Data Acquisitions and Instrumentation Project

Virtual

Automated Cooling System

Dinu Ion-Irinel
Group 325CA, CTI, UPB
ion_irinel.dinu@stud.acs.upb.ro

June 1, 2022

Contents

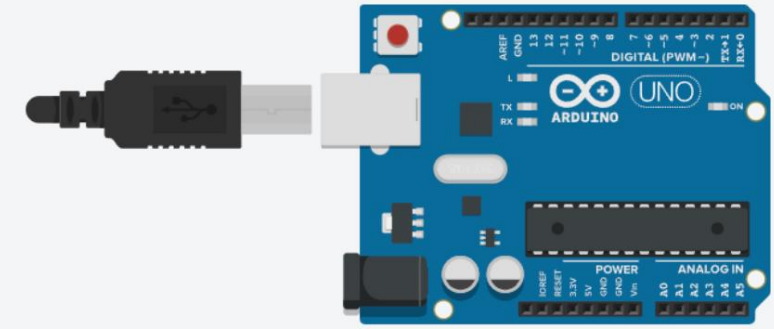
1. Introduction

1.1 General description

The project aims to create an Automated Cooler System, a cooler that uses two cooling motors driven by a temperature sensor and an Arduino board. Depending on the temperature taken from the sensor, the motors start at a certain speed. Also, three LEDs are used that illuminate depending on the temperature value. Both engines are surrounded by an LED strip, which simulates a play of colors.

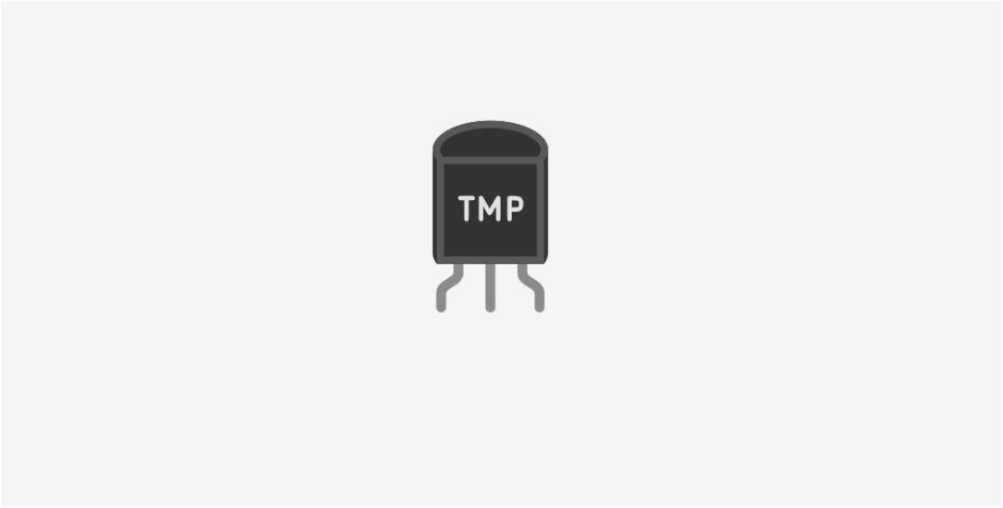
2 Components of the implemented circuit

2.1 Arduino One R3 board



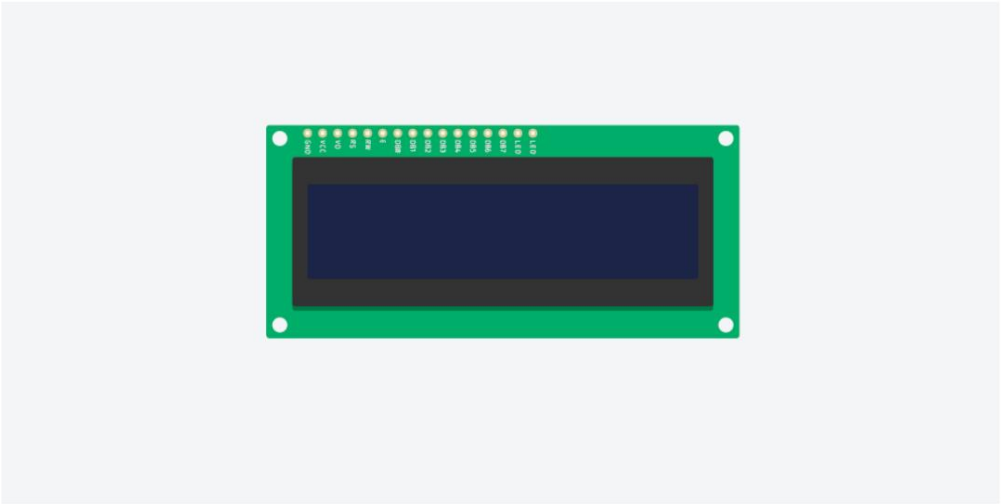
This component represents the base of the circuit, it is a component around which the whole circuit is built, powering and connecting all the other components to fulfill the desired functionality.

