Controlling Fan Speed Based on Temperature Using Arduino

Research Project Submitted to the Department of Electrical Engineering in Partial Fulfillment of the Requirements for the degree of B.Sc. in Electrical Engineering

Prepared By:
Ravand jihanger

Supervised by:
Azad N. Abdulla

Dedication

This study is dedicated to our beloved parents, who have been our source of inspiration and gave us strength when we thought of giving up, who continually provide their moral, spiritual, emotional, and financial support.

Acknowledgment

Foremost, we acknowledge the Almighty God for giving us physical and spiritual strength during our studies. Also, we would like to take this opportunity to express our gratitude and sincere thanks to our supervisor Azad N. Abdulla. Appreciate the help. Finally, we extend our appreciation to our family for supporting us during our studies.

Abstract

In accordance with our needs, this project's independent automatic fan speed controller regulates the speed of an electric fan. It has a small number of components and can be used for a variety of purposes.The level of fan speed and the detected temperature.The microcontroller receives an electrical (analog) signal from the temperature sensor LM35 after it has sensed the temperature. Values are simultaneously shown on the fan speed controller LCD panel, which regulates an electric fan's speed in accordance with our needs.The project helps maintain and manage temperature in the process industries.
By connecting an LM35 temperature sensor to an Arduino, we will demonstrate in this brief tutorial how to create an automatic temperature-based fan speed controller that can track both the fan speed and temperature in real time.

Table of Contents

List of Figures

Chapter One
Introduction

First of all We have to know that this system is helping a huge progress in our world to keep our temperature devices stable with upgrading devices.

Microcontrollers are utilized nowadays to perform automated operations more accurately in a wide range of life professions. Almost all modern devices, such as cooling systems, mechanical, items to play with, and computers, rely on microcontrollers to function. The Central Processing Unit (CPU) is the main component of a microcontroller. a key component in the creation of smart systems is the microcontroller. The primary use of a microcontroller is as a single-chip microprocessor for robotic and control in industry.

So a device that senses the ambient temperature and regulates the fan speed to keep the temperature stable is called a temperature-based fan speed controller. An Arduino board can be used to create this system through the use of an electrical circuit.

1.1 Problem Statement:

The issue comes in hot and cold weather when the air conditioner fan is still running but not under our control or the fan not working as we want in the specific temperature. if weather start increasing heat and the fan not able to control it because of the speed limit there for we have to change the fan to bigger one, or add another fan to control the temperature.

1.2 Research Goals/Objectives:

The function is uncontrollable and needs to be manually activated and deactivated, or the fan's speed needs to be lowered. so we work on it in every level of temperature the fan must be not at the same speed or must be stop when the weather is cold. Also we have to make the fans speed faster when temperature increasing.

Chapter Two
Theoretical Background

In this project, many electrical components have been used for sensing, switching, controlling…etc. In this chapter, each component will be defined. Components that have been used in this project are as follows:

2.1 Arduino Nano:

Arduino Nano is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, relays, LEDs, servos, and motors as an output. so a portable, complete, and breadboard-friendly board called the Arduino Nano is built around the ATmega328P or ATmega628 microcontroller123. It is smaller in size and comes with a Mini-B USB cable124, but it has the same connectivity and functionality as the Arduino Uno board. The Arduino Software can be used to program it.

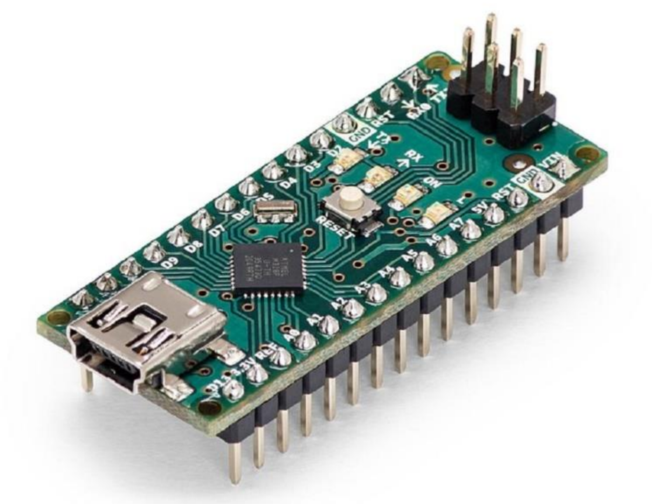

Figure 1: Arduino Nano

2.2 Temperature Sensor:

We use LM 35 as temperature sensor. LM 35 is precision The temperature sensor has an output linearly proportional to Temperature Celsius. The LM35 is designed to operate from -55°Degrees C to 150° C with linear scaling factor of +10mv/° C.

Figure 2: Temperature Sensor

2.3 DC Motor:

The most typical kind of motor is a DC motor, or direct current motor. DC motors only have two leads: a positive lead and a negative lead. The motor will turn if you connect these two leads directly to a battery. The motor will turn in the opposite direction if the leads are switched. For DC motors, the three most important quality factors are speed, torque, and operating voltage.

