Temperature Based Fan Speed Controller Using Arduino


Corresponding Author: Dr. Shailaja M. E&C Tontadarya college of Engineering Gadag.

ABSTRACT: We all know that we are moving towards the automation. So basically in this project we will be controlling the speed of the fan according to the temperature of the surrounding. The system will get the temperature from the temperature sensor and it will control the speed according to the temperature set by the user. It also uses the LCD display to display temperature from – 55 to 125°C and speed of the fan in percentage. The Arduino device plays a major role as the processing part. If the surrounding temperature is greater than set temperature fan turns on vice-versa.

By this project it is helpful to save the energy where saving of energy is one of the most important thing we are facing load shedding.

KEYWORDS: Arduino UNO, LM35, LCD display.

I. INTRODUCTION:

In this project we will be using the concept of firing angle to control fan speed. Basically the project is based on controlling fan speed with respect to the temperature.

The system will get the temperature from the temperature sensor via the ADC and it will control the speed according the set by the user. The temperature is set in the source code are used to increment and decrement the temperature value according to the user requirement.

In this project, an arduino Uno forms the processing part, which firstly senses the temperature via the ADC the controller then compares the data with the set temperature which the user can set via the keypad. If the current temperature is greater than the set temperature the controller turns ON the fan and the set speed will be proportional to the difference between the set temperature and the current temperature. If the current temperature is less than the set temperature the fan will be turned OFF. We will be using a transistor to provide the require firing angle to the fan and the fan speed will change accordingly.

We will be interfacing LCD display which will be used to display the set temperature as well as the fan speed in percentage.

II. BLOCK DIAGRAM: The below fig.1 shows the block diagram of the temperature based fan speed controller,

As shown in above block diagram arduino uno acts as the processing devise or controlling device, temperature sensor and power supply are the input devices to the arduino, fan and LCD display are the output units.

III. CIRCUIT DIAGRAM:

The below fig.2 shows the circuit diagram of the temperature based fan speed controller,
**Working:** The LM35 temperature sensor senses the surrounding room temperature and analog signal is given to the Arduino. Arduino converts the analog signal into a digital signal. It calculates the speed of the fan rotation in percentage and displays the fan speed and temperature readings in an LCD display.

The calculated fan speed is sent to the fan through a low frequency pulse width modulation signal. This adjusts the speed of the fan by varying the duty cycle. A 2n2222 transistor is used because it is good efficient and can act as a switch and amplify the signal.

**IV. MAJOR COMPONENTS USED:**
Specification of the major components used in the building of the circuit diagram is as shown in fig.3, fig.4, fig.5.

- **Arduino UNO:**

  ![Arduino UNO](image)

  **Specifications:**
  1. Microcontroller - ATmega328
  2. Operating Voltage - 5V Input Voltage
  3. Input Voltage - 6 to 20V
  5. DC Current for the 3.3V Pin - 50 mA
  6. Flash Memory - 32KB SRAM : 2KB
  7. EEPROM - 1KB
  8. Clock Speed - 16MHz

- **LM35**

  ![LM35](image)

  **Specifications:**
  1. Measures directly in degree Celsius
  2. Linear + 10.0mV/degree Celsius
  3. 0.5 degree Celsius accuracy (at +25 degree Celsius).
  4. Rated between -55 to +150 degree Celsius range.
  5. Low cost due to wafer-level trimming
  6. Operates from 4 to 30 volts
  7. Less than 60 Microampere current drains.
  8. Low self-heating, 0.08 degree Celsius and Nonlinearity only +/- 1/4 degree Celsius.

- **LCD Display**

  ![LCD Display](image)

  **Specifications:**
  1. Ground(0V)
  2. Supply voltage; 5V(4.7V – 5.3V)
  3. Contrast adjustment;
  4. When low command register is selected & when high data register is selected.
  5. Low to write to the register; High to read from the
6. Sends data to data pins when a high to low pulse is given.
7. 8-bit data pins
8. Back light Vcc(5V)
9. Back light Ground(0V)

V. SOURCE CODE:

```c
#include <LiquidCrystal.h>

//LiquidCrystallcd(12, 11, 5, 4, 3, 2);
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x3F, 16,2);
int tempPin = A0
int fan = 11;
int led = 8;
int temp;
int tempMin = 30; int tempMax = 60; int fanSpeed;
int fanLCD;

void setup() {
  pinMode(fan, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(tempPin, INPUT);
  lcd.begin(16,2);
  Serial.begin(9600);
}

void loop()
{
  temp = readTemp();
  Serial.print( temp );
  if(temp < tempMin)
    {
      fanSpeed = 0;
      analogWrite(fan, fanSpeed);
      fanLCD=0;
      digitalWrite(fan, LOW);
    }
  if((temp >= tempMin) && (temp <= tempMax))
    {
      fanSpeed = temp;//map(temp, tempMin, tempMax, 0, 100);
      fanSpeed=1.5*fanSpeed;
      fanLCD = map(temp, tempMin, tempMax, 0, 100); // speed of fan to display on LCD100
      analogWrite(fan, fanSpeed);
    }
  if(temp > tempMax)
    {
      digitalWrite(led, HIGH);
    }
  else
    {
      digitalWrite(led, LOW);
    }
  lcd.print("TEMP: ");
  lcd.print(temp);
  lcd.print("C ");
  lcd.setCursor(0,1);
  lcd.print("FANSPEED: ");
  lcd.print(fanLCD);
  lcd.print("");
  delay(200);
```

lcd.clear();
}

intreadTemp()
{
    temp = analogRead(tempPin);
    return temp * 0.48828125;
}

VI. RESULT:
The below fig.6 shows the working module of the temperature based fan speed controller.

The following table.1 depicts the results obtained by operating the prototype model at various different temperatures. It defines the behavior of the embedded system about how it reacts to variation of temperature at real time.

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Duty Cycle</th>
<th>PWM Value</th>
<th>Fan Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 26</td>
<td>0%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>20%</td>
<td>51</td>
<td>227</td>
</tr>
<tr>
<td>27</td>
<td>40%</td>
<td>102</td>
<td>428</td>
</tr>
<tr>
<td>28</td>
<td>60%</td>
<td>153</td>
<td>654</td>
</tr>
<tr>
<td>29</td>
<td>80%</td>
<td>204</td>
<td>826</td>
</tr>
<tr>
<td>Greater 29</td>
<td>100%</td>
<td>255</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table.1

VII. CONCLUSION & FUTURE SCOPE:
Conclusion: This project elaborates the design construction of fan speed control system to control the room temperature. Efficient temperature sensor was chosen to take the room temperature. Hence the fan speed will increase automatically if the room temperature is increased. As conclusion, the system perform very well for any temperature changes in the sourandng and can be classified as automatic control device.

Future scope:
- The proposed model can be improvised further to monitor humidity, light & at a same time to control them.
- The project will concentrate on electric fan rather than other type of fan such as ceiling fan.
- We can also draw graphs of variation in the parameters using computer.
- It is used to send the data to the remote location using internet.
REFERENCES:

