Project-2

Automatic Temperature Control System Using Arduino

Submitted by Kothuri.Venkata Mahalakshmi

Automatic Temperature Control System Using Arduino

Abstract

Automatic temperature control system is an important application used in almost all modern gadgets and smart homes. The system for controlling temperature automatically is achieved by using Arduino-Uno based microcontroller system. Arduino Uno due to its increased popularity *finds* its varied range of applications. Temperature sensor LM35 and Arduino Uno are the hardware used interfaced with computer, and the temperature is controlled in the room. Temperature is displayed on LCD display employing A1 pin of hardware with the help of analog pin utilizing pulse width modulation (PWM). We have designed temperature control as an automatic system that has been not attempted before the way it has been implemented.

Keywords Temperature control *Arduino Uno
*Temperature sensor
*LCD display

Introduction

Temperature control becomes an important task in many of automatic operations. There are sensors, right from simple to smart sensors that are used for detecting the temperature. The environmental monitoring application, room temperature control are few of popular examples of temperature control. Now, with the advent of new technologies-hardware and software support temperature can be controlled, monitored, and recorded more flexibly and with the programmable ways. Information and communication technology (ICT) or smart appliances are using some sort of temperature control; this may be arti*fi*cial intelligence (AI)-based refrigerator or washing machine.

Microcontroller-based temperature control has become so important that it acts as benchmark for testing and simulation of particular sensors for detection and monitoring of temperature

automatically. Various types of projects like minor projects as well as major projects are carried out on suitable hardware and software platform.

This paper presents an application of control theory using ICT and hardware-based temperature control including design of a circuit (hardware) and implementation and testing on Arduino Uno board. The test results are displayed with the help of LCD display. The program is written in Arduino IDE and facilitates the display of temperature in degree centigrade and also in Fahrenheit. The Arduino Uno board facilitates the temperature measurements input to the fan and cooling system ON/OFF that is automatically done based on varied values of temperature.

METHODOLOGY:

Hardware Implementation:

Hardware implementation was obtained on Arduino IDE interfaced with P-IV computer. Data flow and block diagram of the hardware implementation are shownin Figs. 1 and 2, respectively.

Figures 1 and 2 are simple and self-explanatory where temperature sensors are connected with the help of Arduino and LCD display of 16 × 2 matrix. The fan was additionally connected for cooling mechanism so that automatic control could be achieved which is main objective of the proposed work. The hardware design isvery simple without any circuit complexity (Fig. 3).

We used temperature sensor IC LM35 that helps in generating a small voltage for detecting the change in temperature across the temperature sensor.