2.2 Temperature sensor



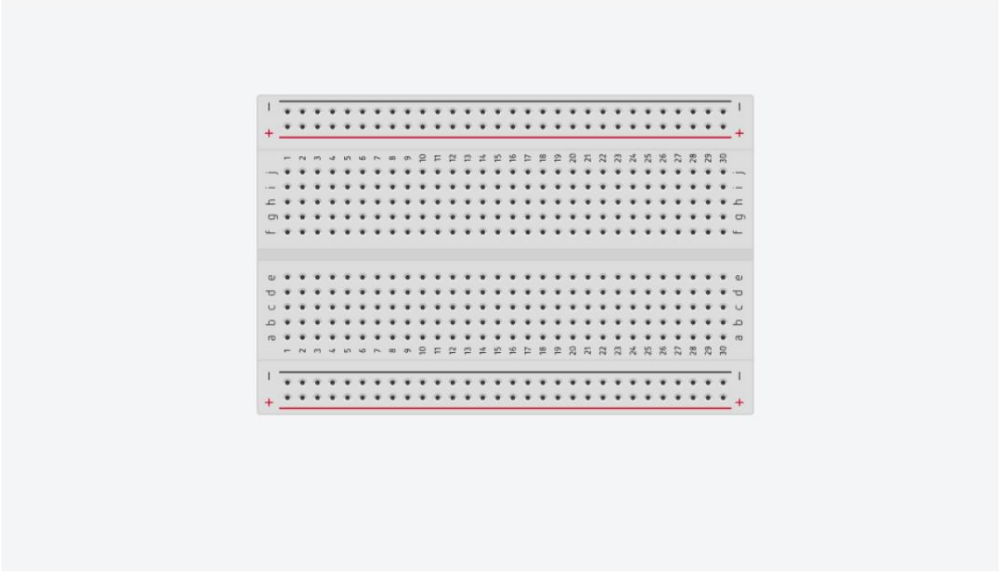
It represents a temperature sensor, TMP36 with the help of which the temperature is taken which is transmitted further to the other components.

2.3 16x2 LCD Display



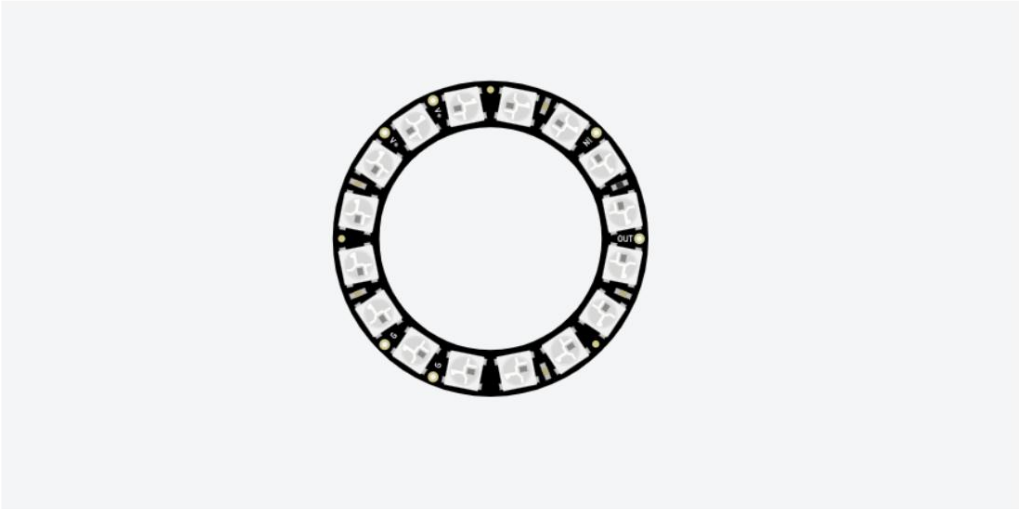
It represents a 16 x 2 LCD, an electronic display with which messages are displayed, such as temperature, engine speed, etc.

