AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112

BONAFIDE CERTIFICATE

This is to certify that the Open Lab project report entitled “**Automated Fan”.**

Submitted by

CB.EN.U4ECE19051	**SRIRANGAM SAI TEJA**
CB.EN.U4ECE19052	**SUNKAVALLI VINAY KUMAR**
CB.EN.U4ECE19058	**V.G.AKSHAYA**
CB.EN.U4ECE19059	**VIGAASINY.K.P**

In partial fulfillment of the requirements for the award of the **Degree of Bachelo**r **of Technology** in **ELECTRONICS AND COMMUNICATION ENGINEERING** is a Bonafide record of the work carried out at Amrita School of Engineering, Coimbatore.

Chairperson ECE
Dr. M. Jayakumar

The project was evaluated by us on:

Open Lab Coordinator
Name :Mr.C.B.Rajesh
Designation :Assistant Professor

Open Lab Coordinator
Name : Mr.P.Sudheesh
Designation : Assistant Professor

CONTENTS:

ABSTRACT:

The Problem with conventional fan is that its speed has to be manually regulated whenever there is a change in temperature and many a times it is left running even if there is no one around, leading to wastage of electricity. Hence, our project focuses on automation of fan to overcome the limitations of manually controlled fans.

Arduino micro controller controls all the functions. The temperature and humidity sensor DHT11 senses the temperature and humidity, and converts it into an electrical signal, which is then forwarded to the microcontroller. A circuit with Arduino UNO as a main processor, DHT11 sensor, LDR sensor and few other electronic components is designed and implemented to control the turning ON/OFF of fans and its speed automatically.

INTRODUCTION:

Automatic system and automation are the requirement of today's technology.In automated fan, the fan is turned on/off with respect to the human detection rather than the use of manual switching system. Other function is to control the speed of a fan with respect of temperature, humidity and number of people in the room. In this project Arduino Uno forms the processing part, which firstly detect the number of humans with the use of LDR sensor and senses the temperature and the humidity with the use of DHT11. Arduino Uno senses the temperature and humidity and controls the speed with the set temperature and humidity. This is set by the user. The current temperature and humidity determine the speed of the fan(s).
On the other hand, when there is no human presence, the fan remains off. Number of fans turned ON depends on the number of people present in the room.

Hence, for turning on the fan there are three conditions to be true. First is human detection, second is temperature and third is humidity.
After the fan turns ON, its speed will change according to temperature and humidity. and number of fans turned ON/OFF depends on the people present in the room. Whenever the temperature and humidity increases fan’s speed will increase.

Arduino UNO:

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. The Uno contains a trace that can be cut to disable the auto-reset. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing).

Fig.1: Arduino UNO

DHT11 Sensor:

The DHT11 Digital Temperature and Humidity Sensor is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

