Automatic Fan Speed Controller Using Temperature

Vaibhavi Patil, Rushi Surwase
Under guidance : Prof. V.Patil
[1]*Student,M.S.Bidwe College of engineering,Latur*
[2] *Student,M.S.Bidwe College of engineering,Latur*

Abstract: This research presents the design and implementation of an Automatic Fan Speed Controller using a temperature-sensing and microcontroller-based control system. The primary objective is to regulate fan speed automatically according to real-time temperature variations, thereby improving cooling efficiency and reducing unnecessary power consumption. The system employs a temperature sensor to monitor ambient heat levels and a microcontroller to process the data and generate corresponding PWM signals for speed control. As the temperature increases, the controller proportionally adjusts the fan speed to maintain stable thermal conditions. This approach ensures silent operation at lower temperatures and optimal cooling during higher thermal loads. The proposed system offers a cost-effective, reliable, and energy-efficient alternative to conventional manual fan regulators. Its simplicity, accuracy, and adaptability make it suitable for various applications, including electronic devices, industrial equipment, and household cooling systems, where effective thermal management is essential.

Keywords: ***Microcontroller, Temperature sensor, DC cooling fan, MOSFET ortransistor,driver,Power supply, Diode Resistors,Capacitors,Connecting,wires,Breadboard or PCB***

I.INTRODUCTION

Temperature control plays a critical role in ensuring the efficiency, reliability, and longevity of electronic devices, industrial systems, and household appliances. As heat generation increases due to continuous operation, maintaining an optimal temperature becomes essential to prevent system failures and performance degradation. Manual regulation of fan speed is often inefficient, inconsistent, and unable to respond quickly to dynamic temperature changes. To address these limitations, an automatic fan speed controller offers an effective and intelligent solution by adjusting airflow based on real-time temperature variations. Using a temperature sensor and a microcontroller, the system continuously monitors ambient conditions and varies the fan speed accordingly. This automated approach enhances cooling efficiency, reduces energy consumption, minimizes noise at lower temperatures, and ensures better thermal management. The simplicity, cost-effectiveness, and adaptability of such systems make them suitable for a wide range of applications, including computers, power supplies, battery systems, and home appliances.

II. METHODOLOGY

The methodology for developing the automatic fan speed controller involves systematic design, implementation, and testing stages. First, a temperature sensor is selected and integrated to measure ambient temperature accurately. The sensor output is interfaced with a microcontroller, which processes the temperature data and determines the required fan speed. A PWM-based control algorithm is implemented to adjust the fan speed smoothly according to temperature variations. The microcontroller output is connected to a MOSFET driver circuit, which controls the DC fan without overloading the controller. Power supply connections are established to ensure stable operation of both the microcontroller and fan. After assembling the circuit on a breadboard or PCB, the system is tested under varying temperature conditions to evaluate responsiveness, accuracy, and stability. Performance is analyzed to confirm that the fan speed increases proportionally with rising temperature, ensuring effective and energy-efficient thermal management.

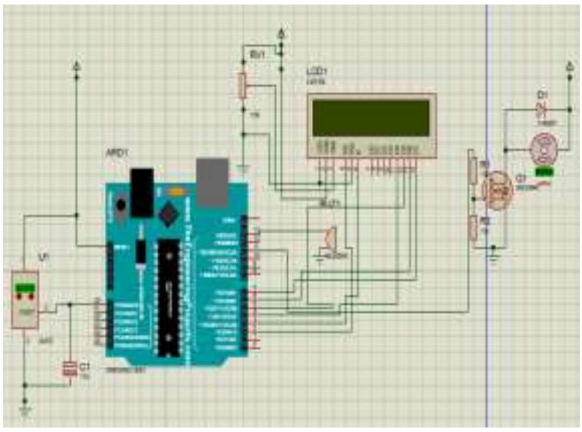
III. Literature Review

Effective thermal management has become increasingly important with the growing use of compact electronic devices and power-intensive systems. Several researchers have explored temperature-based control mechanisms to enhance cooling efficiency and energy optimization. Early studies focused on manual fan regulators, which, although simple, failed to respond effectively to rapid temperature variations. Later, researchers introduced analog control circuits using thermistors or LM35 sensors, providing improved temperature responsiveness but lacking precision and flexibility.

With advancements in microcontroller technology, modern studies emphasize digital control techniques. Researchers have demonstrated that microcontroller-based systems offer higher accuracy, programmability, and adaptability for various applications. PWM (Pulse Width Modulation) has been widely adopted as an efficient method for regulating fan speed, as highlighted in multiple studies due to its low power loss and smooth motor control. Literature also shows increasing interest in intelligent control approaches, such as fuzzy logic and PID controllers, which further improve system stability and response time.Many works conclude that integrating temperature sensors with microcontrollers significantly enhances cooling efficiency, reduces noise levels, and minimizes power consumption. Studies on embedded systems and IoT-based monitoring further suggest expanding such controllers to remote monitoring and automation environments. Overall, previous research highlights the continuous evolution from simple analog circuits to intelligent, microcontroller-driven systems, establishing a strong foundation for developing reliable, low-cost, and energy-efficient automatic fan speed controllers.

