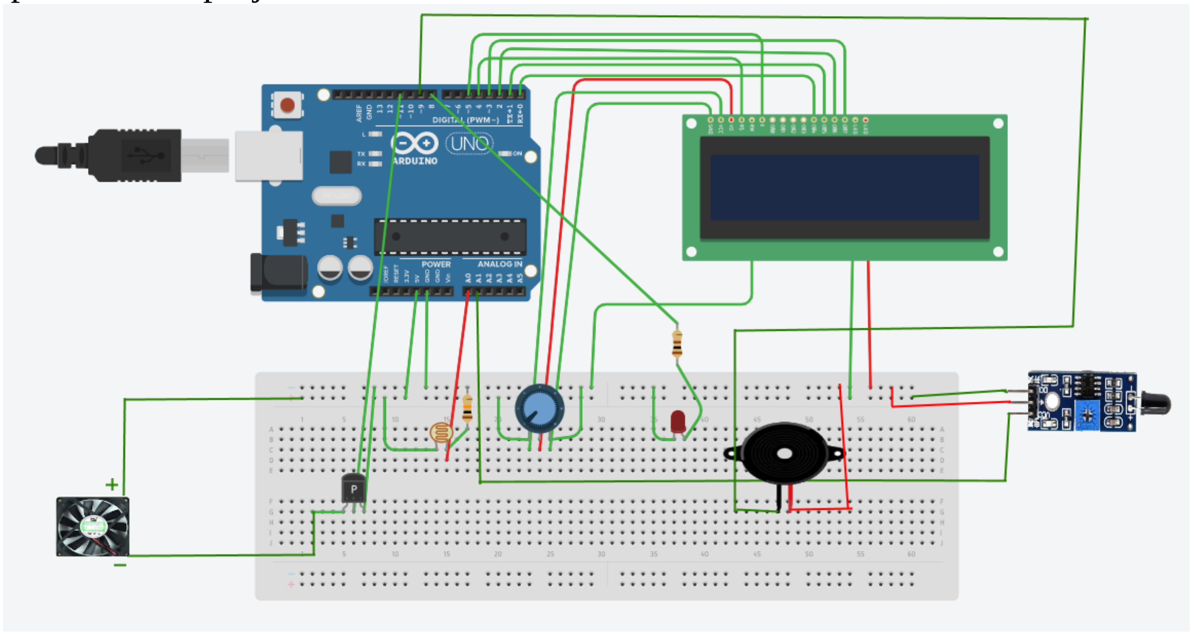
Contents:

Circuit Diagram

The following circuit diagram shows all the necessary connections required to implement this project.



Components Required

- ★ Arduino UNO
- ★ DHT11 Temperature and Humidity Sensor
- ★ Flame Sensor
- ★ LDR
- ★ LED
- ★ Buzzer
- ★ Breadboard
- ★ Power supply
- ★ 16 x 2 LCD Display
- ★ 10K Ohm Potentiometer
- ★ 100 Ohm Resistor
- ★ 10k Ohm Resistor
- ★ PnP Transistor
- ★ Connecting wires

Circuit Description

We will see the circuit design of DHT11, LDR, and Flame sensor interfacing with Arduino.

DHT11 Temperature and Humidity Sensor:

DHT11 is a part of DHTXX series of Humidity sensors. The other sensor in this series is DHT22. Both these sensors are Relative Humidity (RH) Sensor. As a result, they will measure both the humidity and temperature. Although DHT11 Humidity Sensors are cheap and slow, they are very popular among hobbyists and beginners.

The DHT11 Humidity and Temperature Sensor consists of 3 main components. A resistive type humidity sensor, an NTC (negative temperature coefficient) thermistor (to measure the temperature), and an 8-bit microcontroller, which converts the analog signals from both the sensors and sends out a single digital signal.

This digital signal can be read by any microcontroller or microprocessor for further analysis.

DHT11 Humidity Sensor consists of 3 pins: VCC, Data Out, and GND. The range of voltage for VCC pin is 3.5V to 5.5V. A 5V supply would do fine. The data from the Data Out pin is serial digital data.

DHT11 Sensor can measure a humidity value in the range of 20 - 95% of Relative Humidity (RH) and a temperature in the range of 0 - 50C. The sampling period of the sensor is 1 second.

All the DHT11 Sensors are accurately calibrated in the laboratory and the results are stored in the memory. A single wire communication can be established between any microcontroller like Arduino and the DHT11 Sensor.

Also, the length of the cable can be as long as 20 meters. The data from the sensor consists of integral and decimal parts for both Relative Humidity (RH) and temperature.

Flame Sensor: One can detect a flame or in a 760 nanometer wavelength of ~ 1100 nm range of the light. The detection angle is 60 degrees. Sensitivity adjustable (shown in blue digital potentiometer adjustment). The comparator output signal clean, good waveform, driving ability, more than 15mA. The working voltage is 3.3V-5V. The output format: digital switching outputs (0 and 1). A fixed bolt hole for easy installation.

