**IoT Based Smart Mirror**

Mr D krishna, Assistant.Professor  
CSE(IoT)  
ACE Engineering CollegeHyderabad, India  
[xxxxxx@gmail.com](mailto:xxxxxx@gmail.com)

R.vijayeeshwar,Student   
CSE(IoT)  
ACE Engineering CollegeHyderabad, India  
[vijayeeshwar123@gmail.comJ.Niharika](mailto:vijayeeshwar123@gmail.comJ.Niharika) ,Student  
CSE(IoT)  
ACE Engineering College  
Hyderabad, India  
niharika[@gmail.com](mailto:eshwarsatish9@gmail.com)

k.Jaydeep,Student  
CSE(IoT)  
ACE Engineering CollegeHyderabad, India  
[jaydeep@gmail.com](mailto:jaydeep@gmail.com)

**Abstract:**

Introducing Automatic Fan Speed Controller,this project is a standalone automatic fan speed controller that controls the speed of an electric fan according to our requirement based on temperature .In these project we use breadboard, LCD display, I2c maodule, Aurdino Uno board, jumper wires, temperature sensor, motodriver.The sensed temperature and fan speed level values are simultaneously displayed on the LCD panel. A temperature sensor has been used to measure the temperature of the room and the speed of the fan is varied according to the room temperature.

The temperature-based fan speed control system can be done by using an electronic circuit using an Arduino board.Now Arduino board is very progressive among all electronic circuits, thus we employed Arduino board for fan speed control.

The proposed system is designed to detect the temperature of the room and send that information to the Arduino board. Then the Arduino board executes the contrast of current temperature and set temperature based on the inbuilt program of the Arduino.

I**. INTRODUCTION**

In various electronic and mechanical systems, controlling the temperature is crucial for ensuring optimal performance and longevity. One effective method to achieve this is through an automatic fan speed controller based on temperature. This system intelligently adjusts the speed of a fan according to the temperature of its surroundings or the device it is cooling. By doing so, it maintains a stable and safe operating temperature, thereby enhancing efficiency and preventing overheating.

Automatic fan speed controllers find applications in numerous fields, including computer cooling systems, home appliances, industrial machinery, and HVAC (Heating, Ventilation, and Air Conditioning) units.

The core principle involves using sensors to monitor temperature levels and a control mechanism to regulate the fan's speed accordingly. This not only ensures energy efficiency but also reduces noise levels by operating the fan only when necessary.

In this project/development (if this is part of a specific project or study), the focus will be on designing a reliable and responsive automatic fan speed controller. This will involve selecting appropriate sensors, implementing a control algorithm, and possibly integrating feedback mechanisms to fine-tune performance.

The goal is to create a system that adapts seamlessly to varying thermal conditions, providing consistent cooling or ventilation as needed.

By the end of this study/project, we aim to demonstrate the effectiveness of an automatic fan speed controller in maintaining optimal operating conditions, contributing to energy savings, equipment longevity, and improved user comfort

.

In many electronic and mechanical systems, managing heat dissipation is critical for maintaining efficiency and preventing component damage. An automatic fan speed controller based on temperature offers an intelligent solution to this challenge. By continuously monitoring ambient or device-specific temperatures, this system adjusts the speed of the cooling fan accordingly. This dynamic control not only ensures optimal thermal conditions but also minimizes energy consumption by running the fan only when necessary.

This project aims to explore the design and implementation of such a controller, focusing on sensor selection, control algorithms, and integration into practical applications. The ultimate objective is to create a reliable system that enhances operational stability, prolongs equipment lifespan, and reduces overall energy costs.

This introduction sets the stage for discussing the significance and objectives of implementing an automatic fan speed controller based on temperature. It provides a clear overview of the system's purpose and benefits, guiding the reader towards understanding the importance of such technology in various industries and applications.

**II. LITERATURE SURVEY**

A literature survey on automatic fan speed controllers based on temperature reveals a diverse range of approaches, technologies, and applications across various fields. Here’s a concise overview based on existing research and developments:

1. **Sensor Technologies**: Various temperature sensing technologies are utilized in automatic fan speed controllers. Common sensors include thermistors, which offer cost-effective temperature measurement with good accuracy and response time. Additionally, digital temperature sensors like DS18B20 and LM35 provide precise digital outputs, suitable for interfacing with microcontrollers.
2. **Control Algorithms**: Several control algorithms are employed to adjust fan speed based on temperature readings. Proportional-Integral-Derivative (PID) control is widely adopted due to its ability to respond smoothly to changes in temperature and maintain stability. Adaptive algorithms, such as fuzzy logic and neural networks, are also explored to optimize fan speed adjustments under varying environmental conditions.
3. **Hardware Implementations**: The implementation of automatic fan speed controllers typically involves microcontrollers or dedicated fan control ICs. Microcontrollers such as Arduino and Raspberry Pi are popular choices due to their versatility and ease of integration with sensors and actuators. Fan control ICs like the LM96000 series provide dedicated hardware solutions for precise fan speed regulation based on temperature inputs.
4. **Applications**: Automatic fan speed controllers find applications in diverse industries. In computing, these controllers are crucial for maintaining optimal operating temperatures in CPUs and GPUs. They are also used in industrial equipment, HVAC systems, and home appliances to improve energy efficiency and equipment longevity. Automotive applications focus on cooling electronics and batteries in electric vehicles to enhance performance and safety.
5. **Performance Evaluation**: Research often includes performance evaluations to validate the effectiveness of automatic fan speed controllers. Parameters such as temperature stabilization time, energy savings, noise reduction, and overall system reliability are key metrics. Comparative studies between different control strategies and hardware implementations provide insights into selecting the most suitable approach for specific applications.
6. **Challenges and Future Directions**: Challenges in automatic fan speed controllers include robust sensor calibration, real-time responsiveness, and integration with existing systems. Future research directions emphasize advancements in sensor technology for higher accuracy and lower power consumption, as well as the development of intelligent algorithms for adaptive fan speed control in dynamic environments.

Overall, the literature survey highlights the evolution of automatic fan speed controllers from basic temperature sensing to sophisticated control algorithms and integrated solutions. Ongoing research aims to further enhance efficiency, reliability, and applicability across a wide range of thermal management scenarios.

**III. PROBLEM STATEMENT**

In many electronic devices and systems, effective thermal management is essential for maintaining optimal performance and preventing overheating. Traditional cooling solutions often operate fans at fixed speeds or rely on manual adjustments, leading to inefficiencies in energy usage and potentially inadequate cooling under varying load conditions.

The objective of this project is to design and implement an automatic fan speed controller that adjusts the speed of a cooling fan based on real-time temperature measurements. This controller aims to achieve the following goals:

1. **Dynamic Thermal Regulation**: Automatically adjust fan speed to maintain a stable and safe operating temperature range, optimizing cooling efficiency and system reliability.
2. **Energy Efficiency**: Minimize energy consumption by running the fan only when necessary, aligning operational speed with actual thermal demands.
3. **User-Friendly Integration**: Develop a user-friendly interface or control mechanism that allows for easy monitoring of temperature levels and adjustment of fan settings.
4. **Scalability and Adaptability**: Design the controller to be scalable for different types of cooling systems and adaptable to various environmental conditions and load variations.
5. **Performance Evaluation**: Conduct thorough testing and evaluation to validate the controller's effectiveness in enhancing thermal management, energy efficiency, and overall system performance.

By addressing these objectives, this project aims to contribute to the advancement of automated cooling solutions in electronic and mechanical applications, improving reliability, energy efficiency, and user convenience.

**IV. EXISTING SYSTEM**

* A fan is a powered machine used to create a flow of air.
* A fan consists of a rotating arrangement of vanes or blades, generally made of wood, plastic, or metal, which act on the air.
* The rotating assembly of blades and hub is known as an impeller, rotor, or runner.
* The speed is usually changed by us manually.

**V. PROPOSED SYSYTEM**

* The proposed system is designed to detect the temperature of the room and send that information to the Arduino board.
* in this project we will provide two functions. First one is, control the turned on/off the fan with respect to the temperature rather than the use of manual switching system.
* Other function is control the speed of a fan with respect of temperature set. In this project Arduino Uno forms the processing part.
* Which senses the temperature with the use of LM35(Temperature sensor).
* Arduino Uno senses the temperature and control the speed with the set temperature.

**VI. HARDWARE AND SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

* LCD display.
* I2c maodule.
* Aurdino Uno board.
* jumper wires.
* temperature sensor.
* motodriver.

**SOFTWARE REQUIREMENTS:**

Operating system- Windows 7 or higher

Software - Arduino IDE

Programming Language – Arduino programming language

**VII. MODULES**

To implement this project, we have designed following modules.

Designing an Automatic Fan Speed Controller based on temperature involves breaking down the system into modules that handle specific functionalities. Below is a conceptual breakdown of modules for such a system:

**1. Temperature Sensing Module:**

- Responsible for reading the current temperature of the environment.

