Project submitted in partial fulfillment for the Degree of B. Tech in Applied Electronics & Instrumentation Engineering under West Bengal University of Technology

TEMPERATURE BASED FAN SPEED CONTROLLER

By

Biman Kr Pal (Roll No.11705515054)
Sagar Ghosh (Roll No. 11705515058)
Subhankar Paul (Roll No.11705515059)
Avijit Dhibar (Roll No.11705514012)

Guide Name

Mr. Kalyan Biswas

श्रमम् बिना न किमपि साध्यम्

DEPARTMENT OF APPLIED ELECTRONICS & INSTRUMENTATION ENGINEERING, RCC INSTITUTE OF INFORMATION TECHNOLOGY, CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015,
November 2017

ACKNOWLEDGEMENT

It is a great privilege for us to express our profound gratitude to our respected teacher Mr.Kalyan Biswas, Applied Electronics &Instrumentation Engineering, RCC Institute of Information Technology, for his constant guidance, valuable suggestions, supervision and inspiration throughout the course work without which it would have been difficult to complete the work within scheduled time.

We would like to express our gratitude towards Mr.Arijit Ghosh,Ms.Naiwrita Dey,Mr.Srijan Bhattacharya,Mr.Avishek Paul for his/her kind co-operation and encouragement which helped me in completion of this project.

We are also indebted to the Head of the Department, Applied Electronics & Instrumentation Engineering, RCC Institute of Information Technology for permitting us to pursue the project.

We would like to take this opportunity to thank all the respected teachers of this department for being a perennial source of inspiration and showing the right path at the time of necessity.

Biman Kr Pal

Sagar Ghosh

Subhankar Paul

Avijit Dhibar

CERTIFICATE OF APPROVAL

The project report titled "**TEMPERATURE BASED FAN SPEED CONTROLLER** " prepared by **Biman Kr Pal** , Roll No: 11705515054, **Sagar Ghosh** , Roll No: 11705515058, **Subhankar Paul** , Roll No: 11705515059, **Avijit Dhibar** , Roll No: 11705514012; is hereby approved and certified as a creditable study in technological subjects performed in a way sufficient for its acceptance for partial fulfilment of the degree for which it is submitted.

It is to be understood that by this approval, the undersigned do not, necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

Mr. Kalyan Biswas

[Supervisor]

Mr. Kalyan Biswas

[Head of the Department]

Applied Electronics & Instrumentation Engineering

RECOMMENDATION

I hereby recommend that the project report titled "**TEMPERATURE BASED FAN SPEED CONTROLLER** " prepared by **Biman Kr Pal**, Roll No: 11705515054, **Sagar Ghosh**, Roll No: 11705515058, **Subhankar Paul** , Roll No: 11705515059, **Avijit Dhibar** , Roll No: 11705514012 be accepted in partial fulfillment of the requirement for the Degree of Bachelor of Technology in Applied Electronics & Instrumentation Engineering, RCC Institute of Information Technology.

Mr. Kalyan Biswas

......…………………………….

(Supervisor)

Table of content:

List of Figures

ABSTRACT

This project is a standalone automatic fan speed controller that controls the speed of an electric fan according to our requirement. Use of embedded technology makes this closed loop feedback control system efficient and reliable. Microcontroller (ATMega8 / 168 / 328) allows dynamic and faster control. Liquid crystal display (LCD) makes the system user-friendly. The sensed temperature and fan speed level values are simultaneously displayed on the LCD panel. It is very compact using few components and can be implemented for several applications including air-conditioners, water-heaters, snow-melters, ovens, heat-exchangers, mixers, furnaces, incubators, thermal baths and veterinary operating tables. ARDUINO micro controller is the heart of the circuit as it controls all the functions. The temperature sensor LM35 senses the temperature and converts it into an electrical (analog) signal, which is applied to the microcontroller. The sensed and set values of the temperature are displayed on the 16x2-line LCD. The micro controller drives Transistor to control the fan speed. This project uses regulated 12V, 2A power supply. This project is useful in process industries for maintenance and controlling of Boilers temperature.

