## **Project Name: Smart vegetable room temperature and humidity based fan monitoring system**

## Overview: Smart vegetable room temperature and humidity based fan monitoring system

In this tutorial, I will show you how to make a **smart vegetable room humidity** **temperature-based fan speed controller using Arduino.** It is a very useful project and has a wide range of applications. A **temperature and humidity controlling** system is required in different vegetable room like **cabbage rooms, potato room, Broccoli room, okra room, cucumber room, fruits room, cauliflower room, eggplant room, other houses,** etc.

So this basic project can be very useful in understanding how we can control the humidity and temperature of a vegetable room at your home. You can also take this as a DIY project which has a wide range of use. Because this is an **automatic fan speed controller** that will **act to the temperature changes and humidity variation**

**Problem solving**

**This my project it will be able to solve many problem created by high temperature and humidity form bad collection/stock room.**

**Their quality is affected by hot weather; for example radishes become pithy and lecture bolts producing unwanted seed stalks. Many others cool-season vegetable such as cabbage, broccoli ,etc. so my project is needed to implement in vegetable room to decrease temperature to prevent vegetable from a vegetable spoilage before transform to a consumer.**

## **Components Required**

1. Arduino
2. Led
3. DHT11 Temperature & Humidity Sensor
4. 16×2 LCD Display
5. 2N2222 Transistor
6. buzzer
7. 12 Volt DC Fan
8. Breadboard
9. 12 Volt Power Supply
10. Jumper Wires

## How does it Work?

## In the **first part,** the **DHT22 sensor** senses the surrounding temperature and humidity. In the **second part**, the sensor’s output is taken and the temperature value is converted to the Celsius scale. Similarly, We have programmed our Arduino to generate **PWM (pluse width modulation)** and put it at the **base** terminal of the **transistor**. The working of this project is divided into three parts: then finally the transistor generates voltage with respect to the PWM value.Hence the fan speed is controlled by using PWM signals. Likewise,the **last part** of the system shows the temperature and fan speed on **16×2** **LCD Display**.

A detailed demo about the DHT11 on [**temperature and humidity measuring system**](https://circuitdiagrams.in/temperature-and-humidity-monitoring-system/) where you will find the working principle of the DHT11 sensor.

The analog value is then applied to the analog to digital converter pin of the Arduino. After that the analog value will be converted into digital value using successive approximation method. When the temperature value cross the threshold value, Arduino sends a command signal to the motor driver using PWM output to switch on the dc fan.

We have created a PWM signal at the PWM pin of the Arduino board and applied it at the base terminal of the 2N2222 NPN transistor. Then transistor creates a voltage according to the PWM input given.

## **Circuit Connection: Temperature Controlled Fan**

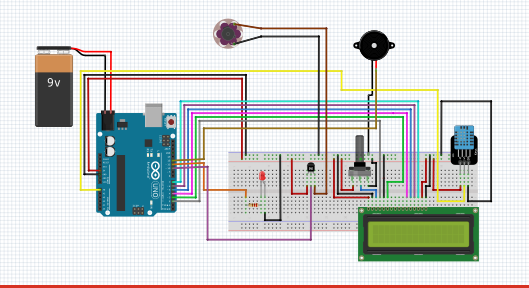
The circuit connections of **Temperature Based Automatic Fan Speed Controller using Arduino** is very easy. Here **16×2 LCD** is directly interfaced to Arduino using **A4 and A5** pins. LCD is used for displaying temperature and Fan Speed Status. The DHT11 sensor module is connected to the **A0** pin of Arduino. Similarly, the **D9** pin is used for controlling fan speed through the **2N2222 NPN transistor**.

**Please refer to the below table. It gives you an idea of what PWM valu will be the speed of the fan. You can change values according to your needs in the coding part.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature** | **Duty cycle** | **PWM(pulse width modulation)** | **Fan speed** |
| Less than 28 | OFF | 0 | OFF |
| 28 | 20% | 51 | 20% |
| 29 | 4O% | 102 | 4O% |
| 30 | 60% | 153 | 60% |
| 31 | 80% | 204 | 80% |
| Great than 32 | 100% | 255 | 100% |

## **Program Code**

This is a complete program source code for Temperature Based **Automatic Fan Speed Controller using Arduino**. Before uploading the code, please refer to the **PWM** and **Speed** table above. Comment the **DHT22** line and uncomment **DHT11** if you are using the DHT11 sensor instead of DHT22.



