

Introduction to Embedded Systems

Stabilized Temperature Box Project Report

Ahmet Oral (180709008)

Teacher: Assoc. Dr. Ufuk Bal

June 15 2022

Table of Contents

- Introduction..... 3**
- Required Components..... 3**
- Tools and Platforms..... 3**
- Component Examination 4**
 - DHT11 Heat Sensor4
 - One Channel Relay5
- Circuit Connections and Planning..... 6**
 - DHT11 to Arduino Uno6
 - Relay Connections6
- Coding..... 7**
- Photos of the Project..... 8**
- Running the Project.....10**

Introduction

In this project, my goal is to create a temperature stabilizer. The temperature can be chosen by user. To do this, I will need: strong lightbulb, Arduino, heat sensor, relay, a cut cable to connect relay and lightbulb and a box to put everything together. As a lightbulb I will use an old lightbulb that produces lots of heat. My Arduino board will be Arduino Uno and as a heat sensor I will use DHT11 sensor because it is cheap and good enough to be used in these kinds of projects.

The circuit will be running on 5 Volts which is enough to power the relay and DHT11. Relay and sensor will be connected to Arduino Uno. Software in Arduino will get temperature data from DHT11 and decide if the lightbulb should be Onn or Off.

Required Components for The Circuit

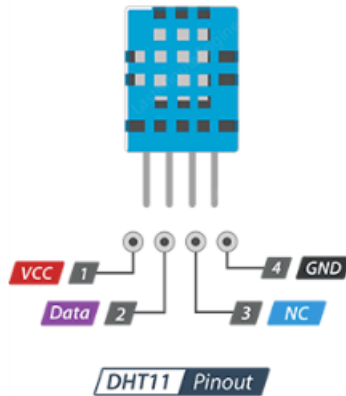
- Arduino Uno
- DHT11 Heat Sensor
- Jumper Cables
- Solderless Breadboard
- Strong Lightbulb that capable of producing heat
- A Box to assemble everything together
- Cut cable to power lightbulb from home outlet and connect to relay
- One Channel Relay that works with 5 Volt

Tools and Platforms

As a development environment, I will use Arduino IDE which is the platform we will be able to send and receive data. From this platform, we will be able to control our lightbulb and access every data from DHT11 from Serial Monitor. It also supports required libraries to run DHT11 sensor.

Component Examination

DHT-11 Heat Sensor

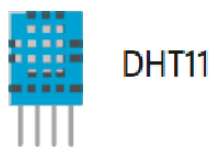


VCC pin supplies power for the sensor. Although supply voltage ranges from 3.3V to 5.5V, 5V supply is recommended. In case of 5V power supply, you can keep the sensor as long as 20 meters. However, with 3.3V supply voltage, cable length shall not be greater than 1 meter. Otherwise, the line voltage drop will lead to errors in measurement.

Data pin is used to communication between the sensor and the microcontroller.

NC Not connected

GND should be connected to the ground of Arduino.



Operating Voltage	3 to 5V
Max Operating Current	2.5mA max
Humidity Range	20-80% / 5%
Temperature Range	0-50°C / ± 2°C
Sampling Rate	1 Hz (reading every second)
Body size	15.5mm x 12mm x 5.5mm
Advantage	Ultra low cost

One Channel Relay



Control Pins:

IN pin is used to control the relay. It is an active low pin, meaning the relay will be activated when you pull the pin LOW and it will become inactive when you pull the pin HIGH.

GND is the ground connection.

VCC pin supplies power to the module.

Output Terminal:

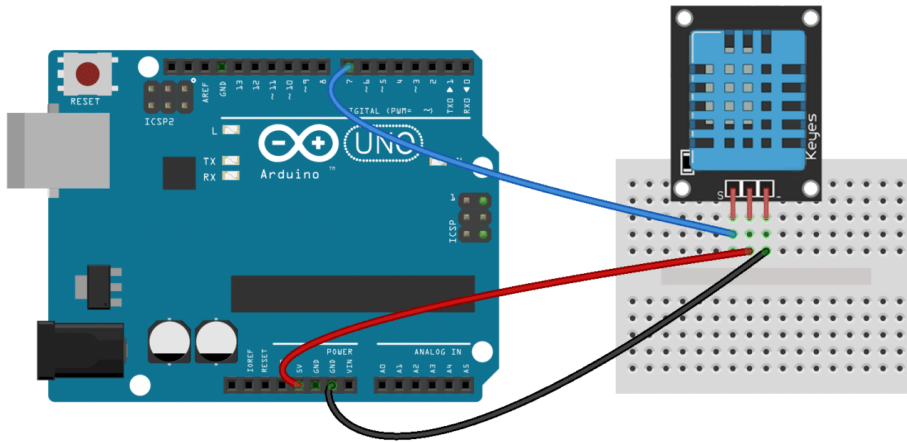
COM pin is connected to the signal you are planning to switch.

NC pin is connected to the COM pin by default, unless you send a signal from the Arduino to the relay module to break the connection.

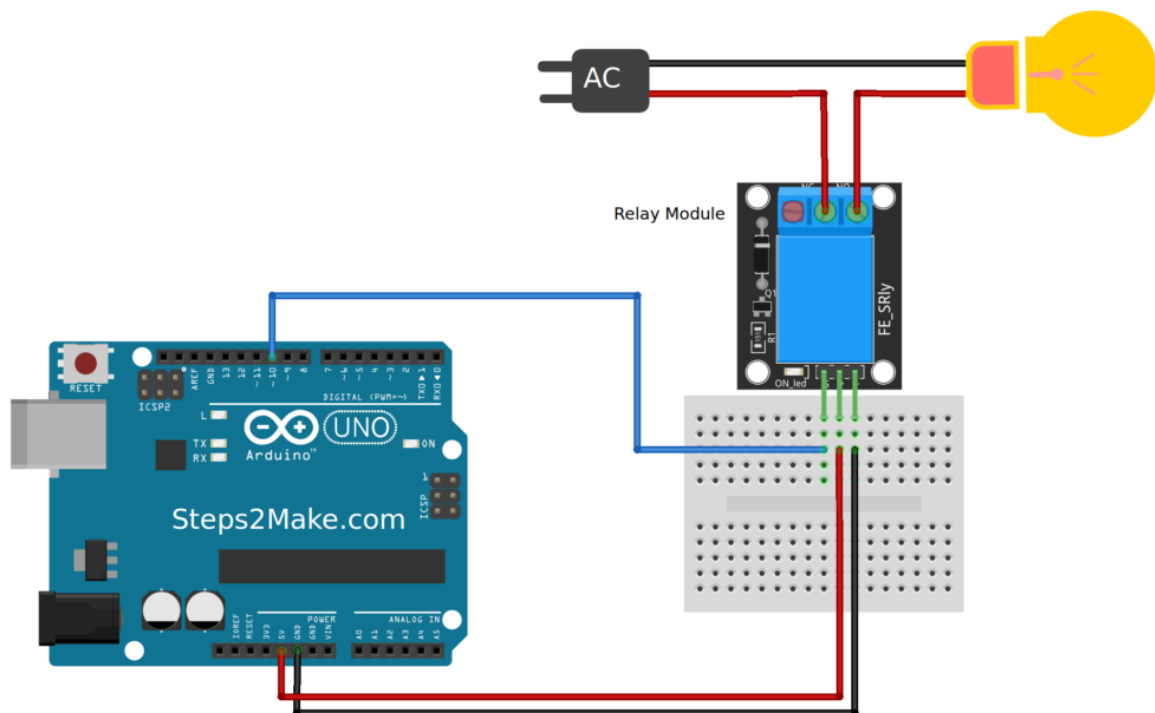
NO pin is open by default, unless you send a signal from the Arduino to the relay module to make the connection.