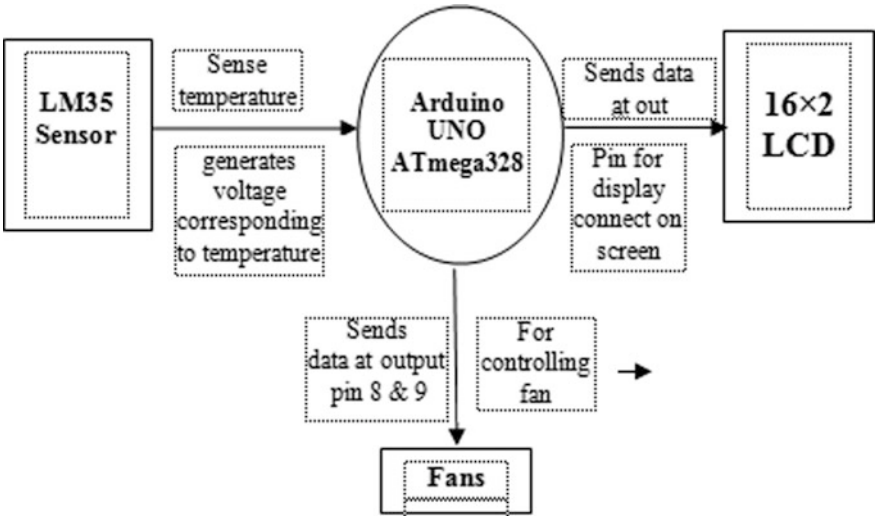


Fig. 1 Data flow in hardware implementation

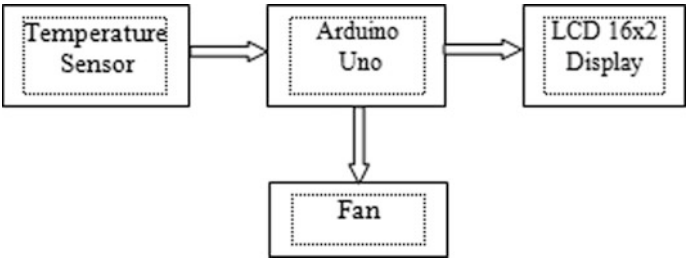


Fig. 2 Block diagram of temperature control hardware

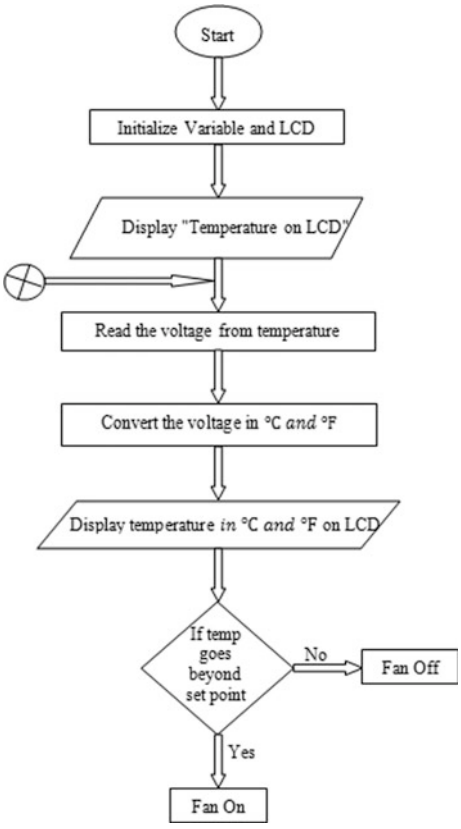


Fig.3 Flow chart of the control system

The generated voltage was continuous and analog signal generated through analog pin of Arduino hardware. Arduino (ATmega328) controller was used, and the voltage is taken as analog input in port 1 (A1) of Arduino Uno. The hardware reads the analog signal, and it is then converted into suitable digital output with the help of appropriate analog-to-digital converter (ADC) circuit that is inbuilt in it. ADC has maximum capacity of 10-bits output generating from 0 to 1023 (1024 combinations).

Digital data corresponding to analog input received through the port is converted and multiplied by a coefficient 0.488 just to normalize in centigrade. Similarly, suitable multipliers such as 1.8 and afterward adding with 32 used to convert the temperature measurement in Fahrenheit scale. The hardware sends the data to 16 × 2 LCD display which is connected with controller as shown in Fig. 4. Pin 1 and pin 2 are connected to ground and supply VCC, respectively, through Arduino for activating or switching ON the

LCD. Pin 3 enables through 10 K resister as adjustment of brightness value of LCD display.

So, the display can be made on both the scales of temperature. A control bit, either 0 or 1, is also sent by the Arduino to port 6 basically for providing control