- Provides an interface to obtain temperature readings periodically or on-demand.

**2. Control Algorithm Module:**

- Implements the logic to determine the appropriate fan speed based on the temperature readings.

- May include algorithms for proportional control, PID (Proportional-Integral-Derivative) control, or other control strategies.

**3. Fan Control Module:**

- Interfaces with the physical fan unit to adjust its speed.

- Sends commands to set the fan speed based on the decisions made by the Control Algorithm Module.

**4. User Interface Module:**

- Provides a means for users to interact with the system.

- Allows users to start/stop the automatic fan control, view current temperature, and adjust settings if applicable.

**5. Logging and Monitoring Module:**

- Records temperature readings, fan speed adjustments, and system events for monitoring and analysis purposes.

- May include logging to a database or file system, and real-time monitoring capabilities.

**6. Configuration Module:**

- Stores and manages configuration parameters such as temperature thresholds, fan speed ranges, and control settings.

- Allows for easy adjustment of system parameters without modifying the core logic.

**7. Safety and Fault Handling Module:**

- Monitors system health and responds to abnormal conditions such as sensor failures, communication errors, or overheating.

- Implements safety mechanisms to prevent fan operation in extreme conditions or when critical failures occur.

**8. Communication Module (Optional):**

- Facilitates communication with external systems or remote monitoring applications.

- Enables integration with home automation systems, IoT platforms, or centralized monitoring solutions.

These modules work together to create a robust Automatic Fan Speed Controller system that adjusts fan speeds based on environmental conditions. Depending on the complexity and specific requirements of the project, these modules can be further refined or combined as needed. Each module encapsulates specific responsibilities, promoting modularity, maintainability, and scalability of the overall system.

**VIII SAMPLE CODE:**

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#define DHTPIN 7

#define DHTTYPE DHT11

#define MOTOR\_PIN\_ENA 9

#define MOTOR\_PIN\_IN1 10

#define MOTOR\_PIN\_IN2 11

#define TEMPERATURE\_THRESHOLD 29

#define TEMPERATURE\_THRESHOLD1 32

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal\_I2C lcd(0x3F, 16, 2);

void setup() {

Serial.begin(9600);

dht.begin();

Wire.begin();

lcd.init();

lcd.backlight();

lcd.begin(16, 2);

lcd.setCursor(0, 0);

lcd.print("Temperature:");

}

void loop() {

delay(2000);

float temperature = dht.readTemperature();

if (isnan(temperature)) {

Serial.println("Failed to read temperature from DHT sensor!");

return;

}

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" °C");

else {

// Decrease motor speed

analogWrite(MOTOR\_PIN\_ENA, 45);

digitalWrite(MOTOR\_PIN\_IN1, HIGH);

digitalWrite(MOTOR\_PIN\_IN2, LOW);

lcd.setCursor(0, 1);

lcd.print("Motor Speed: Low");

}

lcd.setCursor(12, 0;

lcd.print(temperature);

}

**IX. OUTPUT SCREENS**





**X . PROJECT DEPLOYMENT**

Deploying an automatic fan speed controller involves transitioning from an experimental setup to a practical application in a real-world environment. Here's a step-by-step guide on how to deploy such a system effectively:

**1. Finalize System Design and Components:**

* **Review and Optimize:** Ensure the automatic fan speed controller design meets the specific requirements of the deployment environment, considering factors such as room size, ventilation, and cooling needs.
* **Component Selection:** Choose reliable components based on the results of your experimental setup, ensuring they can withstand continuous operation and environmental conditions.

**2. Installation and Setup:**

* **Mounting and Wiring:** Install the temperature sensor in an optimal location where it can accurately measure ambient temperature. Ensure proper mounting of fans for efficient airflow and noise control.
* **Connectivity:** Establish connections between the temperature sensor, microcontroller, fan(s), and any additional components like motor drivers or power supplies. Double-check wiring for accuracy and secure connections.

**3. Programming and Calibration:**

* **Code Deployment:** Upload the finalized control algorithms and logic onto the microcontroller. Test the program to verify functionality and responsiveness in controlling fan speeds based on temperature inputs.
* **Calibration:** Fine-tune the system by calibrating temperature readings and fan speed adjustments to ensure accurate performance. Adjust thresholds or PID parameters as needed to optimize energy efficiency and comfort.

**4. Testing and Validation:**

* **Initial Testing:** Conduct initial tests to verify that the system operates correctly under normal operating conditions. Monitor temperature readings and fan speeds to confirm accurate control and responsiveness.
* **Stress Testing:** Perform stress tests to evaluate how the system handles extreme temperature variations or load conditions. Measure energy consumption and noise levels at different fan speeds to validate performance metrics.