INTRODUCTION

With the advancement in technology, intelligent systems are introduced every day. Everything is getting more sophisticated and intelligible.There is an increase in the demand of cutting edge technology and smart electronic systems. Microcontrollers play a very important role in the development of the smart systems as brain is given to the system. Microcontrollers have become the heart of the new technologies that are being introduced daily. A microcontroller is mainly a single chip microprocessor suited for control and automation of machines and processes. Today, microcontrollers are used in many disciplines of life for carrying out automated tasks in a more accurate manner. Almost every modern day device including air conditioners, power tools, toys, office machines employ microcontrollers for their operation. Microcontroller essentially consists of Central Processing Unit (CPU), timers and counters, interrupts, memory, input/output ports, analog to digital converters (ADC) on a single chip. With this single chip integrated circuit design of the microcontroller the size of control board is reduced and power consumption is low. This project presents the design and simulation of the fan speed control system using PWM technique based on the room temperature. 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 using PWM technique. The duty cycle is varied from 0 to 100 to control the fan speed depending upon the room temperature, which is displayed on Liquid Crystal Display.

PROPOSED SYSTEM

In the proposed systems, microcontroller plays a vital role in the smart systems development.Microcontrollers have become an essential part in the present technologies that are being presented daily. This article discusses temperature based fan speed control and monitoring system using an Arduino system.This system is used to control the cooling system automatically based on the room temperature.The system uses an Arduino board to implement a control system. Since this system is proposed to control the cooling system and it is very important to know Arduino controlled system well.

DESCRIPTION

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.

The outcome obtained from the operation is given through the o/p port of an Arduino board to the LCD display of related data. The generated pulses from the board which is further fed to the driver circuit to get the preferred output to the fan.

Pulse Width Modulation (PWM):

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, we change, or modulate, that pulse width. If we repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency.

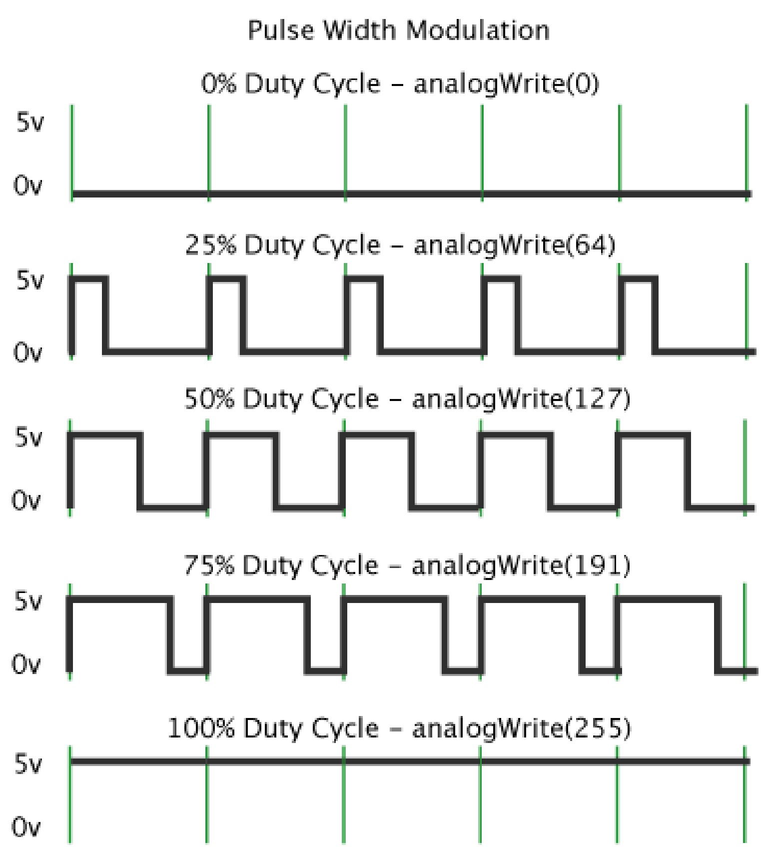


Fig.1 Pulse Width Modulation

Temperature Sensor:

We are using LM 35 as temperature sensor. LM 35 is a precision temperature sensor whose output is linearly proportional to Celsius Temperature. The LM35 is rated to operate from -55° Centigrade to 150° Centigrade with a linear scale factor of +10mv/° C

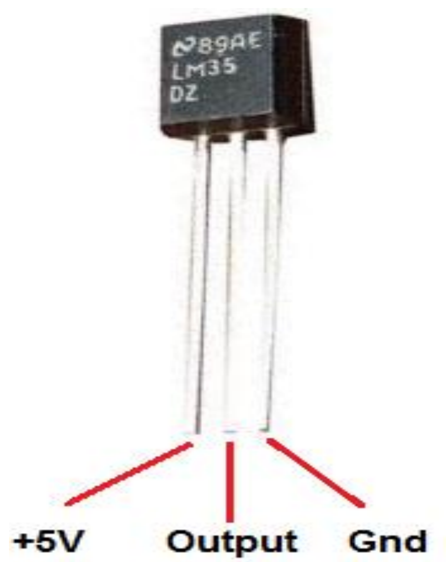


Fig.2 LM 35 temperature sensor

Features:

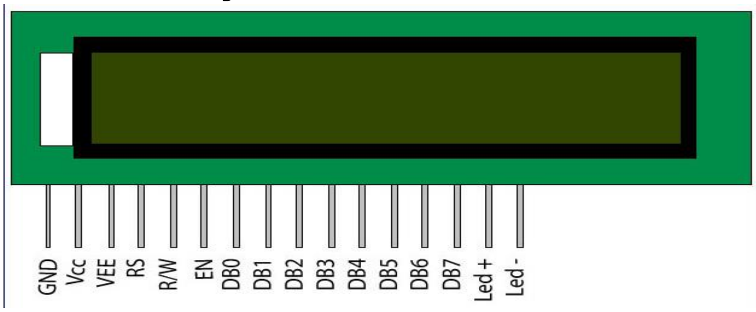
- Calibrated directly in degree Celsius (centigrade)
- Linear +10.0 mV/ degree Celsius
- 0.5 degree Celsius accuracy (at +25degree Celsius)
- Rated for full -55 to +150 degree Celsius range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 Micro ampere current drains
- Low self-heating, 0.08 degree Celsius in still air
- Nonlinearity only +/- 1/4 degree Celsius typical
- Low impedance output, 0.1 Ohm for 1mA load

Liquid Crystal Display:

The LCD is a dot matrix liquid crystal display that displays alphanumeric characters and symbols. 16X2 LCD digital display has been used in the system to show the room temperature. Liquid Crystal Display screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over sevensegments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & evencustom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Fig.3 16x2 LCD



Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Arduino UNO Basic Information:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide

its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

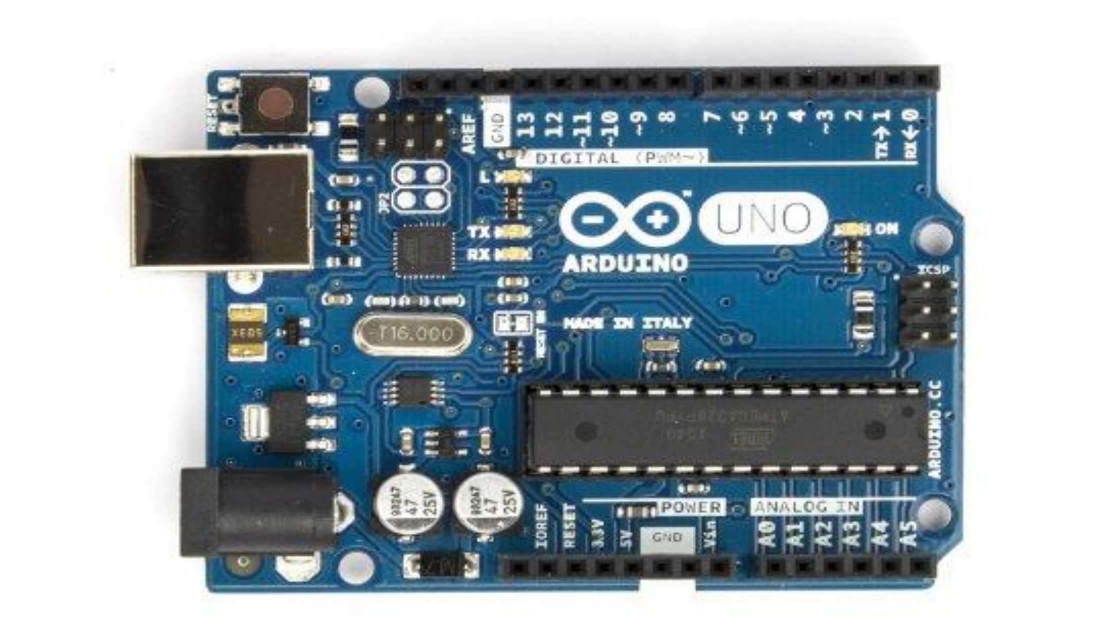


Fig.4 Arduino UNO

SPECIFICATIONS

Microcontroller	: ATmega328
Operating Voltage	: 5V
Input Voltage (recommended)	: 7-12V
Input Voltage (limits)	: 6-20V
Digital I/O Pins	: 14 (of which 6 provide PWM output)
Analog Input Pins	:6
DC Current per I/O Pin	:40 mA
DC Current for 3.3V Pin	:50 mA
Flash Memory	:32 KB of which 0.5 KB used by : Bootloader
SRAM	:2 KB
EEPROM	:1 KB
Clock Speed	:16 MHz

General Pin functions:

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN**: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**: Ground pins.
- **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **Reset**: Typically used to add a reset button to shields which block the one on the board.

Special Pin Functions

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

- **Serial:** pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM**(**P**ulse **W**idth **M**odulation) 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function.
- **SPI**(**S**erial **P**eripheral **I**nterface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI**(**T**wo **W**ire **I**nterface): A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- **AREF**(**A**nalog **REF**erence: Reference voltage for the analog inputs.

Communication:

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows serial communication on any of the Uno's digital pins.

Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

Code:

```
#include <LiquidCrystal.h>

LiquidCrystallcd(7,6,5,4,3,2);

inttempPin = A1;   // the output pin of LM35

int fan = 11;       // the pin where fan is

int led = 8;        // led pin

int temp;

inttempMin = 30;   // the temperature to start the fan

inttempMax = 70;   // the maximum temperature when fan is at
100%

intfanSpeed;

intfanLCD;


void setup() {

pinMode(fan, OUTPUT);

pinMode(led, OUTPUT);

pinMode(tempPin, INPUT);

lcd.begin(16,2);

}


void loop() {

temp = readTemp();     // get the temperature

if(temp <tempMin) {   // if temp is lower than minimum temp

fanSpeed = 0;      // fan is not spinning

digitalWrite(fan, LOW);
```

```
   }
if((temp >= tempMin) && (temp <= tempMax)) {  // if temperature
is higher than minimum temp
fanSpeed = map(temp, tempMin, tempMax, 32, 255); // the actual
speed of fan
fanLCD = map(temp, tempMin, tempMax, 0, 100);  // speed of fan
to display on LCD
analogWrite(fan, fanSpeed);  // spin the fan at the fanSpeed
speed
   }


if(temp >tempMax) {        // if temp is higher than tempMax
digitalWrite(led, HIGH);  // turn on led
   } else {                   // else turn of led
digitalWrite(led, LOW);
   }


lcd.print("TEMP: ");
lcd.print(temp);      // display the temperature
lcd.print("C ");
lcd.setCursor(0,1);   // move cursor to next line
lcd.print("FANS: ");
lcd.print(fanLCD);    // display the fan speed
lcd.print("%");
delay(200);
lcd.clear();
}
```

```
intreadTemp() {  // get the temperature and convert it to
celsius

temp = analogRead(tempPin);

return temp * 0.48828125;

}
```