Circuit Connections and Planning

DHT-11 to Arduino Uno Connections:



Relay to Arduino Uno Connections:



Coding

In my code, I will need to:

- Install Required Libraries
- Receive data from DHT11
 - Define Connected Pin
 - Define DHT Type
 - Create Required Parameters and Read Data
- Control Lightbulb
 - If temperature is higher than expected -> Turn the Lightbulb Off by sending signal to Relay
 - If temperature is lower than expected -> Turn the Lightbulb On by sending signal to Relay
 - If temperature is in optimal range -> Do nothing

Code :

Defining Parameters

```
#include "DHT.h"

#define role 2
#define led0 8 // green
#define led1 7 // red
#define led2 9 // blue

#define DHTPIN 3
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(role, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop() {
  delay(5000);
  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);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
}
```

Controlling Lightbulb and Led's

```
// Compute heat index in Celsius (isFahrenheit = false)
float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("% Temperature: "));
Serial.print(t);

if(t<35.00){
  digitalWrite(led2,HIGH);
  digitalWrite(led0,LOW);
  digitalWrite(led1,LOW);

  digitalWrite(role,HIGH);
  Serial.print(" - Light is On \n");
}
if(t>=36.00){
  digitalWrite(led1,HIGH);
  digitalWrite(led2,LOW);
  digitalWrite(led0,LOW);
}
if (t>=35.00){
  digitalWrite(role,LOW);
  if(t<36.00){
    digitalWrite(led0,HIGH);
    digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);
  }
  Serial.print(" - Light is Off \n");
}
```

Photos of The Project



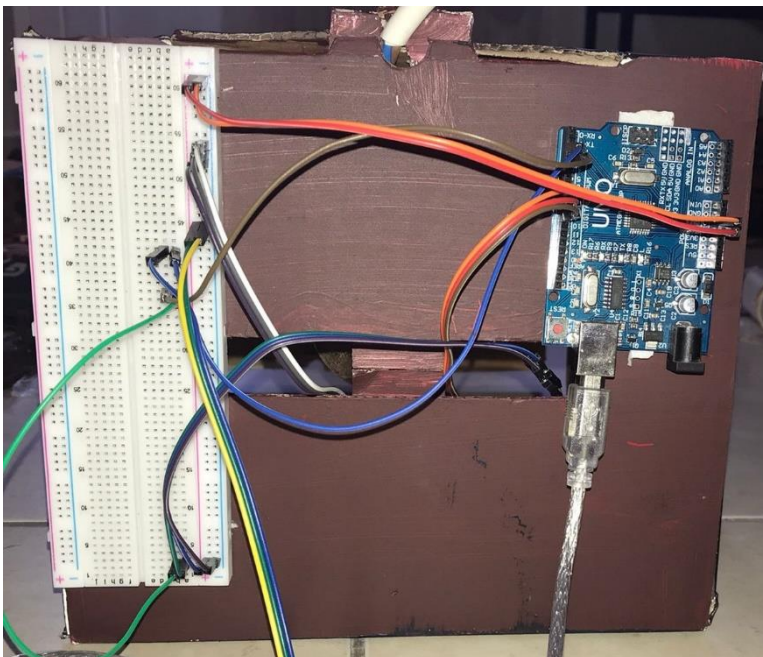
Box When it's Turned Off

Three LED's on top is for us to see the temperature inside the box.