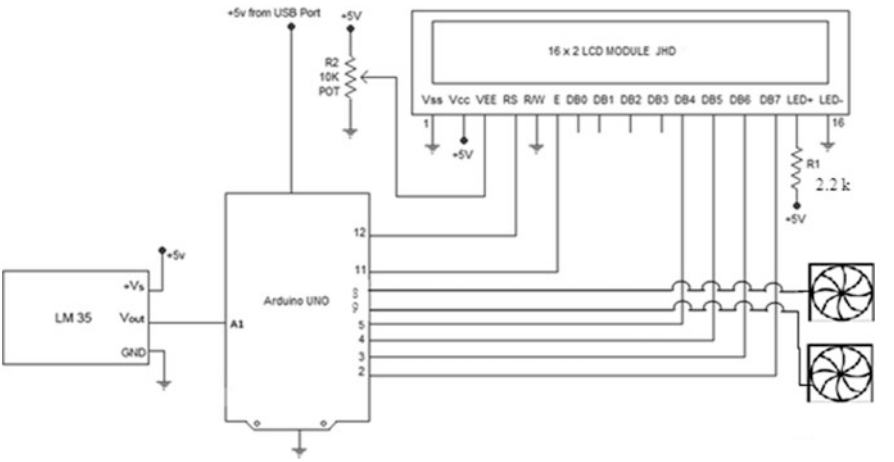
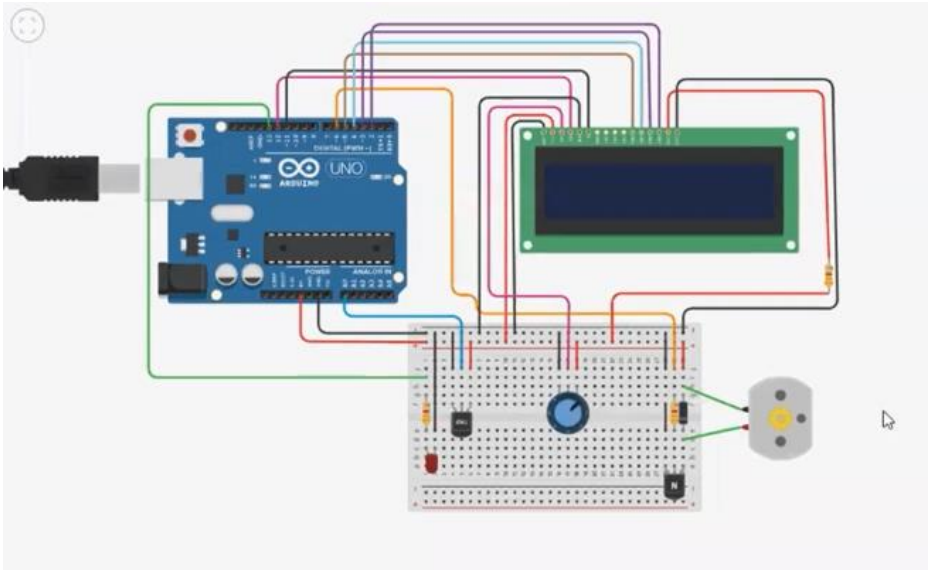


Fig. 4 Temperature control system

application. Pins 8 and 9 can be seen connected with fans just to adjust, normalize, and automatically control the temperate. Whenever temperature goes down after certain value, the fans will be OFF else it will keep running. This is how the control of temperature becomes automatic. Figure 3 shows the flowchart of the system.



The above circuit is the screen shot of the automatic temperature control system using arduino in tinkercad.

Text

1 (Arduino Uno R3)

```
const int temp_pin=A0;
const int heater_pin=13;
const int fan_pin=6;
float MinTemp=20,MaxTemp=25;
#include<LiquidCrystal.h>
LiquidCrystal LCD (12,11,5,4,3,2);
void setup()
{
  LCD.begin(16,2);
  pinMode(heater_pin,OUTPUT);
  pinMode(heater_pin,OUTPUT);
  LCD.print("Room temp (c):");
  LCD,setCursor(1,1);
  LCD.print(MinTemp);
  LCD.print("-");
  LCD.print(MaxTemp);
  delay(1000);
}
void loop()
{
  float Eqv_volt,SensorTemp;
  Eqv_volt=analogRead(temp_pin)*5.0/1023;
  SensorTemp=100.0*Eqv_volt-50.0;
  LCD.clear();
```