Fig.2: DHT11 Sensor

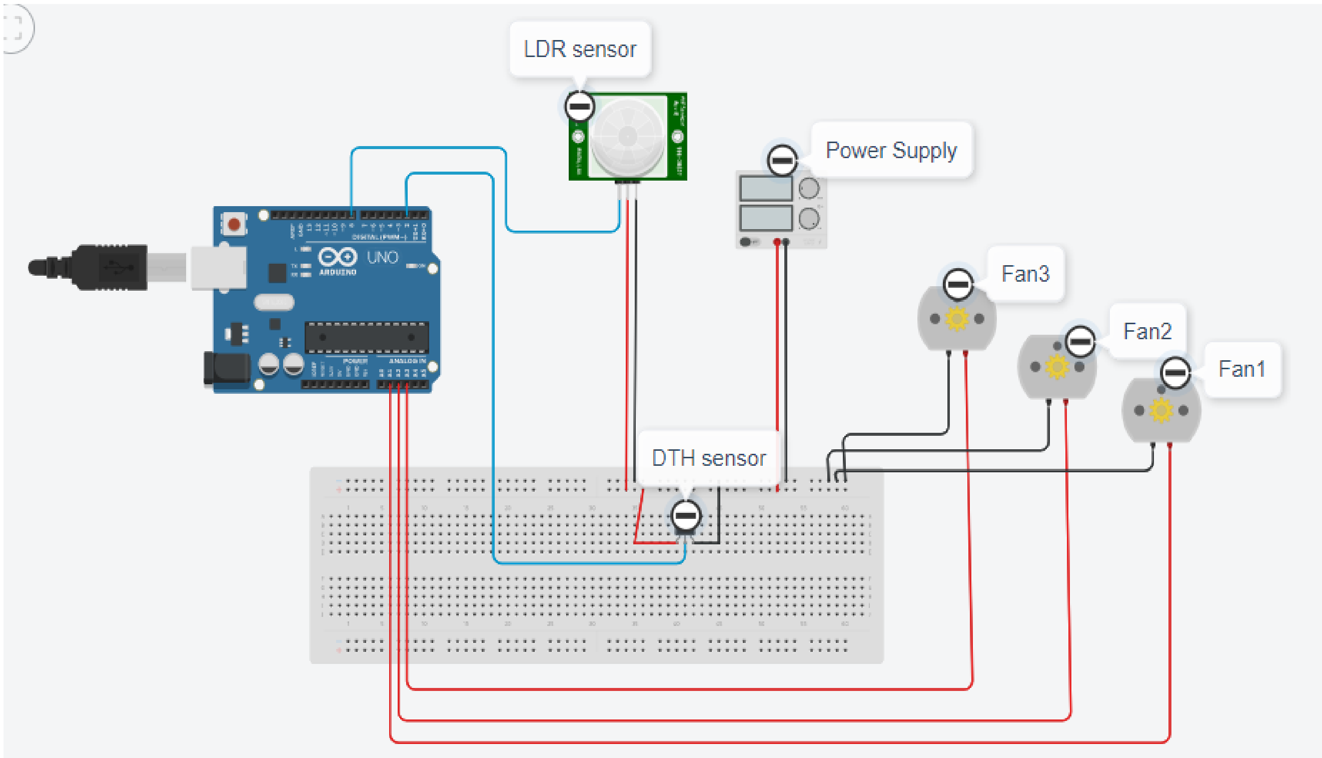
LDR Sensor:

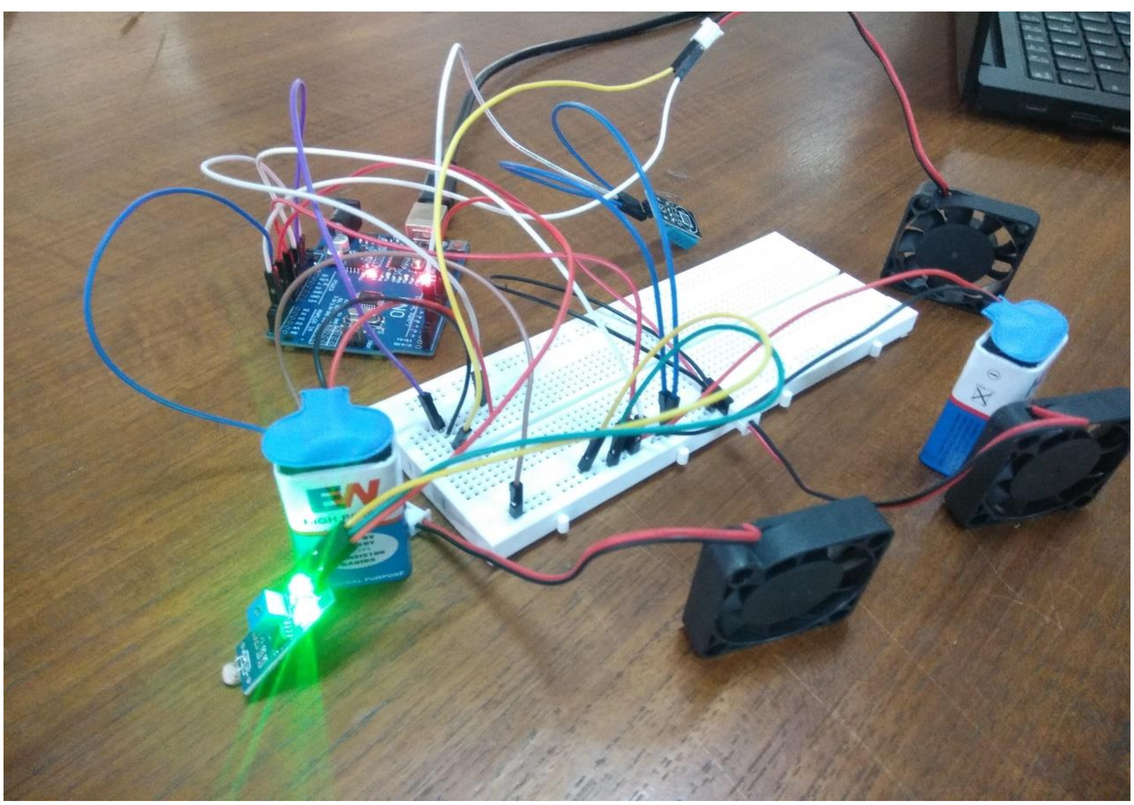
What is an LDR (Light Dependent Resistor) An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits. Light Dependent Resistors (LDR) are also called photoresistors.