Red -> Temperature is higher than 36C

Green -> Temperature is at 35C

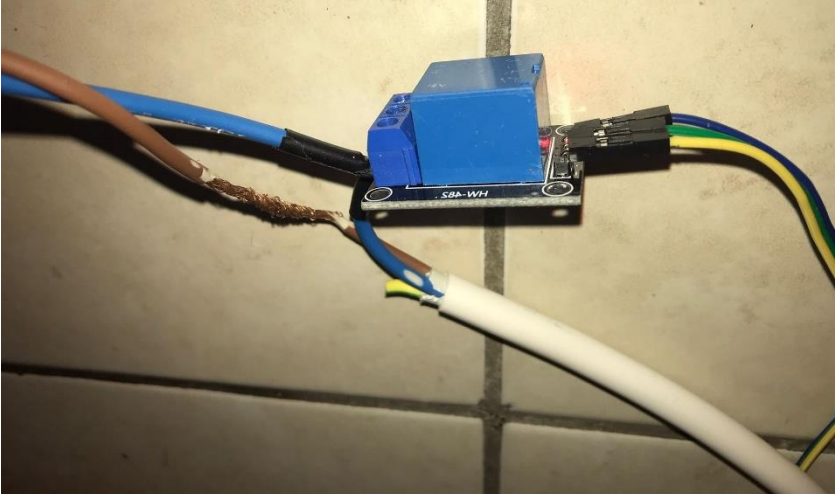
Blue -> Temperature is lower than 35C



Circuit Board Connections

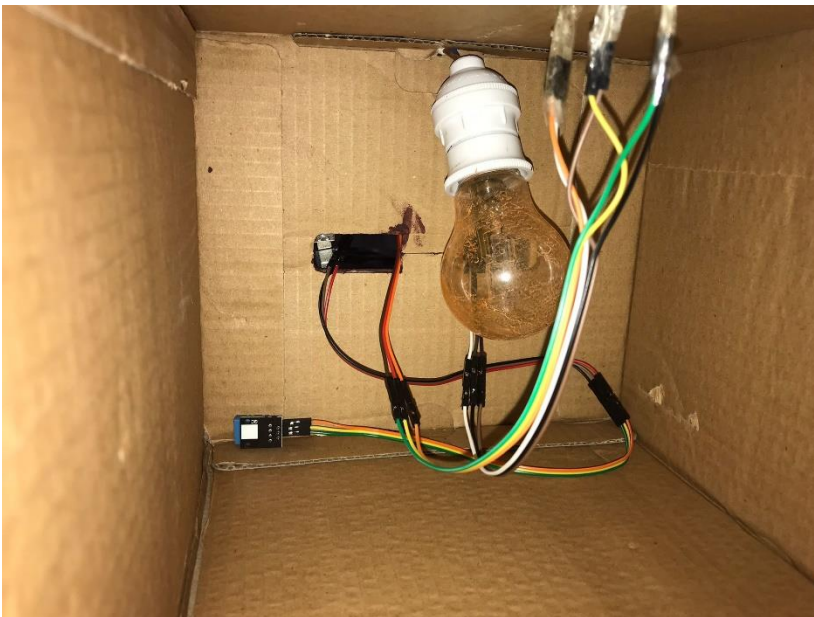
The Ground and 5Volt cables from Arduino are connected to the breadboards + and -. I power the whole circuit from there.

The box has small holes in the back for the cables to come in.



Relay Connections

I Cut a power cable to make connections between relay and bulb. The connections are the same as the Relay Diagram above

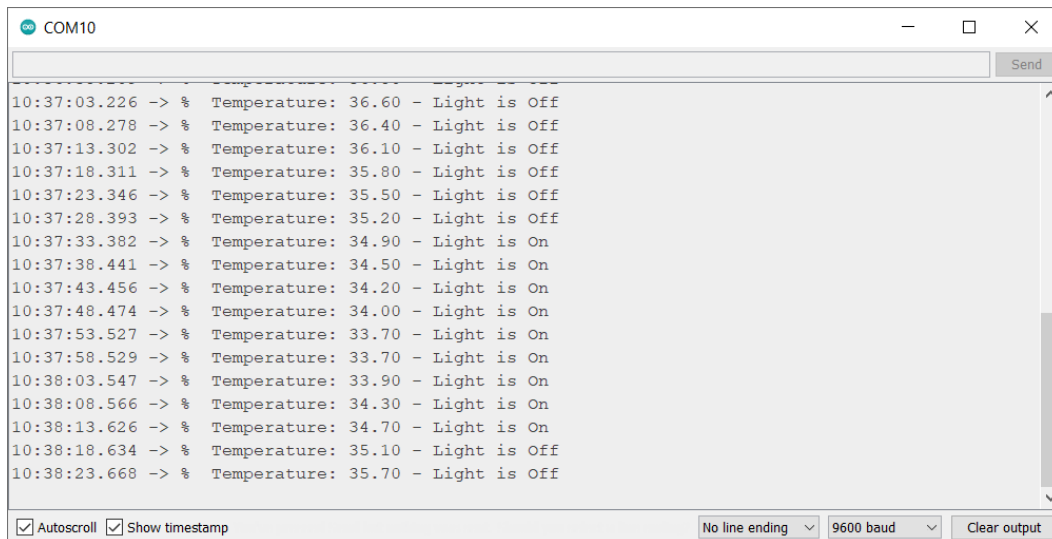


Inside of The Box

Heat Sensor can be seen on bottom left. Cables in the middle are to power and control LED's.