Serial Monitor

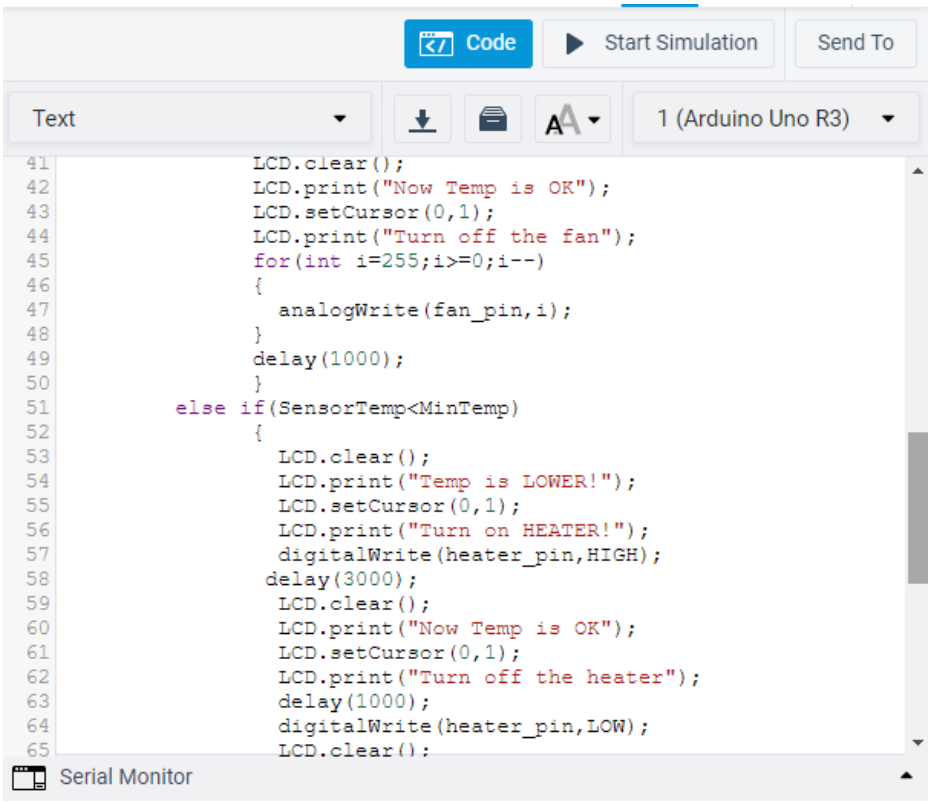
Code | Start Simulation | Send To

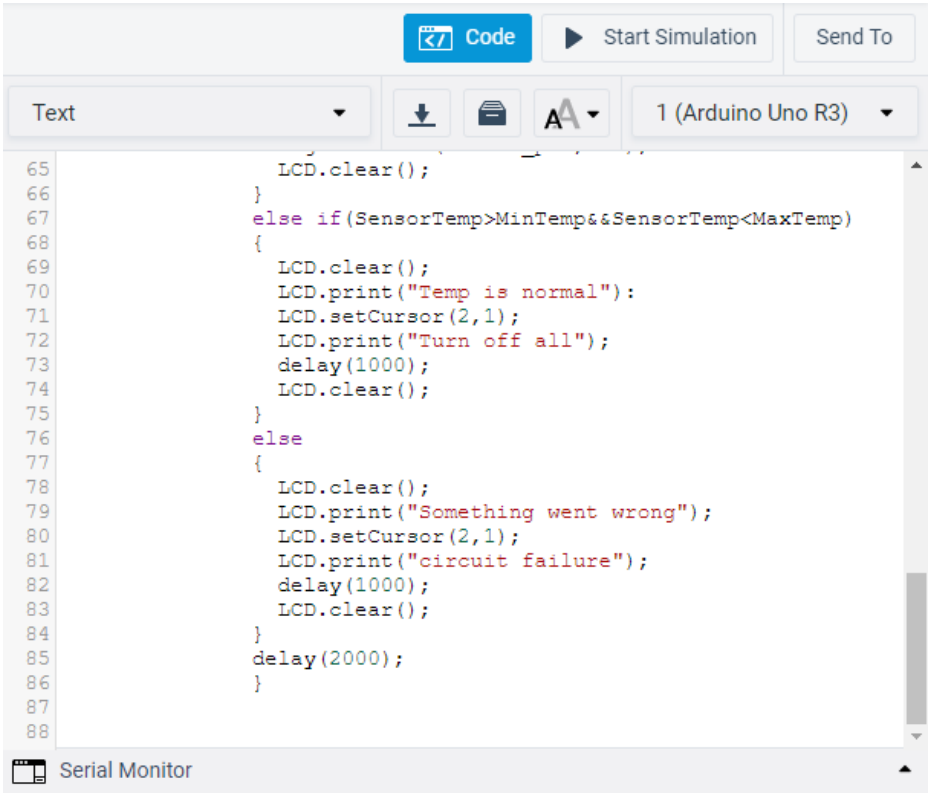
Text

1 (Arduino Uno R3)

```
void loop()
{
  float Eqv_volt,SensorTemp;
  Eqv_volt=analogRead(temp_pin)*5.0/1023;
  SensorTemp=100.0*Eqv_volt-50.0;
  LCD.clear();
  LCD.print("sensor reading ");
  LCD.setCursor(2,1);
  LCD.print(SensorTemp);
  LCD.print("C");
  delay(1000);
  if(SensorTemp>MaxTemp)
  {
    LCD.clear();
    LCD.print("Temp is higher!");
    LCD.setCursor(0,1);
    LCD.print("Turn on the fan!);
              for(int i=0;i<=255;i++)
              {
                analogWrite(fan_pin,i);
              }
              delay(1000);
              LCD.clear();
              LCD.print("Now Temp is OK");
              LCD.setCursor(0,1);
```

Serial Monitor


Code
Start Simulation
Send To
Text
1 (Arduino Uno R3)
LCD.clear();
LCD.print("Now Temp is OK");
LCD.setCursor(0,1);
LCD.print("Turn off the fan");
for(int i=255;i>=0;i--)
{
analogWrite(fan_pin,i);
}
delay(1000);
}
else if(SensorTemp<MinTemp)
{
LCD.clear();
LCD.print("Temp is LOWER!");
LCD.setCursor(0,1);
LCD.print("Turn on HEATER!");
digitalWrite(heater_pin,HIGH);
delay(3000);
LCD.clear();
LCD.print("Now Temp is OK");
LCD.setCursor(0,1);
LCD.print("Turn off the heater");
delay(1000);
digitalWrite(heater_pin,LOW);
LCD.clear();
Serial Monitor



The above images are the screen shots of code for automatic temperature control system using arduino doni in tinkercad.

Results:

However, experimental setup was done and lot of temperature measurement was recorded with suitable displays. Few of sample displays and the observations are presented here with brief discussion.