Fig.3: LDR Sensor

Design and Implementation:

- Circuit:





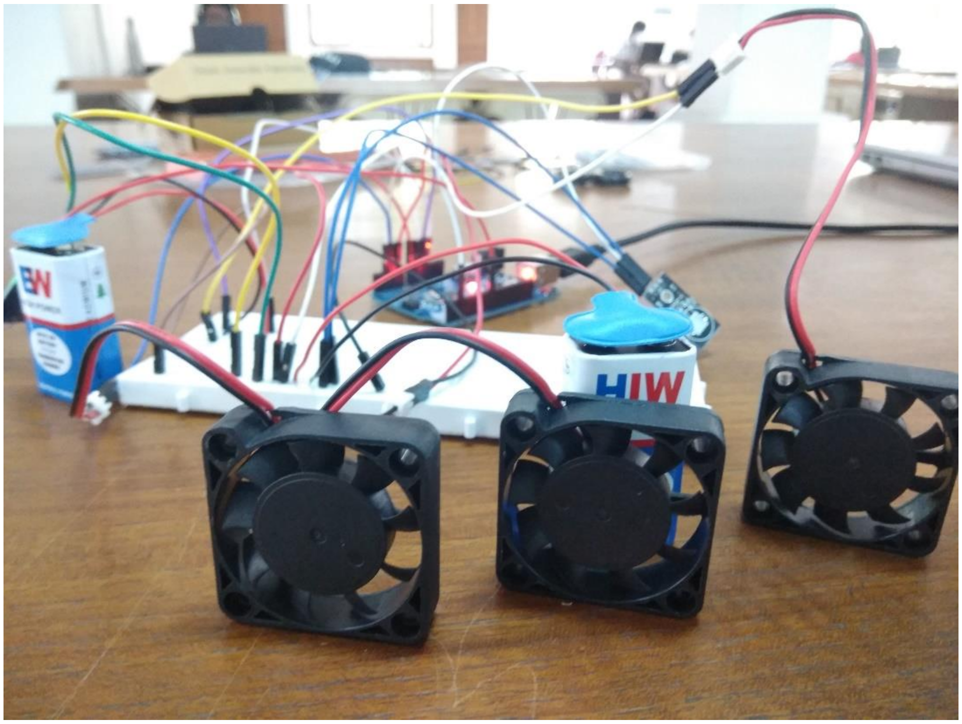


Fig.4: a) The Circuit diagram b) The Circuit connections with LDR, DHT11 and Arduino c) Circuit connections with 3 fans