Running The Project

To run the project, all needs to be done is to power Arduino Uno and plug lightbulbs cable to any 220 Volt outlet. To power Arduino Uno we can use USB Cable or an external power supply. Once Arduino Uno has enough power and lightbulb's plug is connected to 220V, program will start receiving data and control the lightbulb according to temperature:



```
COM10
10:37:03.226 -> % Temperature: 36.60 - Light is Off
10:37:08.278 -> % Temperature: 36.40 - Light is Off
10:37:13.302 -> % Temperature: 36.10 - Light is Off
10:37:18.311 -> % Temperature: 35.80 - Light is Off
10:37:23.346 -> % Temperature: 35.50 - Light is Off
10:37:28.393 -> % Temperature: 35.20 - Light is Off
10:37:33.382 -> % Temperature: 34.90 - Light is On
10:37:38.441 -> % Temperature: 34.50 - Light is On
10:37:43.456 -> % Temperature: 34.20 - Light is On
10:37:48.474 -> % Temperature: 34.00 - Light is On
10:37:53.527 -> % Temperature: 33.70 - Light is On
10:37:58.529 -> % Temperature: 33.70 - Light is On
10:38:03.547 -> % Temperature: 33.90 - Light is On
10:38:08.566 -> % Temperature: 34.30 - Light is On
10:38:13.626 -> % Temperature: 34.70 - Light is On
10:38:18.634 -> % Temperature: 35.10 - Light is Off
10:38:23.668 -> % Temperature: 35.70 - Light is Off
```

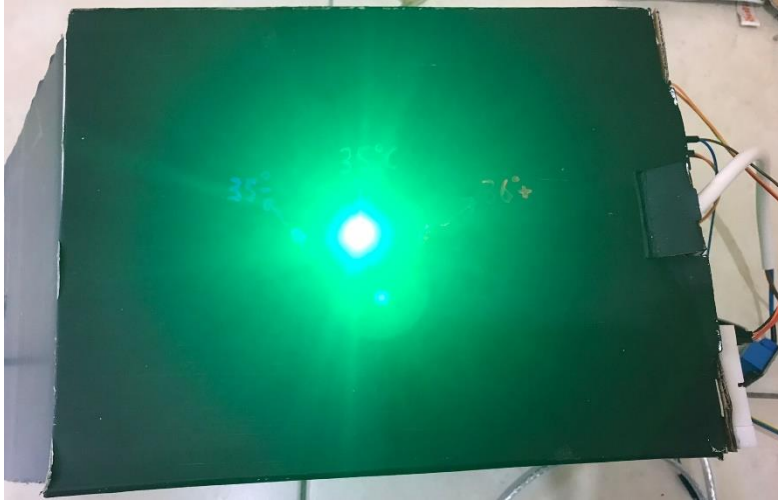
Autoscroll Show timestamp No line ending 9600 baud Clear output

The image above is the temperature data from the DHT11. For this project, I chose 35C as desired temperature. In the Serial Monitor, we can see that when temperature is above 35C, it says "Light is Off" because it has reached it's desired temperature. When the temperature is higher than 36, I programmed it to light **RED LED** on top of the box to alert that maximum temperature has been exceeded. In real life it looks like this:



In this state, the lightbulb is turned off and circuit is on hold until temperature drops below 35C

In Serial Monitor we can see that the lightbulb is also off when temperature inside the box is between 35C and 36C. This is the optimum temperature we want. In this state, the program will turn off the lightbulb and light the **GREEN LED** which indicates that the temperature is on it's optimum state. This looks like:



In this state, the lightbulb is turned off and circuit is in it's desired temperature. It is on hold mode until temperature drops below the desired level.

When it is activated, unless the room temperature is higher than 35C, it will start on increase heat mode. The program will turn on the lightbulb and wait until desired temperature has been reached inside the box. In this state program will light the **BLUE LED** which shows that the temperature is below 35C and it needs to increase the heat by keeping the lightbulb On. This looks like:



Here, the temperature inside the box is below desired level. The lightbulb is turned On by the program an it will stay that way until required temperature inside the box has been reached.