ThermoLightCooler

Theodor-Ioan Rolea, 323CA
Faculty of Automatic Control and Computer Science, Politehnica University of Bucharest
tedy.rolea@gmail.com

1 Introduction

This project focuses on creating a temperature monitoring system that has an auxiliary fan which can be turned on if the temperature reaches above a certain threshold. The thermistor sends data to the microcontroller that then lights up the RGB LED either RED (High temperature, above 26°C), GREEN (Normal temperature, between 25°C and 26°C) or BLUE (Low temperature, below 25°C). Then, if the temperature is high and we want to, we can turn on the auxiliary fan by pressing the VOL+ button on the remote. This will send the IR sensor the signal for the fan to start up. The fan will run until the temperature is below 26°C or we stop it using the VOL- button on the remote.

1.1 Simplified block diagram

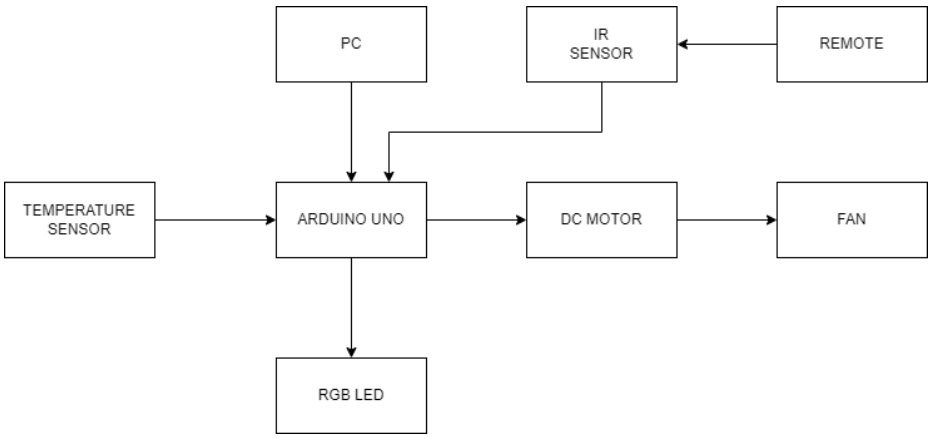

Fig. 1. Simplified block diagram

2 Implementation

2.1 Physical project

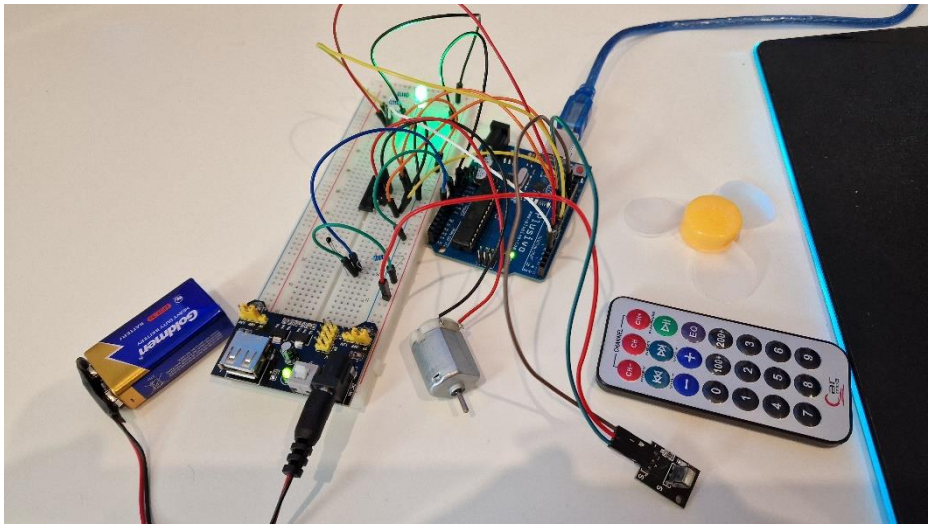

Fig. 2. Normal temperature

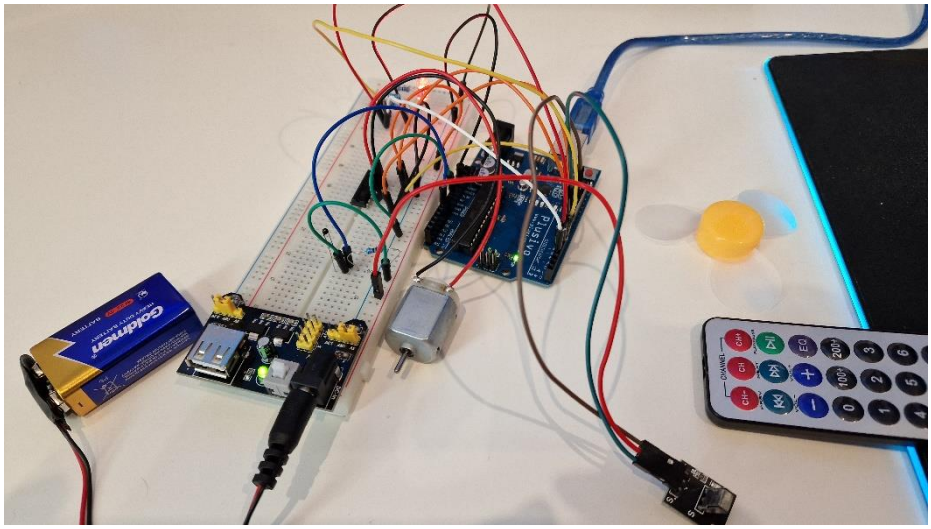

Fig. 3. High Temperature

2.2 Tinkercad implementation