#include "DHT.h" //Library for DHT11 sensor

#include<LiquidCrystal.h> //Library for 16x2 LCD display

LiquidCrystal lcd(2,3,4,5,6,7); //Define 16x2 LCD display pins

#define DHTPIN A1 //Define analog pin A0 as DHT11 sensor pin

#define DHTTYPE DHT11

#define pwm 10 //Define digital pin 10 as PWM output

#define BUZZER 12;//You can use either D9 or D10 as a PWM output

#define LED 1;

byte degree[8] =

{

0b00011,

0b00011,

0b00000,

0b00000,

0b00000,

0b00000,

0b00000,

0b00000

};

//Initialize DHT sensor for normal 16MHz Arduino

DHT dht(DHTPIN, DHTTYPE);

void setup()

{

digitalWrite(12,OUTPUT);

digitalWrite(1,OUTPUT);

lcd.begin(16, 2);

lcd.createChar(1, degree);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("WELCOM TO MYPROJECT");

lcd.setCursor(0,1);

lcd.print("MY NAME IS PHILIMINE");

delay(5000);

lcd.clear();

analogWrite(pwm, 255);

Serial.begin(9600);

dht.begin();

}

void loop()

{

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h)|| isnan(t))

{

Serial.println("Failed to Read From DHT Sensor");

return;

}

digitalWrite(1,HIGH);

lcd.setCursor(0,1);

lcd.print("Temp=");

lcd.print(t); //Showing temperature on LCD display

lcd.print(" \*C");

lcd.setCursor(20,1);

lcd.print("Humidity =");

lcd.print(h); //showing humidity on LCD display

lcd.print(" %");

lcd.setCursor(0,0);

if(t<20)

{

analogWrite(9,0);

lcd.print("Fan Speed= 0% ");

digitalWrite(10,LOW);

digitalWrite(12,LOW);

delay(1000);

}

else if(t>20&&t<24)

{

analogWrite(pwm, 51);

lcd.print("Fan Speed= 20% ");

digitalWrite(12,LOW);

delay(1000);

}

else if(t>24&&t<26)

{

analogWrite(pwm, 102);

lcd.print("Fan Speed= 40% ");

digitalWrite(12,LOW);

delay(1000);

}

else if(t>26&&t<28)

{

analogWrite(pwm, 153);

lcd.print("Fan Speed= 60% ");

digitalWrite(12,HIGH);

delay(1000);

}

else if(t>28&&t<32)

{

analogWrite(pwm, 204);

lcd.print("Fan Speed= 80% ");

digitalWrite(12,LOW);

delay(1000);

}

else if(t>34)

{

analogWrite(pwm, 255);

lcd.print("Fan Speed= 100% ");

analogWrite(12,1);

digitalWrite(12,HIGH);

delay(2000);

}

digitalWrite(11,HIGH);

delay(2000);

} analogWrite(pwm, 153);

lcd.print("Fan Speed= 60% ");

digitalWrite(11,HIGH);

delay(500);

}

else if(t==39)

{

analogWrite(pwm, 204);

lcd.print("Fan Speed= 80% ");

digitalWrite(11,HIGH);

delay(500);

}

else if(t>39)

{

analogWrite(pwm, 255);

lcd.print("Fan Speed= 100% ");

digitalWrite(11,LOW);

delay(2000);

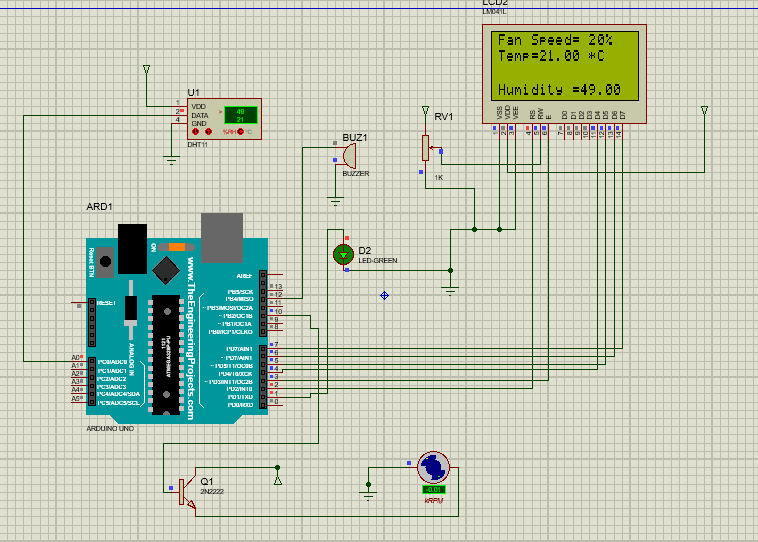
}

delay(1000);

}

**Proteus simulation**

The proteus is a simulation and design software use for lab centre electronic for electrical and electronic design circuits. Proteus is the best simulation software for various with microcontroller it is mainly common software because of availability of almost all microcontroller in it.

****

**Thank you!!**