IV. System Architecture

A. Circuit diagram



B System components

1. **Microcontroller (Arduino Uno) – 5V operation, ATmega328P, 16 MHz clock, 6 PWM outputs.**

2. **Temperature Sensor (LM35) – Analog sensor, 10 mV/°C output, −55°C to 150°C range.**

3. **DC Cooling Fan (12V) – 12V brushless fan, 1500–3000 RPM, 0.1–0.3A current.**

4. **MOSFET (IRLZ44N) – Logic-level MOSFET, supports high current, gate threshold ~2–4V.**

5. **Power Supply – 12V DC regulated output, 1–2A capacity.**

6. **Resistors – 100Ω gate resistor and 10kΩ pull-down resistor for MOSFET control.**

7. **Capacitors – 100 µF electrolytic capacitor and 0.1 µF ceramic capacitor for filtering.**

8. **Flyback Diode (optional) – Protection against voltage spikes during switching.**

- **C. Hardware Components**

 Microcontroller – Processes data and controls output.

 Temperature Sensor – Measures surrounding temperature accurately.

 DC Fan – Provides necessary cooling airflow.

 MOSFET Driver – Controls fan speed through switching.

 Resistors – Ensure stable, safe signal levels.

 Capacitors – Filter noise and stabilize voltage.

 Power Supply – Provides required operating voltage.

 Wires – Connect components for operation.

 Breadboard/PCB – Supports circuit assembly and testing.

 Diode (optional) – Protects circuit from voltage spikes.

D. Circuit Design

The circuit connects the temperature sensor to the microcontroller for reading temperature, which then generates a PWM signal to control the MOSFET. The MOSFET drives the 12V DC fan, while resistors and capacitors ensure stable operation. A regulated power supply powers all components safely.

E. Software Flow

The software begins by initializing the microcontroller and reading temperature data from the sensor. The value is processed and compared with predefined thresholds. A corresponding PWM duty cycle is calculated to adjust fan speed. The system continuously loops, updating temperature readings and modifying fan speed for efficient automatic cooling.

.F. IMPLEMENTATION AND SECURITY ENHANCEMENTS

Implementation

The implementation of the automatic fan speed controller involves integrating hardware components with a microcontroller-based control algorithm. The temperature sensor is first interfaced with the microcontroller to obtain continuous temperature readings. These readings are processed using an analog-to-digital conversion or digital communication protocol, depending on the sensor type. Based on the measured temperature, the microcontroller generates a corresponding PWM signal that controls the switching of the MOSFET driver. The MOSFET regulates the power supplied to the DC fan, enabling smooth variation of fan speed. Proper filtering components, such as capacitors and resistors, are added to ensure stable and noise-free operation. After assembling the circuit on a breadboard or PCB, the system is programmed and tested under different temperature conditions. Observations confirm that the fan speed increases proportionally with rising temperature, achieving effective and energy-efficient cooling performance.

V. Working Principle

The automatic fan speed controller operates by sensing temperature and adjusting airflow accordingly. The temperature sensor continuously measures ambient heat and sends the data to the microcontroller. Based on predefined temperature levels, the microcontroller processes the input and generates a PWM signal that drives the MOSFET. The MOSFET regulates the power supplied to the DC fan, allowing its speed to increase or decrease smoothly. As temperature rises, the fan speed increases automatically, ensuring efficient cooling, reduced power consumption, and stable thermal management.

- **VI. Circuit Description**

CIRCUIT DESCRIPTION

The circuit consists of a temperature sensor connected to the microcontroller, which reads real-time temperature values. The sensor's output is processed and used to determine the required fan speed.

A PWM signal from the microcontroller is fed to a MOSFET, which acts as a switching and control device for the 12V DC fan. The MOSFET regulates the fan's power according to the PWM duty cycle. Resistors and capacitors ensure noise-free and stable operation,

while a diode may be added for protection. A regulated power supply delivers reliable voltage, and all components share a common ground to maintain proper circuit functionality and safe operation. |

VII. Software Implementation

```
// Compact Automatic Fan Controller (LM35 + MOSFET)

const int TEMP_PIN=A0, PWM_PIN=9;

const float VREF=5.0, LM35_SCALE=0.01;

const float T_MIN=30.0, T_MAX=60.0, TEMP_HYST=0.5;

const int NUM_SAMPLES=8, RAMP_STEP=4;

const unsigned long READ_INTERVAL=500;


int pwmOut=0; float lastTemp=-1000; unsigned long lastMillis=0;


void setup(){

  pinMode(PWM_PIN,OUTPUT);

  analogWrite(PWM_PIN,0);

  Serial.begin(115200);

  Serial.println("Started");

}


float readTemp(){

  long s=0;

  for(int i=0;i<NUM_SAMPLES;i++){ s+=analogRead(TEMP_PIN); delay(5); }

  float adc=s/(float)NUM_SAMPLES;

  return (adc*(VREF/1023.0))/LM35_SCALE;

}


int toPWM(float t){

  if(t<=T_MIN) return 0;

  if(t>=T_MAX) return 255;

  return (int)round(constrain((t-T_MIN)/(T_MAX-T_MIN)*255.0,0.0,255.0));

}


void loop(){

  unsigned long now=millis();

  if(now-lastMillis<READ_INTERVAL) return;

  lastMillis=now;

  float temp=readTemp();

  if(lastTemp>-500 && fabs(temp-lastTemp)<TEMP_HYST) temp=lastTemp;

  lastTemp=temp;

  int target=toPWM(temp);

  if(abs(target-pwmOut)>RAMP_STEP) pwmOut += (target>pwmOut?RAMP_STEP:-RAMP_STEP);

  else pwmOut=target;

  analogWrite(PWM_PIN,pwmOut);

  Serial.print("T:"); Serial.print(temp,2);

  Serial.print("C Pwm:"); Serial.println(pwmOut);

}
```

