inverters and computers. In process industries, the same principle helps maintain boiler and furnace temperature within safe limits.

Several limitations remain. A single sensor measures temperature at only one point, which may not represent the whole space. Electrical noise from the switching fan can disturb ADC readings unless the layout and filtering are done carefully. Simple controllers cannot detect a stalled fan; using the tachometer output of a four-wire fan allows fault detection [6]. Future improvements include multiple sensors, humidity sensing for comfort control, wireless monitoring through Wi-Fi or Bluetooth modules, data logging, and self-tuning PID or fuzzy algorithms that adapt to the thermal behaviour of the equipment [16].

VI. CONCLUSION

This paper has reviewed the design of microcontroller-based automatic fan speed controllers. Temperature sensing with the LM35 or digital sensors, ADC conversion in an AVR microcontroller, PWM drive through a transistor stage, and display on a 16×2 LCD together form a compact closed-loop system.

Control strategies from on–off to PID and fuzzy logic offer increasing smoothness and stability at increasing complexity. The reference design presented here, built on an Arduino-class ATmega328 with a regulated 12 V, 2 A supply, is low in cost, reliable and energy-saving, and can be adapted to

© 2026 IJSRET

domestic and industrial applications such as incubators and boiler temperature control.

REFERENCES

1. K. Ogata, Modern Control Engineering, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2010.
2. M. A. Mazidi, S. Naimi, and S. Naimi, The AVR Microcontroller and Embedded Systems: Using Assembly and C. Upper Saddle River, NJ, USA: Pearson, 2011.
3. M. Banzi and M. Shiloh, Getting Started with Arduino, 3rd ed. Sebastopol, CA, USA: Maker Media, 2014.
4. N. S. Nise, Control Systems Engineering, 7th ed. Hoboken, NJ, USA: Wiley, 2015.
5. M. H. Rashid, Power Electronics: Devices, Circuits, and Applications, 4th ed. Harlow, U.K.: Pearson, 2014.
6. Intel Corporation, "4-Wire Pulse Width Modulation (PWM) Controlled Fans," Specification, Rev. 1.3, Sep. 2005.
7. Texas Instruments, "LM35 Precision Centigrade Temperature Sensors," Datasheet SNIS159, Dallas, TX, USA.
8. Maxim Integrated, "DS18B20 Programmable Resolution 1-Wire Digital Thermometer," Datasheet, San Jose, CA, USA, 2019.
9. J. Fraden, Handbook of Modern Sensors: Physics, Designs, and Applications, 5th ed. Cham, Switzerland: Springer, 2016.
10. Microchip Technology Inc., "ATmega48A/PA/88A/PA/168A/PA/328/P megaAVR Data Sheet," Chandler, AZ, USA, 2018.
11. P. Horowitz and W. Hill, The Art of Electronics, 3rd ed. Cambridge, U.K.: Cambridge University Press, 2015.
12. Hitachi, Ltd., "HD44780U (LCD-II) Dot Matrix Liquid Crystal Display Controller/Driver," Datasheet, 1998.
13. K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning, 2nd ed. Research Triangle Park, NC, USA: Instrument Society of America, 1995.
14. L. A. Zadeh, "Fuzzy sets," Information and Control, vol. 8, no. 3, pp. 338–353, 1965.
15. E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1–13, 1975.
16. F. Vahid and T. Givargis, Embedded System Design: A Unified Hardware/Software Introduction. New York, NY, USA: Wiley, 2002.

© 2026 IJSRET