LDR: This is a very small light sensor. It is called a light sensor or photocell or LDR (light dependent sensor) or photo-resistor. A photocell changes resistance depending on the amount of light it is exposed to. These little sensors make great ambient light triggers (when a light in the room turns on, do something).

Features:

- Light resistance: ~1k Ohm
- Dark resistance: ~10k Ohm
- Max voltage: 150V
- Max power: 200mW

Working methodology:

A simple project is built using Arduino UNO and DHT11 Humidity and Temperature Sensor, LDR, and Flame sensor. Here the DHT11 sensor display temperature and humidity on an LCD display. If the temperature is more than 33∘C, then the fan is automatically turned on. If the temperature is 33∘C or less then the fan will be automatically turned off. LDR detects the room light intensity. If there is not enough light in the room then the led automatically turns on. And the flame sensor detects fire. If any fire in the room is detected then the buzzer gives a signal.
If you want to use this library, you need to download this library separately and add it to the existing libraries of Arduino.

CODE:

```
#include <DHT.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(4, 5, 0, 1, 2, 3);
byte degree_symbol[8] = //defined symbol for degree notation
{
  0b00111,
  0b00101,
  0b00111,
  0b00000,
  0b00000,
  0b00000,
  0b00000,
  0b00000
};
float t = 0;
int DHTPIN = 12;
DHT dht(DHTPIN, DHT11);
int lightPin = A0; // select the input pin for LDR
int flamePin = A1; // select the input pin for flame sensor
int buzzerPin = 9; // defined output pin for buzzer
int ledPin = 8; // defined output pin for LED
int fanPin = 11; // defined output pin for DC Fan
int lightValue = 0;
int flameValue = 0;
float maxTemp = 30;
void setup() {
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(11, OUTPUT);
  //Serial.begin(9600);
  dht.begin();
  lcd.begin(16, 2);
  lcd.print("Temp: ");
  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.createChar(1, degree_symbol);
  lcd.setCursor(11, 0);
  lcd.write(1);
  lcd.print("C");
  lcd.setCursor(15, 1);
  lcd.print("%");
}
void loop() {
  float h = dht.readHumidity();
  t = dht.readTemperature();
  /*if (isnan(h) || isnan(t)) // Check if any reads failed and exit early (to try again).
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  } */
  lightValue = analogRead(lightPin);
  flameValue = analogRead(flamePin); // read the value from the sensor

  /*Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print("% Temperature: ");
    Serial.print(t);
    Serial.println("°C ");
    Serial.println(lightValue);
    Serial.println(flameValue)*/
  lcd.setCursor(6, 0);
  lcd.print(t);
  lcd.setCursor(10, 1);
  lcd.print(h);
  if (t > maxTemp) {
    digitalWrite(fanPin, LOW);
    if (flameValue < 500) {
      if (lightValue < 100) {
        digitalWrite(ledPin, HIGH);
      }
      else
      {
        digitalWrite(ledPin, LOW);
      }

      digitalWrite(buzzerPin, HIGH);
      //delay(2000);
    }
  }
  else
  {
    digitalWrite(fanPin, HIGH);
  }

  if (flameValue < 500) {
    if (lightValue < 100) {
      digitalWrite(ledPin, HIGH);
    }
    else
    {
      digitalWrite(ledPin, LOW);
    }

    if (t > maxTemp) {
      digitalWrite(fanPin, LOW);
    }
    else
    {
      digitalWrite(fanPin, HIGH);
          }
          digitalWrite(buzzerPin, HIGH);
          //delay(2000);
        }
        else
        {
          digitalWrite(buzzerPin, LOW);
        }

        if (lightValue < 100) {
          if (flameValue < 500)
          {
            digitalWrite(buzzerPin, HIGH);
          }
          else
          {
            digitalWrite(buzzerPin, LOW);
          }

          if (t > maxTemp) {
            digitalWrite(fanPin, LOW);
          }
          else
          {
            digitalWrite(fanPin, HIGH);
          }
          digitalWrite(ledPin, HIGH);
          //delay(2000);
        }
        else
        {
          digitalWrite(ledPin, LOW);
        }
        //delay(2000);
      }
```

Applications

- HVAC (Heating, Ventilation and Air Conditioning) Systems.
- Hospital.
- Home Automation Systems.

Discussions:

- We faced a problem to connect the 16*2 LCD with Arduino as the pinout type of this module wasn't usual to us.
- The male pins of the connecting wires that we bought are not wide enough so the connection often got loose.
- We connected the data pin of the DTH11 sensor to the 10th pin but in the Arduino code, we mentioned the 11th pin as the data pin for the sensor. So, it wasn't giving us any output and It took an hour to figure out the mistake.
- When we run the system for the first time, we noticed that the buzzer was defective.
- Sometimes the DHT11 sensor failed to show data.