Circuit Diagram:

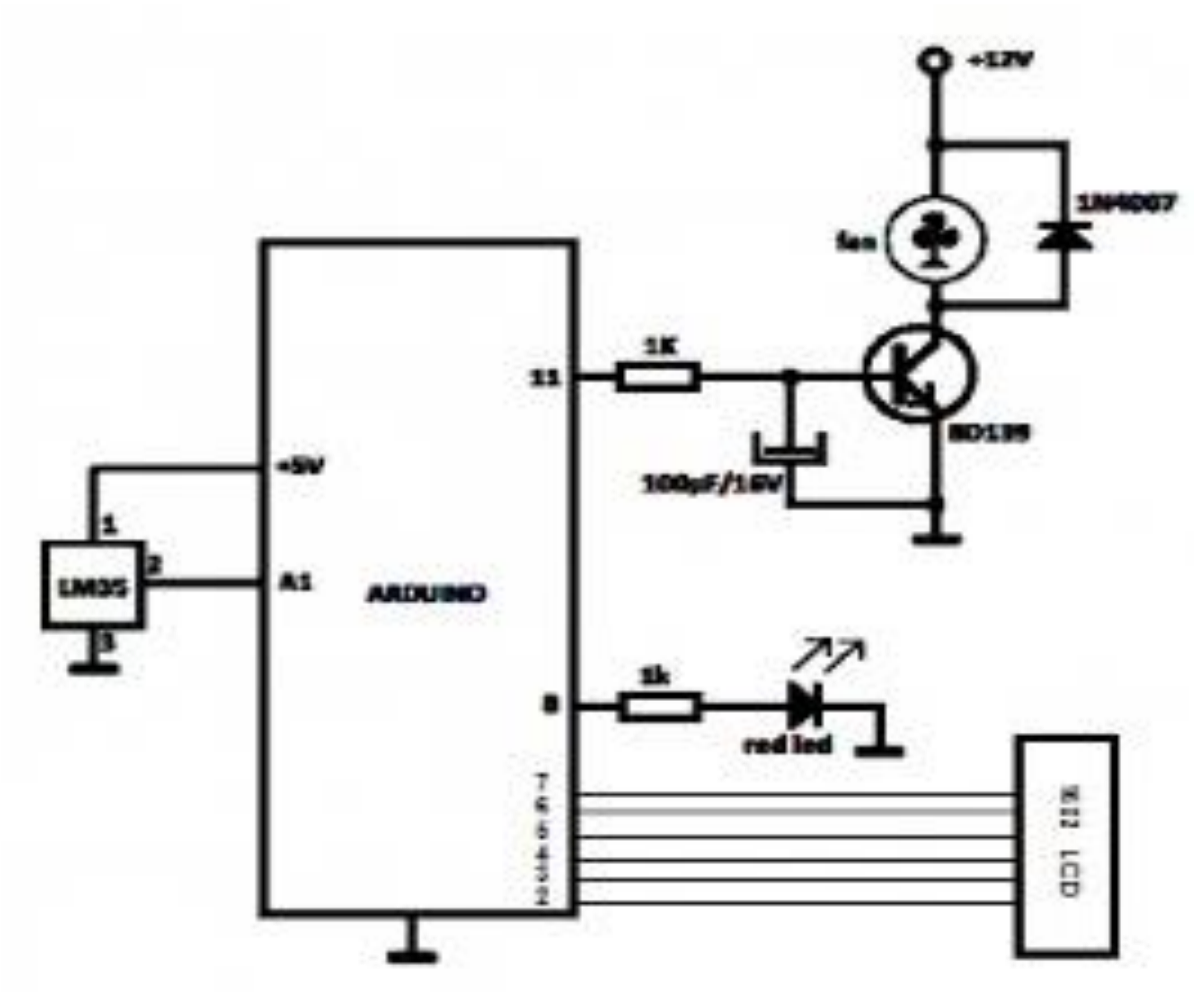


Fig.5 Circuit diagram of Temperature based fan speed controller

Block Diagram:

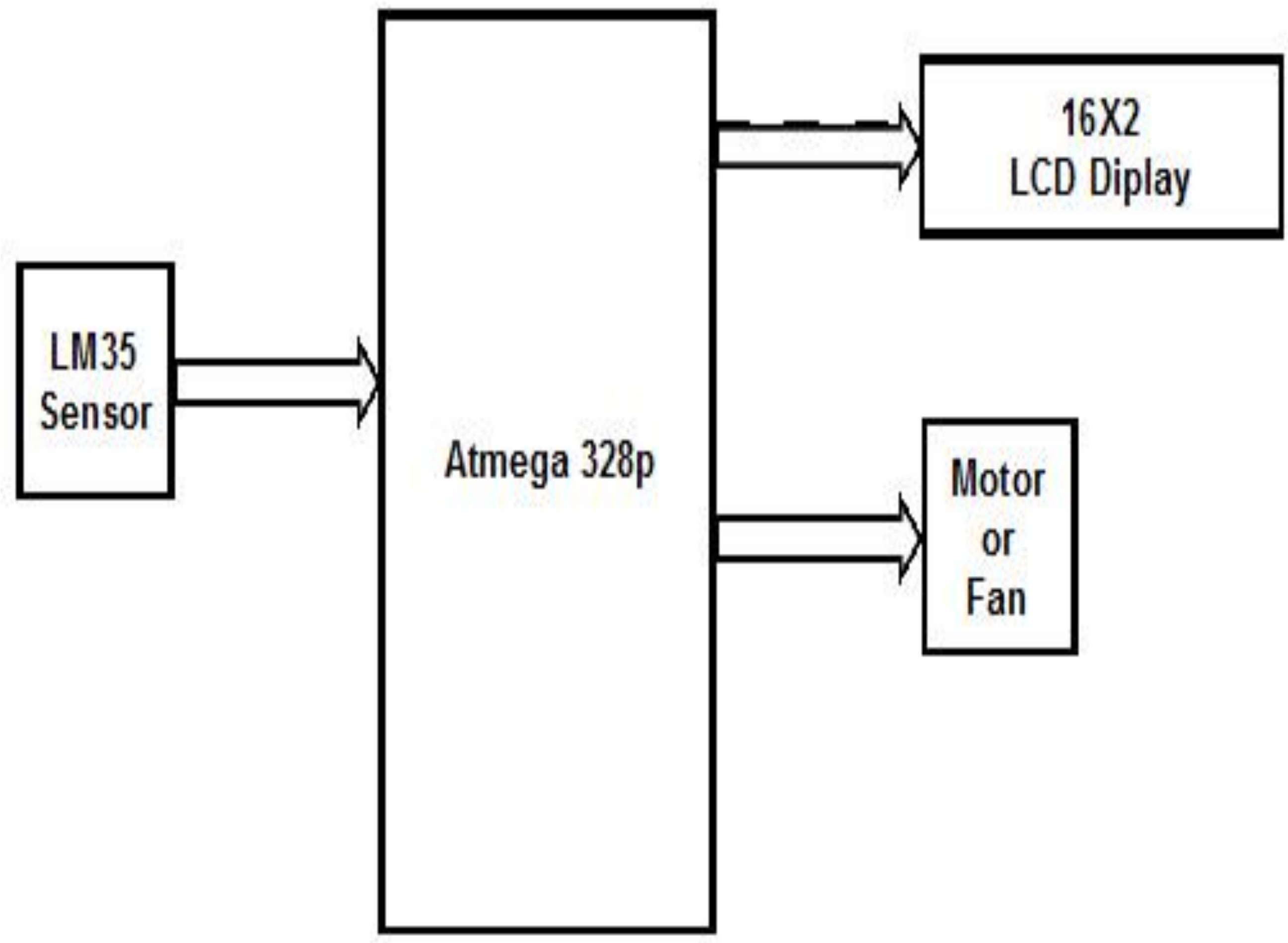


Fig.6 Block diagram of Temperature based fan speed controller

Description:

I used an LCD shield to display the current temperature and speed of the fan, but you can use the circuit without the LCD display. You also need to select the transistor by the type of fan that you use. In my case I used the well-known BD139 transistor and a 9V battery to provide power to the fan and transistor. The LM35 temperature sensor and red led are powered with 5V from the Arduino board.

As you can see in the sketch on the first line I included the LiquidCrystal library (header) that includes useful functions to use when an LCD is connected to the Arduino board. Then I set the pins for the sensor, led and fan.

The most important part is to set the variables temp Min and temp Max with your desired values. Temp Min is the temperature at which the fan starts to spin andtemp Max is the temperature when the red led lights warning you that the maximum temp was reached. For example if you set tempMin at 30 and tempMax at 35 then the fan will start spinning at 30°C and reach its maximum speed at 35°C.

We store the temperature value in the temp variable and then use some if() functions to check if temp is lower than tempMin and if so let the fan OFF (LOW). The next if() is to check if temperature is higher than the minTemp and lower than the tempMax and if so then use the map() function to re-map the temp value from one value to another. In our case fanSpeed will have a value of 32 at tempMin and 255 at tempMax. These values are used to control the speed of the fan using PWM and the analog Write().

The fan LCD re-maps the temp to allow the display of fanSpeed in a 0 to 100% range so you can say that the speed of the fan is directly dependent of the LM35's temperature. When the temperature reaches the value set in tempMax the fan will be at its maximum spinning velocity and the LCD will display FANS: 100% even though the temperature might increase above tempMax.

The rest of the explanation can be read in the comments area of the Arduino sketch.