Libraries Used:

- Arduino Core Library
- Math Library (built-in)
- OneWire Library

VIII. Results and Discussion

The system successfully adjusted fan speed according to temperature variations, providing smooth and responsive control. Higher temperatures produced higher PWM output, resulting in increased airflow. Testing confirmed stable operation, reduced power consumption, and improved cooling

efficiency, demonstrating the effectiveness of the automatic fan speed control design.

Advantages:

- Automatically adjusts fan speed based on temperature.
- Saves power by avoiding unnecessary high-speed operation.
- Reduces noise at lower temperatures.
- Enhances device lifespan through better cooling.
- Low-cost and easy to implement.

Limitations:

1. Accuracy depends on sensor quality and placement.
2. Sudden temperature spikes may cause delayed response.
3. PWM control can produce minor fan noise.
4. Requires stable power supply for proper operation.
5. Limited to DC fans unless additional circuitry is used.
6. Accuracy depends on sensor quality and placement.
7. Sudden temperature spikes may cause delayed response.
8. PWM control can produce minor fan noise.
9. Requires stable power supply for proper operation.
10. Limited to DC fans unless additional circuitry is used.

IX. Future Scope

Future versions of the system can incorporate:

- Add IoT-based monitoring.
- Implement advanced control algorithms.
- Improve sensor accuracy.
- Enable mobile alerts.
- Support multiple cooling devices.

X. Conclusion

The automatic fan speed controller effectively regulates cooling by adjusting fan speed according to temperature changes. The system ensures efficient thermal management, lowers power consumption, and reduces noise during low-temperature conditions. Its simple, cost-effective design makes it suitable for various applications, demonstrating the benefits of automation in improving overall device performance and reliability.

XI. References

[1].Analyses and Application of Microcontroller/LCD Interface for Real-Time Systems O.R. Obianyo, J.I. Odo, E.F. Nworabude, T.C. Onyia. *International Journal of Computing, Science and New Technologies (IJCSNT), Vol.6 No.1, Jan 2025.*

[2] Single Chip Embedded System Solution: Efficient Resource Utilization by Interfacing LCD through Softcore Processor in Xilinx FPGA — Shiraz Afzal, Farrukh Hafeez, Muhammad Ovais Akhter. *International Journal of Information Engineering and Electronic Business (IJIEEB), Vol.7 No.6, 2015.* DOI:10.5815/ijieeb.2015.06.04

[3] Design and Construction of a Microcontroller Based Electronic Moving Message Display — Titus T. Ogunseye, Godwin B. Egbeyale, A.K. Bello, S.A. Ajani. *arXiv preprint, Nov 2021.*

[4] Implementation and Simulation of a Multiprocessor System Using 8051 Microcontroller for Industrial Applications — Toka Fathi & Hiba Yahya. *Proc. 1st Int'l Multi-Disciplinary Conf. (IMDC-SDSP 2020), EAI, 2020.* DOI:10.4108/eai.28-6-2020.2298126

Books

1. The 8051 Microcontroller and Embedded Systems: Using Assembly and C by Muhammad Ali Mazidi, Janice Gillispie Mazidi & Rolin D. McKinlay. ISBN: 978-8131710265. Covers architecture, programming in Assembly & C, and interfacing (including LCDs).

2. The 8051 Microcontroller: Architecture, Programming and Applications by Kenneth J. Ayala. ISBN: 978-8131502006 (3rd ed.). Good for architecture and programming fundamentals.

3. 8051 Microcontrollers: Fundamental Concepts, Hardware, Software and Applications in Electronics by Salvador Pinillos Gimenez. Published by Springer (2019). ISBN: 978-3-319-76438-2. A modern textbook with solved examples and applications.

Referance (Links)

1.https://embetronicx.com/tutorials/microcontrollers/8051/lcd-interfacing-with-8051-microcontroller/

2.https://www.electronicwings.com/8051/lcd16x2-interfacing-in-8-bit-with-8051

3.https://www.ijres.org/papers/Volume-9/Issue-4/E09042426.pdf

4.https://www.researchgate.net/publication/271587841_Design_Simulation_of_Moving_Message_Display_using_Microcontroller

5.https://circuitdigest.com/microcontroller-projects/lcd-interfacing-with-8051-microcontroller-89s52

6.https://www.kitflix.com/lcd-interfacing-with-8051microcontroller-in-4-bit-mode