Mainly, we have two outputs in this work: one for displaying the temperature automatically on LCD display and second was even important that is for automatic switching ON/OFF of fans so as to monitor the temperatures on automatic basis.

LCD display produces the output of temperature as well as the status of fans. For example, in Fig. 4, it can be seen that 28 degree centigrade is being displayed by the LCD display along with status of fan as ON. Actually, the running condition of fans depends on the threshold value set. We set 25 degree centigrade as the value, and therefore, it can be seen that for 28 degree centigrade fan is running (i.e., in ON condition). Obviously, for below, for example 23 degree

centigrade, the fan will go OFF. The interface of display with bread board and Arduino hardware can be seen in Fig. 5.

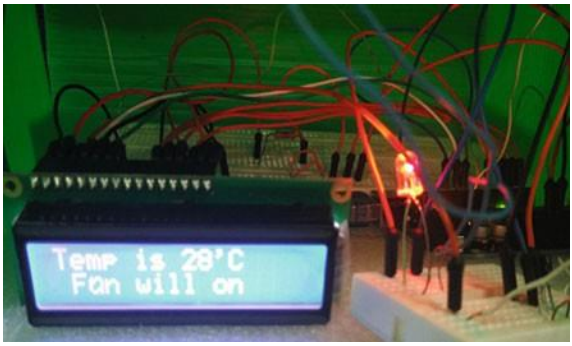


Fig. 5
Display of a temperature and status of fan in automatic control of temperature

Conclusions

I have designed a simple method of temperature control system automatically. Utilizing the concept cooling after certain temperature, Arduino-based hardware along with display was realized in hardware. Few samples are shown in result, but any value of temperature can be generalized in this work. The work is focused mainly on temperature control, and no other parameter is involved. This seems to be robust way of handling only temperature control on automatic basis. This can be extremely useful for persons of physical disability. Soft computing method could be used to make it more robust and fuzzy controlled.

Submitted by

K.V.MahaLakshmi.