International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 08, ISSN No. 2455-2143, Pages 89-93
Published Online December 2023 in IJEAST (http://www.ijeast.com)

ENERGY OPTIMIZATION TECHNIQUE BY AUTOMATIC FAN SPEED CONTROLLER BASED ON SURROUNDING TEMPERATURE BY USING ARDUINO

Swatanshu Shekhar Singh, Vaibhav Saraswat
Department of Electrical Engineering
Malaviya National Institute of Technology
(MNIT)Jaipur, Rajasthan, India

Abstract:- **Automatic Fan speed Controller system is completely based on controlling the fan speed automatically based on Surrounding temperature. As in Software simulation we have used the LM 35 Temp. Sensor to vary the surrounding temperature. On the other side, at the Output, we have used the DC Fan as to observe its speed control based on input Temp i.e. temp of LM35 sensor. Simply if the Temp falls below a preset value then the speed of the fan decrease as required. And if the Temp rises above a preset value then the speed of the fan increases as per the Preset Value. The sensed value of the temperature is displayed on the 16x2 LCD, followed by display of the Fan Speed.**

Keywords: **Automatic, Controlling, Preset, Senses**

I. INTRODUCTION

It's worth noting that implementing temperature-based fan speed control requires appropriate temperature sensors and a control mechanism, such as a microcontroller, to monitor the temperature and adjust the fan speed accordingly. But here we have accomplished the same mechanism without using any costly device or any sensor. We have only used a LM35 Temp sensor at the input side only just to vary the surrounding temperature. The sensed value of the temperature is displayed on the 16x2 LCD, followed by display of the Fan Speed. The system is developed to monitor the room temperature and transmit the data to an Arduino board. The Arduino board compares the current temperature with the desired temperature using its programmed instructions. The results of this operation are displayed on an LCD screen for easy reference. Additionally, the Arduino board generates pulses that are sent to the driver circuit to control the fan and achieve the desired output.

II. METHODOLOGY

This proposed Automatic Fan speed controlling system derived from a set of different components selected in the field of electrical and electronics engineering. This system will detect the Input temperature. And if the Input Temp exceeds the desire value the speed of the Fan increases and if the Input temp falls below a desired value then the speed of the Fan will also decrease gradually. All the parameters including Fan speed and Input Temperature will be displayed on the LCD Screen

The main component of the proposed system consists of a Arduino UNO, LCD for displaying information, Buzzer, Resistor, Potentiometer, Capacitor, Diode, MOSFET and an LM35 Temperature Sensor.

2.1 Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It receives the real-time analog data from the sensors through its Analog to Digital Converter (ADC). The digital equivalent of the analog data is then processed by the microcontroller based on control algorithms programmed and stored in its flash memory. The control algorithm is written such that the real temperature, voltage and current values can be obtained from the sensors output signals. Instead of requiring a physical press of the reset button, the Arduino Uno board incorporates a software-based reset mechanism. The board is designed in such a way that software running on a connected computer can initiate the reset process. This is achieved by connecting one of the hardware flow control lines (DTR) of the ATmega8U2/16U2 to the reset line of the ATmega328 through a 100 nF capacitor. When the DTR line is activated (brought low), the reset line briefly drops, triggering a reset of the microcontroller chip.

This setup has additional consequences. When the Uno is connected to a computer running Mac OS X or Linux, it undergoes a reset each time a software connection is established via USB. During this brief period (approximately half a second), the Uno's boot loader is active. While the boot loader is programmed to ignore malformed data (anything other than a valid code upload), it does intercept the initial few bytes of data sent to the board immediately after the connection is opened.