2.4 BreadBoard



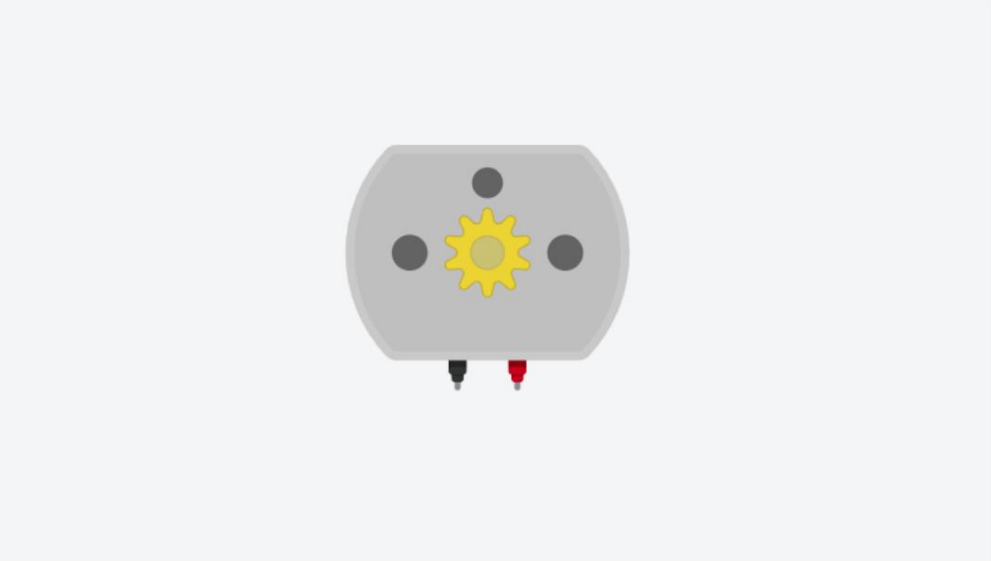
It is a breadboard, a device with which you can easily connect the wires between the circuit components.

2.5 Neon



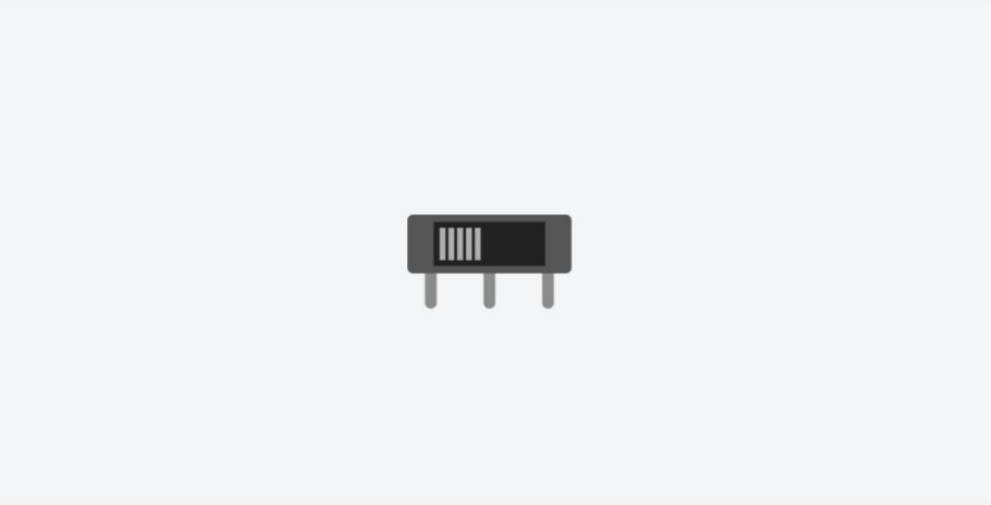
This component is a strip of lights with different color schemes, which will surround the cooler motors. The light strip uses RGB colors.

2.6 Engine



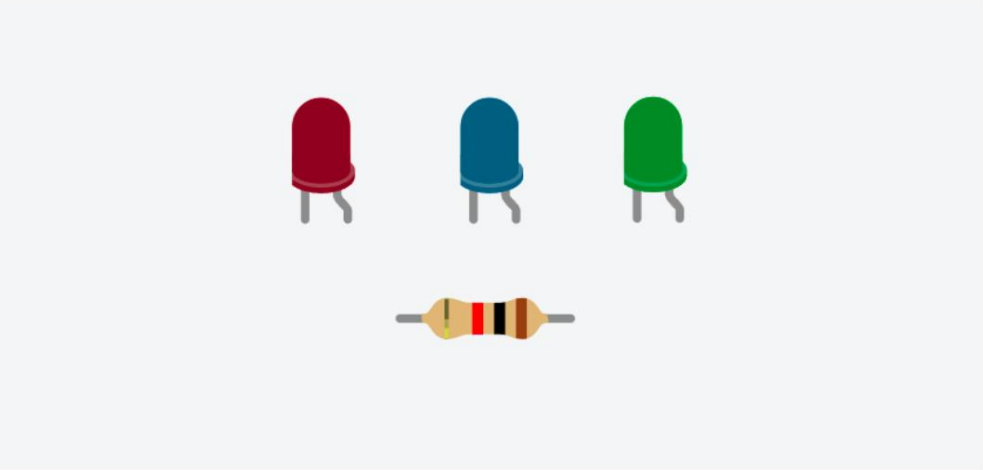
It is a DC Motor, a basic component in the circuit.
The motors start with a certain rotation based on the temperature taken from the sensor.