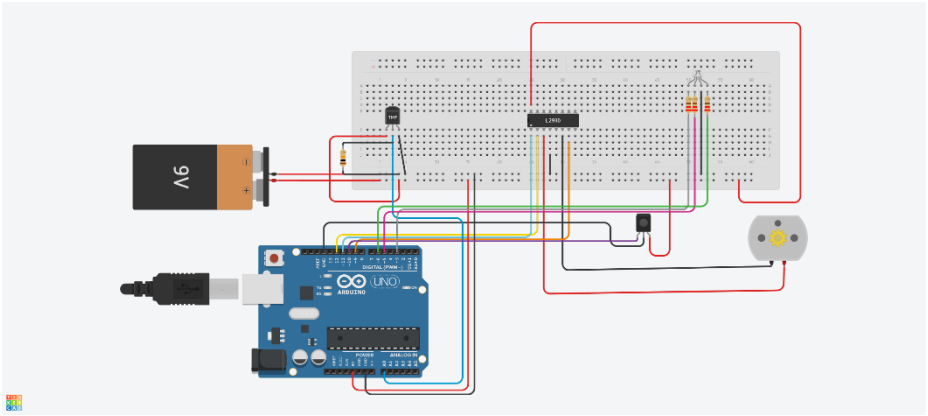

Fig. 4. Tinkercad implementation of the project

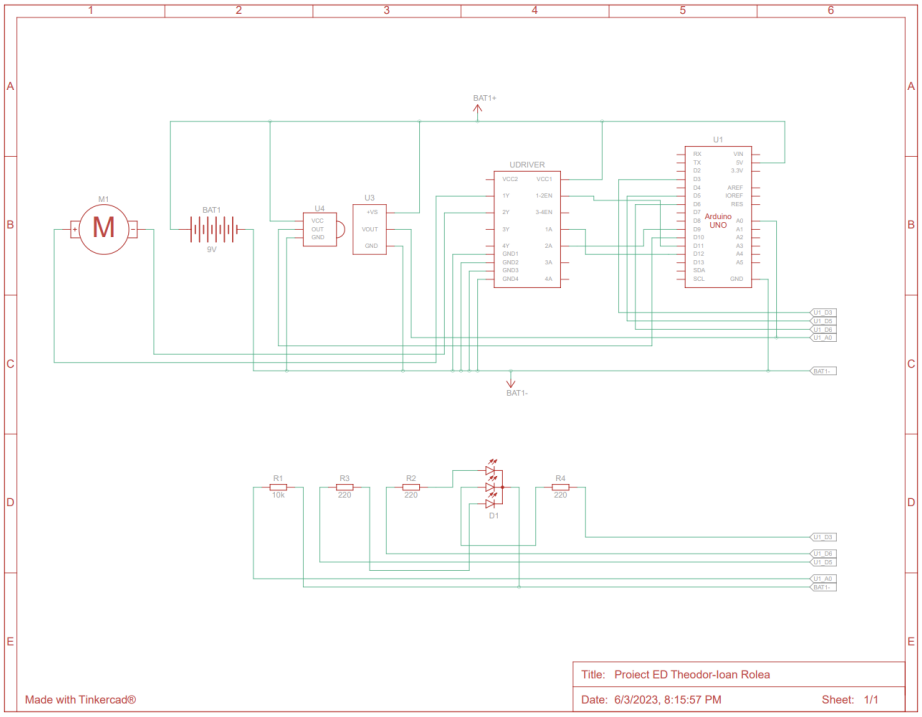

Fig. 5. Tinkercad implementation of the project

Disclaimer: I could not find the exact components I've used for the physical implementation. I have tried to improvise.

2.3 Used Components

- Arduino Uno (Plusivo Uno R3)
- 830 tie-points Breadboard
- Thermistor
- IR sensor
- Remote
- RGB Led
- L293D IC
- Fan blade and 3-6v motor
- Power Supply Module
- 9V battery
- 9V 1A adapter
- 10k & 220 Ohm Resistors
- M-M wires (Male to Male jumper wires)
- F-M wires (Female to Male DuPont wires)

2.4 How it functions

Video of how everything works: https://youtu.be/Go6VDzgRm9Y.

We start the Arduino by plugging it into our PC. The components use **5V** as the base voltage. The thermistor constantly sends data to the Arduino, lighting up the RGB LED different colors based on the ambient temperature: RED (High temperature, above 26°C), GREEN (Normal temperature, between 25°C and 26°C) or BLUE (Low temperature, below 25°C). If the temperature is above 26°C, we can turn on the auxiliary fan. The Arduino receives commands from the remote