Connections:-

- **RESET Pin to Power Supply**
- **Pin A0 to Capacitor followed by Ground**
- **Pin 13 to Buzzer**
- **Pin 111 to MOSFET Circuit**

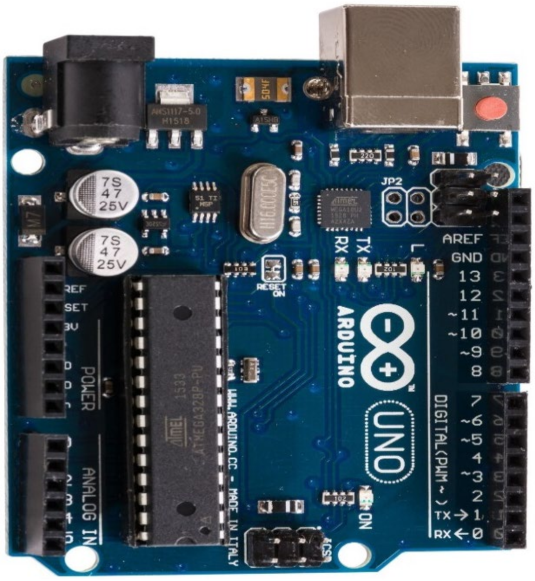


Figure 1: Arduino UNO

2.2 Temperature sensing Unit:- LM35 Temp Sensor

For temperature sensing, we have chosen the LM35 temperature sensor. The LM35 is known for its high precision and accuracy. It provides an output that is directly proportional to the Celsius temperature being measured. The LM35 has a wide temperature range, capable of operating from -55°C to 150°C. It has a linear scale factor of +10mV/°C, ensuring reliable and consistent temperature readings. The LM35 temperature sensor is calibrated directly in degree Celsius (centigrade), ensuring accurate temperature measurements. It has a linear scale factor of +10.0 mV per degree Celsius, allowing for precise temperature sensing. The sensor provides a high level of accuracy, with a tolerance of 0.5 degree Celsius at +25 degrees Celsius. It is designed to operate within a wide temperature range, from -55 to +150 degrees Celsius. The LM35 exhibits low self-heating, with a minimal increase of only 0.08 degree Celsius in still air. It maintains excellent linearity, with nonlinearity typically limited to +/- 1/4 degree Celsius. Additionally, it has a low impedance output of 0.1 ohms, ensuring compatibility with a 1mA load.

Connections:

- **Pin 1 to Power Supply**
- **Pin 2 to A1 of the Arduino**
- **Pin 3 to Ground**

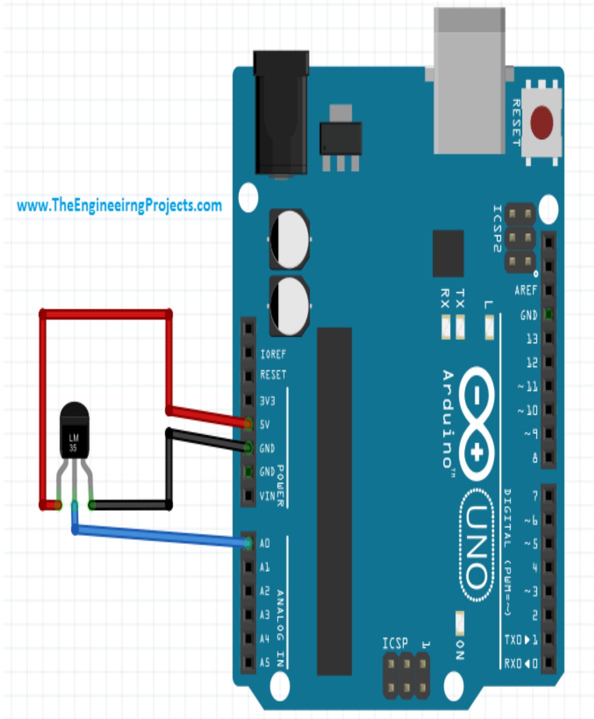


Figure 2: Lm35 Temp Sensor with Arduino UNO

2.3 16*2 LCD Display

A "16 by 2 LCD display" refers to a commonly used alphanumeric liquid crystal display (LCD) module. The "16 by 2" specification indicates that the display has 16 columns and 2 rows, allowing it to display 16 characters per line and 2 lines of text.The control pins are (RS, RW, E) and data pins are (D4-D7). Here we will be using 16 by 2 LCD Display to display the Input Temperature or the Surrounding temperature followed by the Fan Speed. The LCD display consists of two registers: the Command register and the Data register. The Command register is responsible for storing the instructions given to the LCD, such as initializing it, clearing the screen, setting the cursor position, and controlling the display. On the other hand, the Data register stores the actual data that needs to be displayed on the LCD. This data corresponds to the ASCII value of the character that is to be shown on the screen.