In the next project I will make a temperature protection circuit that will turn off the power of equipment when its temperature has reached a certain value.

Application:

1. Temperature based fan speed controller is useful for cooling the processor in the laptops and personal computers "more efficiently". Generally fan in laptop comes with only two or three possible speeds. So it results in more power consumption.

2. The fan designed in this project, has different values of speed according to temperature change. This can be also used in small scale industries for cooling the electrical/mechanical equipment. The whole circuit except motor and fan can be manufactured on a single PCB, and it can be used for temperature based control operations.

Advantages:

1. This project can be used in Home.
2. This project can be used in Industry.
3. This will help in saving the energy / electricity.
4. To monitor the environments that is not comfortable, or possible, for humans to monitor, especially for extended periods of time.
5. Prevents waste of energy when it’s not hot enough for a fan to be needed.
6. To assist people who are disabled to adjust the fan speed automatically.

Disadvantages:

1. It can only be maintained by technical person. Thus, it becomes difficult to be maintained.
2. Due to temperature variation, after sometimes its efficiency may decrease.

Future Scope:

1. We can monitor more parameters like humidity, light and at the same time control them.
2. We can send this data to a remote location using mobile or internet.
3. We can draw graphs of variations in these parameters using computer.
4. When temperature exceeds the limit, a call will be dialed to the respective given number by an automatic Dialer system.

REFERENCES:

https://www.arduino.cc/en/Main/Software
www.ti.com/product/LM35
www.learningaboutelectronics.com
https://components101.com/lm35-temperature-sensor
https://circuitdigest.com/.../16x2-lcd-display-module
https://www.engineersgarage.com/.../16x2-lcd-module
https://www.build-electronic-circuits.com/how-transistor