**5. Integration and Compatibility:**

* **Smart Home Integration (if applicable):** If integrating with smart home systems, ensure compatibility and test remote control features via mobile apps or voice assistants.
* **User Interface:** Set up user interfaces such as displays or control panels for easy monitoring and adjustment of settings.

**6. Documentation and Training:**

* **Documentation:** Create comprehensive documentation that includes system diagrams, wiring schematics, programming code, calibration procedures, and troubleshooting steps.
* **Training:** Provide training for users or maintenance personnel on operating the automatic fan speed controller, understanding error codes, and performing basic maintenance tasks.

**7. Monitoring and Maintenance:**

* **Monitoring:** Implement a monitoring plan to regularly check system performance, including temperature readings, fan speeds, and any error logs.
* **Maintenance:** Establish a maintenance schedule for cleaning fans, checking wiring connections, and updating software as needed to ensure continued optimal operation.

**8. Feedback and Continuous Improvement:**

* **Feedback:** Gather feedback from users to identify any issues or areas for improvement in system performance or usability.
* **Iterative Improvement:** Use feedback and monitoring data to make iterative improvements to the system design, control algorithms, or user interfaces to enhance functionality and user satisfaction.

By following these steps, you can effectively deploy an automatic fan speed controller, ensuring reliable operation and optimized performance in real-world applications. This approach helps in maximizing energy efficiency, maintaining comfort levels, and extending the lifespan of HVAC equipment.

**XI.INTEGRATION AND EXPERIMEMTAL RESULTS**

Integrating and setting up an experimental system for an automatic fan speed controller involves several steps to ensure proper functionality and testing. Here's a structured approach to integrating and setting up such a system:

**Integration Steps:**

1. **Selection of Components:**
   * Choose appropriate components such as a microcontroller (e.g., Arduino, Raspberry Pi), temperature sensor (e.g., LM35, DS18B20), fan(s), and motor driver (if needed).
2. **Circuit Design:**
   * Design the circuit layout considering the connection between the microcontroller, temperature sensor, motor driver (if applicable), and the fan(s).
   * Ensure proper power supply and grounding to avoid electrical issues.
3. **Programming the Microcontroller:**
   * Write or modify the code for the microcontroller to read temperature data from the sensor and control fan speeds accordingly.
   * Implement control algorithms like threshold-based control or PID control for more advanced systems.
4. **Integration of Sensors and Actuators:**
   * Connect the temperature sensor to the microcontroller and verify that it accurately reads ambient temperature.
   * Ensure the fan(s) are correctly connected to the motor driver (if used) and can be controlled via the microcontroller.
5. **Testing and Calibration:**
   * Test the integrated system in a controlled environment to verify temperature sensing accuracy and fan speed control.
   * Calibrate the system if necessary to ensure accurate temperature readings and smooth fan speed adjustments.

**Experimental Setup:**

1. **Environmental Setup:**
   * Place the temperature sensor in a location representative of where the system will be used (e.g., near the cooling area).
   * Ensure adequate ventilation around the fan(s) to simulate real-world conditions.
2. **Data Logging and Monitoring:**
   * Implement data logging to record temperature readings and corresponding fan speeds over time.
   * Use serial communication or a display to monitor real-time temperature and fan speed adjustments.
3. **Performance Evaluation:**
   * Conduct experiments to evaluate the system's performance under various temperature conditions.
   * Measure fan speed response times, energy consumption, and noise levels at different fan speeds.
4. **Integration with External Systems (Optional):**
   * If integrating with smart home systems or external controllers, ensure compatibility and test remote control functionalities.
5. **Safety Precautions:**
   * Implement safety features such as temperature limits or emergency shutdown procedures to protect against overheating or malfunction.

**Documentation and Analysis:**

1. **Documentation:**
   * Document the setup process, circuit diagrams, code explanations, and calibration procedures for future reference.
   * Include detailed notes on component specifications, wiring details, and any troubleshooting steps taken.
2. **Data Analysis:**
   * Analyze collected data to assess the system's performance metrics such as temperature accuracy, energy efficiency, and user satisfaction.
   * Identify areas for improvement based on experimental results and user feedback.
3. **Iterative Improvement:**
   * Based on the analysis, make necessary adjustments to the system design, code, or components to enhance performance and usability.
   * Conduct additional tests or experiments to validate improvements and ensure reliability.