through the IR sensor. Once a command has been detected, we check whether the given command code is either VOL+ or VOL-, and we start or stop the fan based on the command. If the temperature drops below 26°C, the fan will stop automatically. However, it can be stopped anytime by pressing the VOL- button on the remote.

I will also leave the code for the project at the end of this PDF.

3 Bill of materials

This project was made using the modules from this kit : link.
I have tried to find the same or similar products and list them individually here:

Module	Quantity	Price (RON)_	Supplier	Datasheet
Arduino Uno	1	44.99	Optimus Digital	Datasheet
Breadboard	1	9.49	Optimus Digital	Datasheet
Thermistor	1	8.5	Mouser	Datasheet
IR sensor	1	8.99	Optimus Digital	Datasheet (Not the actual one I use)
RGB LED	1	0.99	Optimus Digital	Datasheet
L293D IC	1	40.5	Amazon	Datasheet
Motor	1	8.99	Optimus Digital	-
Power Supply	1	5.99	Optimus Digital	-
Battery 9V	1	5.99	Optimus Digital	-
9V 1A Adapter	1	7.49	Amazon	-
10k Resistor	1	0.10	Optimus Digital	-
220 Resistor	1	0.10	Optimus Digital	-
M-M wires	13	7.73	Optimus Digital	-
F-M wires	3	7.99	Optimus Digital	-
Total		157.84		