- <u>Code</u>:

```
#include "DHT.h"
#define DHTPIN 2     // Digital pin connected to the DHT sensor
#define DHTTYPE DHT11   // DHT 11
DHT dht(DHTPIN, DHTTYPE);

int fan1=A1;
int fan2=A2;
int fan3=A3;
int B=8;
int count = 0;
int i;


void setup() {
Serial.begin(9600);
Serial.println(F("DHTxx test!"));
```

```
pinMode(fan1,OUTPUT);
pinMode(fan2,OUTPUT);
pinMode(fan3,OUTPUT);
pinMode(B,INPUT);
dht.begin();
}


void c()
{
 for(i=0;i<10;i++){
  if(digitalRead(B)>0){
  count++;
Serial.print("count= ");
Serial.println(count);
} delay(500);
}return count;
}

void loop() {
  // Wait a few seconds between measurements.
delay(2000);



  // Reading temperature or humidity takes about 250 milliseconds
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);


  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }


  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
```

```
float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F(" Humidity: "));
Serial.print(h);
Serial.print(F("%  Temperature: "));
Serial.print(t);
Serial.print(F("C "));
Serial.print(f);
Serial.print(F("F  Heat index: "));
Serial.print(hic);
Serial.print(F("C "));
Serial.print(hif);
Serial.println(F("F"));

c();

switch (count)
{
    case 0: analogWrite(fan2,0);
analogWrite(fan1,0);
            break;



   case 1:
   if (t >= 27 && t<30.25)
{ analogWrite(fan1,1000);
Serial.println("rpm=1000");
      }
  if (t >= 30.25 && t<30.5)
   {
analogWrite(fan1,2000);
Serial.println("rpm=2000");
  }
  if (t >= 30.5 && t<30.75)
   {
analogWrite(fan1,3000);
Serial.println("rpm=3000");

  }
  if (t >= 30.75 && t<31)
   {
analogWrite(fan1,4000);
Serial.println("rpm=4000");

  }
```

```
  if (t >= 31.25 && t<34)
{  analogWrite(fan1,5000);
Serial.println("rpm=5000");
   }
        break;
    case 3:
   if (t >= 27 && t<30.25)
{ analogWrite(fan2,1000);
analogWrite(fan1,1000);
Serial.println("rpm1=1000");
Serial.println("rpm2=1000");
      }
   if (t >= 30.25 && t<30.5)
   {
analogWrite(fan2,2000);
analogWrite(fan1,2000);
Serial.println("rpm1=2000");
Serial.println("rpm2=2000");
  }



  if (t >= 30.5 && t<30.75)
   {
analogWrite(fan2,3000);
analogWrite(fan1,3000);
Serial.println("rpm1=3000");
Serial.println("rpm2=3000");

  }
  if (t >= 30.75 && t<31)
   {
analogWrite(fan2,4000);
analogWrite(fan1,4000);
Serial.println("rpm1=4000");
Serial.println("rpm2=4000");

  }
  if (t >= 31.25 && t<34)
{  analogWrite(fan2,5000);
analogWrite(fan1,5000);
Serial.println("rpm1=5000");
Serial.println("rpm2=5000");
}
   break;
```

```
   case 5:
  if (t >= 27 && t<30.25)
{ analogWrite(fan3,1000);
analogWrite(fan2,1000);
analogWrite(fan1,1000);
Serial.println("rpm1=1000");
Serial.println("rpm2=1000");
Serial.println("rpm3=1000");
     }
  if (t >= 30.25 && t<30.5)
  {
analogWrite(fan3,2000);
analogWrite(fan2,2000);
analogWrite(fan1,2000);
Serial.println("rpm1=2000");
Serial.println("rpm2=2000");
Serial.println("rpm3=2000");
  }



 if (t >= 30.5 && t<30.75)
  {
analogWrite(fan3,3000);
analogWrite(fan2,3000);
analogWrite(fan1,3000);
Serial.println("rpm1=3000");
Serial.println("rpm2=3000");
Serial.println("rpm3=3000");

 }
 if (t >= 30.75 && t<31)
  {
analogWrite(fan3,4000);
analogWrite(fan2,4000);
analogWrite(fan1,4000);
Serial.println("rpm1=4000");
Serial.println("rpm2=4000");
Serial.println("rpm3=4000");

 }
```

```
 if (t >= 31.25 && t<34)
   {

analogWrite(fan3,5000);
analogWrite(fan2,5000);
analogWrite(fan1,5000);

Serial.println("rpm1=5000");
Serial.println("rpm2=5000");
Serial.println("rpm3=5000");
}
   break;
    }

}
```

CONCLUSION:

The fan has been successfully automated. Fans are controlled using Arduino based on the LDR sensor and DHT11 sensor.

The number of fans turned ON/OFF is based on the number of people in the room which is detected using the LDR sensor.

The fan’s speed is automatically regulated based on the temperature and humidity which is sensed by DHT11 sensor.

Reference

- Moe Moe San , Cherry Kyaw Win & Khin Zar Mon, “Design and Simulation of Fan Speed Control using Arduino UNO and LM35DZ”, in International Journal of Engineering Research and Advanced Technology (IJERAT), Volume.6, Issue 6, June -2020.

- Nur Mohd Fadzli Bin Nordzi, “Automatic Ceiling fan controller based on temperature sensor”, dissertation, Dept. Elect. Eng., Universiti Teknologi PETRONAS, Malaysia, 2016.

- M.Kumaran, I.Vikram, S.Kishore Kumar, R. Rajesh Kumar, S. Lokesh, “Design of An Automatic Fan Speed Controlling System Using Arduino UNO”, in International Journal of Intellectual Advancements and Research in Engineering Computations, Volume 6, Issue 2