Connections:-

- **Connect the VDD Pin 2 of LCD Display to the Power Supply**
- **Connect the VEE Pin 3 of LCD Display to the Potentiometer**
- **Connect the PIN 4 (RS) of LCD Display to PIN 2 of Arduino UNO**

- **Ground the PIN 5(RW) of LCD Display to Ground**
- **Connect the PIN 6 (E) of LCD Display to PIN 3 of Arduino UNO**
- **Connect the PIN 11 (D4) of LCD Display to PIN 4 of Arduino UNO**
- **Connect the PIN 12 (D5) of LCD Display to PIN 5 of Arduino UNO**
- **Connect the PIN 13 (D6) of LCD Display to PIN 6 of Arduino UNO**
- **Connect the PIN 14 (D7) of LCD Display to PIN 7 of Arduino UNO**
- **Short Circuit the VSS Pin 1 of LCD Display to Pin RW 5 of the LCD Display**

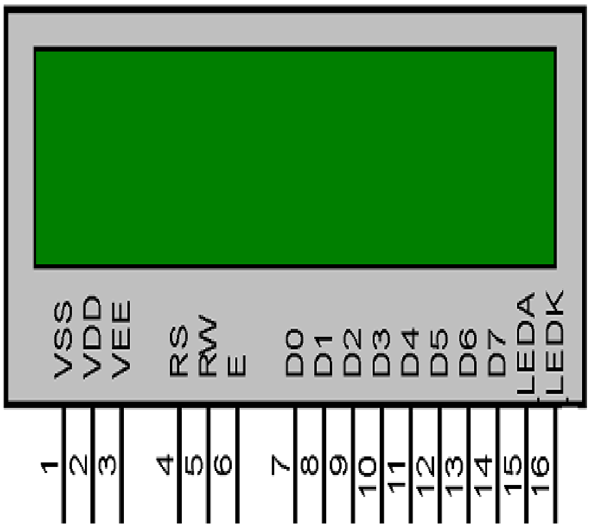


Figure 3: 16*2 LCD Display

2.4 Assembly of the Automatic Fan speed Controller

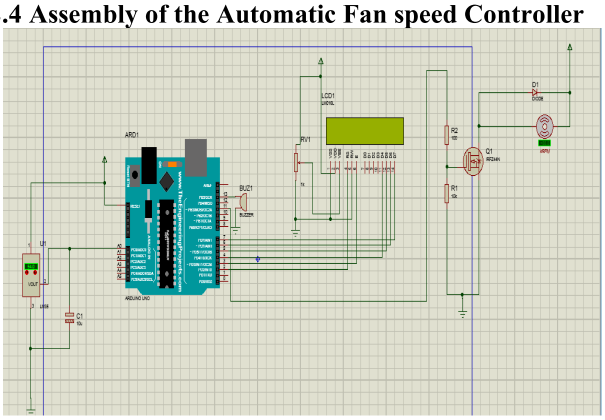
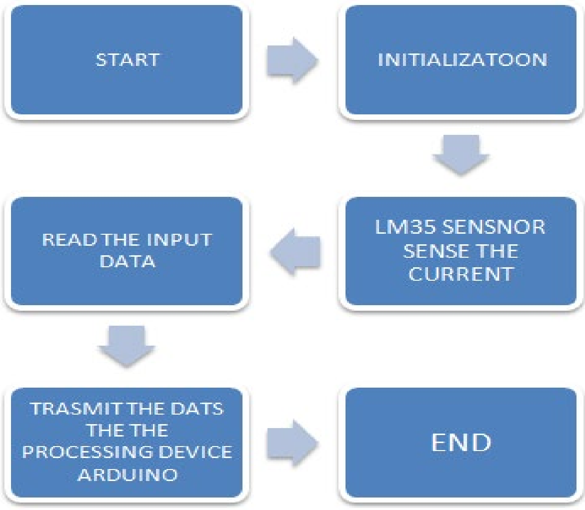