3.1 Code

```
#include "IRremote.h"

int receiver = 10; // Signal Pin of IR receiver to Arduino Digital Pin 11
int startFan = -1;

/*-----( Declare objects )-----*/
IRrecv irrecv(receiver);     // create instance of 'irrecv'
decode_results results;      // create instance of 'decode_results'

//define the pins used
#define BLUE 3
#define GREEN 5
#define RED 6
#define ENABLE 11
#define DIRA 9
#define DIRB 12

int i;
int tempPin = 0;
```

```
void setup()
{
//set the pins used by the LED as output
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);

digitalWrite(RED, HIGH);
digitalWrite(GREEN, LOW);
digitalWrite(BLUE, LOW);

pinMode(ENABLE,OUTPUT);
pinMode(DIRA,OUTPUT);
pinMode(DIRB,OUTPUT);
Serial.begin(9600);

irrecv.enableIRIn(); // Start the receiver
}

// define variables
int redValue;
int greenValue;
int blueValue;

// main loop
void loop()
{
  int tempReading = analogRead(tempPin);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
   tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK *
tempK )) * tempK );      //  Temp Kelvin
  float tempC = tempK - 273.15;            // Convert Kelvin to Celcius
  float tempF = (tempC * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
  Serial.println("Fan off. Current temp:");
  Serial.println(tempC);
#define delayTime 10 // fading time between colors

if (tempC <= 26 && tempC >= 25) {
  greenValue = 255;
  redValue = 0;
  blueValue = 0;
  analogWrite(RED, redValue);
  analogWrite(GREEN, greenValue);
  analogWrite(BLUE, blueValue);
  delay(delayTime);
```

```
  }

  if (tempC > 26) {
    greenValue = 0;
    redValue = 255;
    blueValue = 0;
    analogWrite(RED, redValue);
    analogWrite(GREEN, greenValue);
    analogWrite(BLUE, blueValue);
    delay(delayTime);
    if (irrecv.decode(&results))  // have we received an IR signal?
    {
      Serial.println("switch");
      switch (results.value)

      {

        case 0xFFA857:       // VOL+ button pressed
          startFan = 1;
          Serial.println("up");
          break;

        case 0xFFE01F:        // VOL- button pressed
          startFan = 0;
          Serial.println("down");
          break;

        default:
          Serial.println(results.value, HEX);
      }
      delay(1000);
      irrecv.resume();  // receive the next value
    }
    if (startFan == 1) {
      while (tempC > 26) {
        digitalWrite(ENABLE,HIGH); //enable on
        digitalWrite(DIRA,LOW); //one way
        digitalWrite(DIRB,HIGH);
        int tempReading = analogRead(tempPin);
        // This is OK
        double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
         tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK *
tempK )) * tempK );      //  Temp Kelvin
        float tempC = tempK - 273.15;           // Convert Kelvin to Celcius
        float tempF = (tempC * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
```

```
      Serial.println("Fan on. Current temp:");
      Serial.println(tempC);
      if (irrecv.decode(&results))  // have we received an IR signal?
      {
       Serial.println("switch");
       switch (results.value)

       {

        case 0xFFA857:       // VOL+ button pressed
          startFan = 1;
          Serial.println("up");
          break;

        case 0xFFE01F:       // VOL- button pressed
          startFan = 0;
          Serial.println("down");
          break;

        default:
          Serial.println(results.value, HEX);
       }
       delay(1000);
       irrecv.resume();  // receive the next value
      }
      if (startFan == 0) {
       break;
      }
      if (tempC < 26) {
       digitalWrite(ENABLE,LOW);
       startFan = 0;
       break;
      }
     }
    digitalWrite(ENABLE,LOW);
    startFan = 0;
   } else if (startFan == 0) {
    digitalWrite(ENABLE,LOW);
   }
}

if (tempC < 25) {
  greenValue = 0;
  redValue = 0;
  blueValue = 255;
```

```
  analogWrite(RED, redValue);
  analogWrite(GREEN, greenValue);
  analogWrite(BLUE, blueValue);
  delay(delayTime);
}
delay(delayTime);
}
```