By following these steps, you can effectively integrate and set up an experimental system for an automatic fan speed controller, enabling comprehensive testing and evaluation to refine the system for practical applications.

**XII . FUTURE ENHANCEMENTS**

Enhancing an automatic fan speed controller based on temperature can involve several improvements to its functionality and efficiency. Here are some future enhancements you could consider:

1. **Advanced Temperature Sensing**: Upgrade the temperature sensor to a more precise and responsive model. For example, using digital temperature sensors like DS18B20 or LM35 can provide accurate readings, which in turn can improve the fan speed control accuracy.
2. **PID Control**: Implement a PID (Proportional-Integral-Derivative) control algorithm instead of simple threshold-based control. PID control can dynamically adjust the fan speed based on how close the temperature is to the setpoint, providing smoother and more responsive control.
3. **Variable Fan Speeds**: Instead of discrete speed levels, implement a system that allows for continuous adjustment of fan speed. This can be achieved using PWM (Pulse Width Modulation) techniques to control the fan motor speed.
4. **Integration with Smart Home Systems**: Enable integration with smart home platforms like Alexa, Google Home, or Apple HomeKit. This allows users to control the fan speed remotely or set up automation based on other conditions (time of day, occupancy, etc.).
5. **Feedback Mechanism**: Incorporate feedback mechanisms to verify the effectiveness of the fan speed control. This could involve monitoring the actual temperature drop or rise after fan speed adjustments and using that data to fine-tune the control algorithm.
6. **Energy Efficiency Considerations**: Introduce energy-saving features such as automatically reducing fan speed when the temperature stabilizes within a comfortable range, or implementing a sleep mode when the fan is not needed for extended periods.
7. **User Interface Improvements**: Enhance the user interface by adding a display to show current temperature, setpoints, and fan speed. Provide intuitive controls such as touch-sensitive buttons or a mobile app interface for easier adjustments.
8. **Noise Reduction**: Implement algorithms to minimize fan noise while maintaining effective cooling. This could involve optimizing fan speed profiles or using quieter fan models.
9. **Fault Detection and Diagnostics**: Include diagnostic features to detect fan malfunctions (e.g., motor failure, blocked airflow) and provide alerts or automatic corrective actions.
10. **Data Logging and Analysis**: Incorporate logging capabilities to record temperature and fan speed data over time. This data can be analyzed to optimize the control algorithm further or identify trends in temperature changes.

By integrating these enhancements, you can create a more sophisticated and efficient automatic fan speed controller that improves comfort, energy efficiency, and user experience.

**XIII. CONCLUSION**

In conclusion, the automatic fan speed controller based on temperature offers significant advantages in terms of comfort, energy efficiency, and convenience. By automatically adjusting fan speeds according to real-time temperature fluctuations, it optimizes cooling efficiency while minimizing energy consumption. The controller's ability to maintain a desired temperature range enhances user comfort, ensuring a pleasant environment without the need for manual adjustments.

Future enhancements, such as advanced temperature sensing, PID control algorithms, and integration with smart home systems, promise to further refine and optimize the functionality of these controllers. These improvements will not only increase precision and responsiveness but also enable seamless integration into modern smart home environments, offering users greater flexibility and control over their cooling systems.

Overall, the automatic fan speed controller represents a valuable solution for enhancing environmental control in residential, commercial, and industrial settings, contributing to both comfort and energy savings in a sustainable manner

In summary, the automatic fan speed controller based on temperature is a sophisticated solution that effectively enhances both comfort and energy efficiency in various settings. By continuously monitoring ambient temperatures and adjusting fan speeds accordingly, the controller optimizes cooling performance without constant manual oversight.

This technology not only ensures a consistent and comfortable environment but also contributes to significant energy savings by operating fans only when needed. The ability to integrate advanced sensors and control algorithms further refines its precision and responsiveness, adapting dynamically to changing conditions.

Looking ahead, the ongoing evolution of automatic fan speed controllers holds promise for even greater efficiency and functionality. Future advancements may include enhanced smart home integration, finer control through PID algorithms, and improved diagnostics for proactive maintenance.

In conclusion, the automatic fan speed controller based on temperature represents a cornerstone of modern HVAC systems, offering a sustainable solution that enhances user experience while promoting energy conservation. Its role in optimizing indoor climate control underscores its value in both residential and commercial applications, where comfort and efficiency are paramount.

**XIV. REFERENCES**

* Embedded Systems Design with the Atmel AVR Microcontroller
* PID Controllers: Theory, Design, and Tuning
* IEEE Xplore
* Arduino Programming in 24 Hours, Sams Teach Yourself