Figure 4: Assembly of Components in Software Simulation

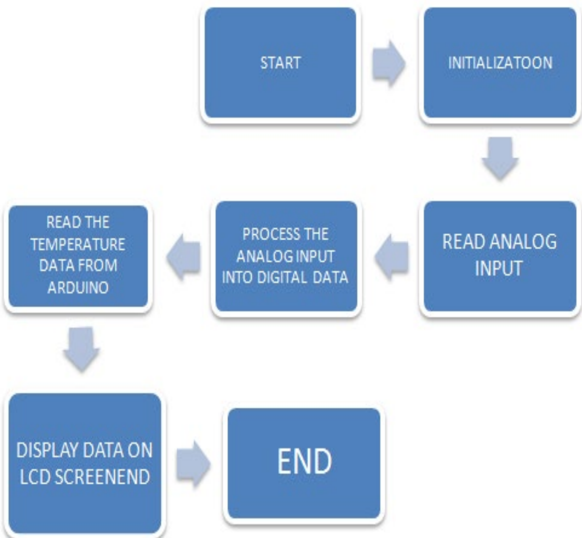
2.5 Programming and Illustration

This section of the paper depicts the development of the Software part. The continuous operation of the Automatic Fan speed Controller has been shown by the Flow chart.

The given flow chart shows the sequential operation of the Software simulation. When the system is turned ON, the initiation takes place, where initially our Input Device, LM35 Sensor draws some Temperature. This current is sensed by theLM35 Temp. Sensor. Further this Temperature is converted from Analog to digital Signal via Arduino



The next Flowchart represents the sequential flow to run the program to perform the simulation. When the processing Unit Arduino process the data from the Input side, the same information is converted into Digital Signal. The processing device segregate the data appropriately which is transmitted to the receiving unit and is displayed on the LCD Screen.



III. RESULT AND DISCUSSION

The result and discussion from those results are interpreted in this section. During the initial conditions the Value of Temperature is 30 Degree Celsius, at the same instant the Speed of Fan is 0%. As we gradually increase the value of the temperature then slightly the speed of the fan will also increase. At some point, like at which the value of variable temperature is 50 Degree Celsius, the speed of the fan will be at its maximum that is 100 %. At this point of time the fan is rotating at its full speed", and as soon as it starts operating at full speed, buzzerwill start Beeping.

The most important here is to set the variables temp Min and temp Max with our desired values. Tmin is the temperature at which the fan starts to spin and Tmax is the temperature when the buzzer will beep and give warning sign that the maximum temp was reached. For example if we set Tmin at 15 and Tmax at 25 then the fan will start spinning at 15°C and reach its maximum speed at 25°C.

The graph below also depicts the same thing. We have plotted the x-y graph of Value of Temperature and of Value of speed of Fan respectively. From Initial conditions the value of Temperature the graph will follow an upward trend, followed by increase in Speed of Fan. But after the threshold point the graph will become constant.

Table 1: Relationship between the Value of Temperature and % Speed of Fan

Value Of Temp. (Degree Celsius)	% Speed of Fan
30	0%
32	10%
34	20%
36	30%
38	40%
40	50%
42	60%
44	70%
46	80%
48	90%
50	100%
52	100%

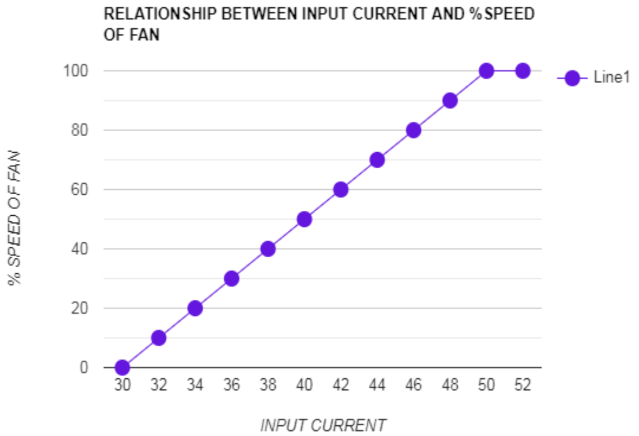


Figure 5: Graphical Representation of Value of Input Temp and % Speed of Fan

We can clearly interpret that the LCD display on the fan module adjusts the temperature readings to represent the fan speed on a scale of 0 to 100%. This means that the fan's RPM speed is directly proportional to the temperature measured by the LM35 sensor. Once the temperature reaches the specified value of Tmax(50 Degree Celsius), the fan will operate at its maximum speed, and the LCD will show "FAN SPEED 100%". It's important to note that even if the temperature continues to rise beyond Tmax, the display will still indicate a fan speed of 100%.

IV. CONCLUSION AND FUTURE SCOPE

A temperature-based fan speed controller is an effective solution for efficiently cooling processors in laptops and personal computers. Typically, laptop fans are limited to two or three fixed speeds, which can lead to higher power consumption. However, in this project, the fan is designed to adjust its speed based on temperature changes. This feature makes it suitable for cooling electrical and mechanical equipment in small-scale industries as well. The entire circuit, excluding the motor and fan, can be integrated onto a single printed circuit board (PCB) for easy implementation of temperature-based control operations.

Conflict of Interest

Both the Authors hereby declared that we do not have any conflict of interest.

Funding Source

None

Authors' Contributions

Author-1 researched literature and conceived the study. He performed simulation and wrote the first draft of the manuscript. Author 2 was involved in data analysis and reviewed while both the authors edited the manuscript and approved the final version of the manuscript.

V. REFERENCES :-

[1] Ghana Shyam Soren, Ram Ashish Gupta, " Temperature controlled DC fan using Microcontroller", National institute of technology-Rourkela.

[2] Kunal Singh, Moloy Dhar, Pritam Roy,"Automatic fan speed control system using Arduino", International Journal of Novel Research and Development.

[3] Mustafa Saad, Hossam Abdoalgader, and Muammer Mohamed, "Automatic Fan Speed Control System Using Microcontroller"

[4] Z. I. Rizman, K. Ho Yeap, N. Ismail, Design an Automatic Temperature Control System for Smart Electric Fan Using PIC, International Journal of

Science and Research (IJSR), India Online ISSN: 2319-7064 Volume 2 ,Issue 9,September 2013.

[5] Sixth Int'l Conference on Electrical, Electronics and Civil Building (ICEECE'2014) Nov. 27-28, 2014 Cape Town (South Africa) Automatic Fan Speed Control System Using Microcontroller Mustafa Saad, Hossam Abdoalgader, and Muammer Mohamed.

[6] K. Singh, M. Dhar, P. Roy, Automatic fan speed control system using Arduino, ISSN: 2456-4184 International Journal of Novel Research and Development (IJNRD)4 April 2017.

[7] Chiueh, H., Choma, J. and Draper, J. (2000). Implementation of a temperature Monitoring Interface Circuit for Power PC Systems (pp.98-101), Proceedings of the 43rd Midwest Symposium on Circuits and Systems, Lansing MI, USA.

[8] Mashud M. A. A, Barman S. C, Bhuiyan M. R. A, Islam Md.Serajul. (2013). Design and Implementation of Three Phase Unbalanced Voltage Motor Controller (pp.35-42), I.J. Intelligent Systems and Applications.

[9] Surabhi, Upendra Prasad, Vivek Kumar Jain “Design and Fabrication of Temperature based DC Fan Speed Control System using Microcontroller and Pulse Width Modulation Technique” International Journal of Innovative Research in Science, Engineering and Technology

[10] Liu, Y., Zeng, J. and Wang, C. (2009). Temperature Monitoring in Laser Assisted Polymer Bonding for MEMS Packaging Using a thin Film Sensor Array , IEEE Sensors Applications Symposium, New Orleans, LA , USA.

[11] K.A Akpado1, C.O Ezeagwu2, A. Ejiofor3, A.O N wokeke.Vol. 2, Issue 7, July 2013, Modeling and Simulation of a Microcontroller Based Temperature Control in a Ventilation System