Figure 3: DC Motor

2.4 Transistor:

A transistor is a semiconductor device used as amplifier and switch electronic signals and electrical power, transistor is a basic electrical component that alters the flow of electrical current.

Figure 4: Transistor

2.5 LCD 16x2:

An LCD is a unique type of display that can only output individual ASCII(American Standard Code for Information Interchange) characters with a fixed size. Using these individual characters then we can form a text.

The number of rectangular areas defines the size of the LCD. The most popular LCD is the 16×2 LCD(the same that was used in this project), which has two rows with 16 rectangular areas or characters. Of course, there are other sizes like 16×1, 16×4, 20×4, and so on, but they all work on the same principle. Also, these types of LCDs can have different backgrounds and text colors.

Figure 5: LCD

Chapter Three
Methodology

It is possible to set up this system with an Arduino board and an electronic circuit. Currently, among all electronic boards, Arduino is quite progressive. we used an Arduino board to control the speed of the fan authority.It is intended for the suggested system to recognize the temperature and forward that data to the Arduino board. Next, the contrast is carried out by the Arduino board. set temperature based on the current temperature and the embedded Arduino application.

The result of the procedure is provided by the connecting Arduino board's o/p port to an associated LCD display statistics. The board's generated pulses, which are additionally supplied to the driver circuit to obtain the desired output for the fan.

Chapter Four
Proposed Circuit Design

Home and industrial applications are possible with this project.This will assist in protecting electricity and energy. as well To see areas that are uncomfortable or difficult for people to detect, especially for a long period of time. This keeps energy from being wasted when the temperature isn't high enough to require a fan. To help people who are unable to control the fan speed automatically.

But on the other hand Just a technical person can maintain it. As a result, maintaining it becomes harder. And Sometimes, its efficiency may drop as a result of temperature changes.

More parameters, such as light and temperature, are under our control at the same time. We can use a mobile device or the internet to deliver this info to a distance location. An automatic dialer system will phone the specified number when the temperature rises above the limit.

- **Proposed Circuit**

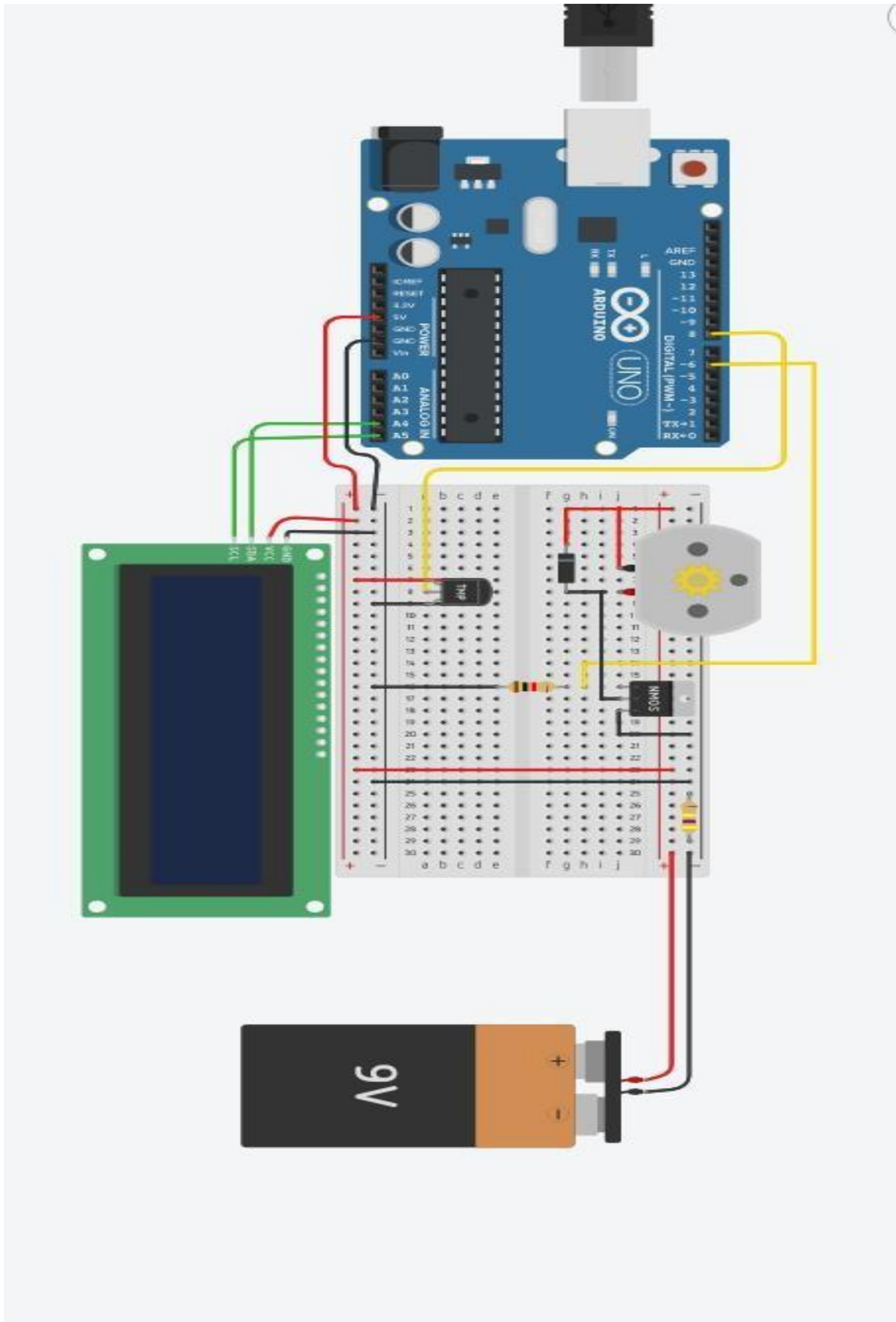

Figure 6: Proposed Circuit

Chapter Five
Conclusion

Experimental research has provided solutions to the cooling issues, making my project suitable for the task. In cooling devices, this system is very helpful. This system is entirely automatic.

The fan speed in this example project will be adjusted based on the real-time temperature measurements obtained from the LM35 temperature sensor.

the fan will spin faster at high temperatures and turn off when the temperature drops. When the temperature on the LM35 temperature sensor is detected, the fan speed will be immediately proportionate.

The temperature is sensed by the temperature sensor LM35, which then transforms it into an electrical (analog) signal that is applied to the Arduino UNO Board's microcontroller. A digital value is created out of the analog value. As a result, the fan's speed and temperature readings are displayed on the LCD.

References

1. Mark Geddes. *Arduino Project Handbook.*
2. Micheal McRoberts. *Beginning Arduino.*
3. Scott Fitzgerald & Michael Shiloh. *Arduino Projects Book*
4. Simon Monk. *30 Arduino Projects: For The Evil Genius*
5. Micheal Margolis. *Arduino Cookbook*

Appendix – Arduino Code

```
#include <LiquidCrystal_I2C.h>


#include <dht.h>
#define dataPin 8 // Defines pin number to which the sensor is connected
dht DHT; // Creats a DHT object
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 column and 2 rows
 int outPin = 6;


void setup()
{
Serial.begin(9600);
lcd.init(); // initialize the lcd
lcd.backlight();

pinMode(outPin, OUTPUT);

}

void loop()
{

//Uncomment whatever type you're using!
int readData = DHT.read22(dataPin); // DHT22/AM2302
//int readData = DHT.read11(dataPin); // DHT11

float t = DHT.temperature; // Gets the values of the temperature
 float h = DHT.humidity; // Gets the values of the humidity

// Printing the results on the serial monitor
```

```
delay(100); // Delays 2 secods
lcd.clear();   // clear display
lcd.setCursor(0, 0);         // move cursor to   (0, 0)
lcd.print("Temp = ");
lcd.print(t);
 lcd.print(" ");
lcd.print((char)176);
 lcd.print(" c");
lcd.setCursor(0, 1);         // move cursor to   (2, 1)
lcd.print("Humidity = ");
lcd.print(h);
lcd.print(" % ");



if (t >=25 ){
lcd.clear();   // clear display
lcd.setCursor(0, 0);                   // move cursor to   (0, 0)
lcd.print("Temp = ");
lcd.print(t); lcd.print(" ");
lcd.print((char)176);
 lcd.print(" c");
lcd.setCursor(0, 1);         // move cursor to   (2, 1)
 lcd.print("Humidity = ");
lcd.print(h); lcd.print(" % ");

analogWrite(outPin, 100);
 delay(5000);


}

if (t >=35 ){
lcd.clear();           // clear display
lcd.setCursor(0, 0);           // move cursor to  (0, 0)
lcd.print("Temp = ");
lcd.print(t);
```

```
lcd.print(" ");
 lcd.print((char)176);
 lcd.print(" c");
lcd.setCursor(0, 1);          // move cursor to    (2, 1)
lcd.print("Humidity = ");
lcd.print(h);
lcd.print(" % ");

analogWrite(outPin, 150);
delay(5000);

}

if (t >=45 ){
lcd.clear();            // clear display
 lcd.setCursor(0, 0);                   // move cursor to    (0, 0)
 lcd.print("Temp = ");
lcd.print(t);
 lcd.print(" ");
lcd.print((char)176);
 lcd.print(" c");
lcd.setCursor(0, 1);          // move cursor to    (2, 1)
lcd.print("Humidity = ");
lcd.print(h);
lcd.print(" % ");

analogWrite(outPin, 200);
 delay(5000);

}

else{
 analogWrite(outPin, 0);

}
```