2.7 SlideSwitch



This is a slideswitch button with which you can turn on or off the power of the electronic display.

2.8 Bulbs



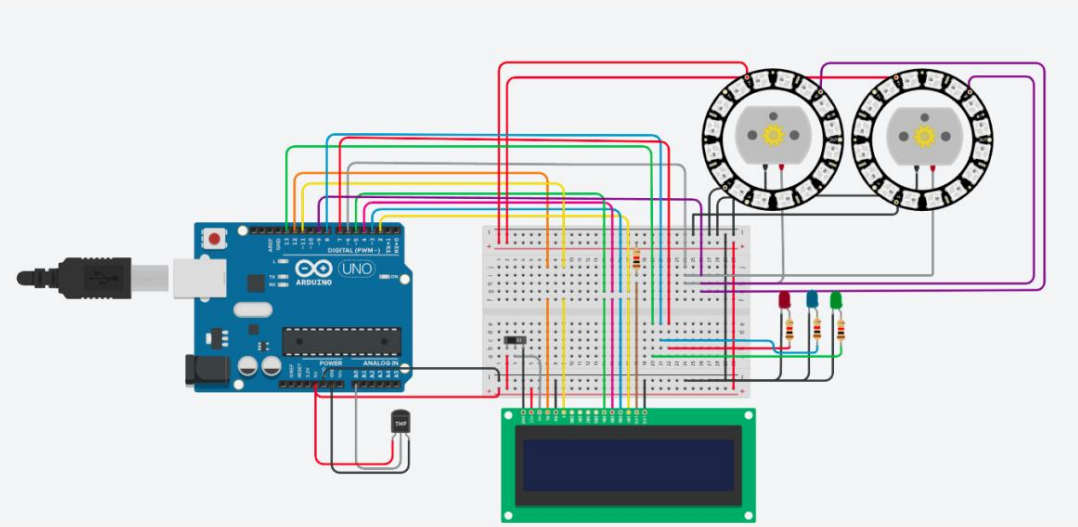
This represents a series of three bulbs, which light up one by one depending on the temperature value. The red light will indicate that the temperature taken from the sensor is very high. In order to limit the current / voltage for the three bulbs, each of them was inserted with a resistor.

3 Functionality description

Certain components inside a computer produce heat. The heat generated can decrease the performance of the computer or sometimes even affect it. State-of-the-art computers use more powerful components to increase performance. The higher the performance, the higher the heat generated. To lower the temperature inside the central unit, the cooler was created, an engine-driven cooling system.

The cooler implemented in the project uses two cooling engines.
The temperature value indicated by the TMP36 temperature sensor is taken over at a certain period of time. After reading, the temperature is converted to degrees Celsius, and a corresponding message is displayed using the electronic display. Also, depending on the temperature (20 - 40 - 60 ..) the motors start at a certain speed, which is displayed in the display, and the bulbs light up similarly.

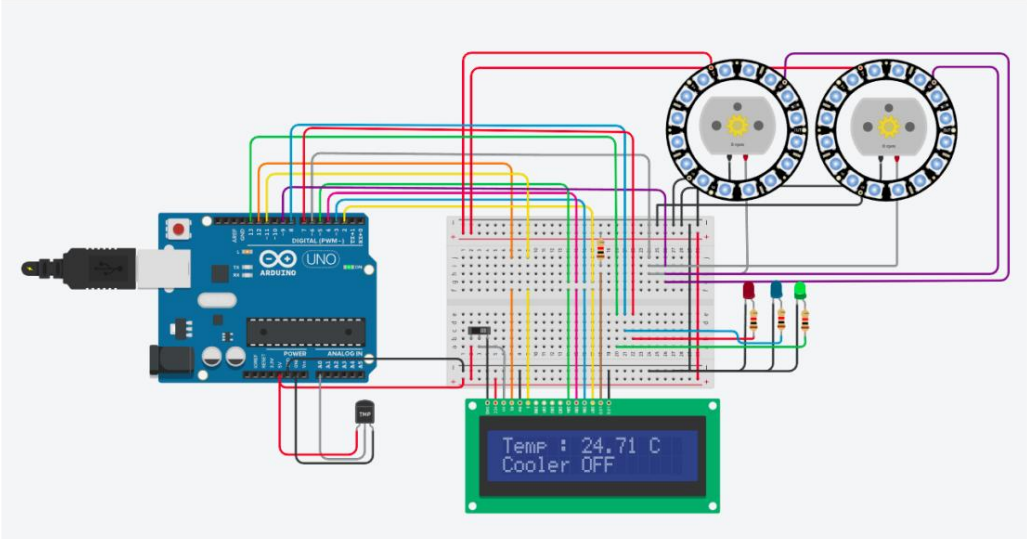
3.1 Circuit diagram



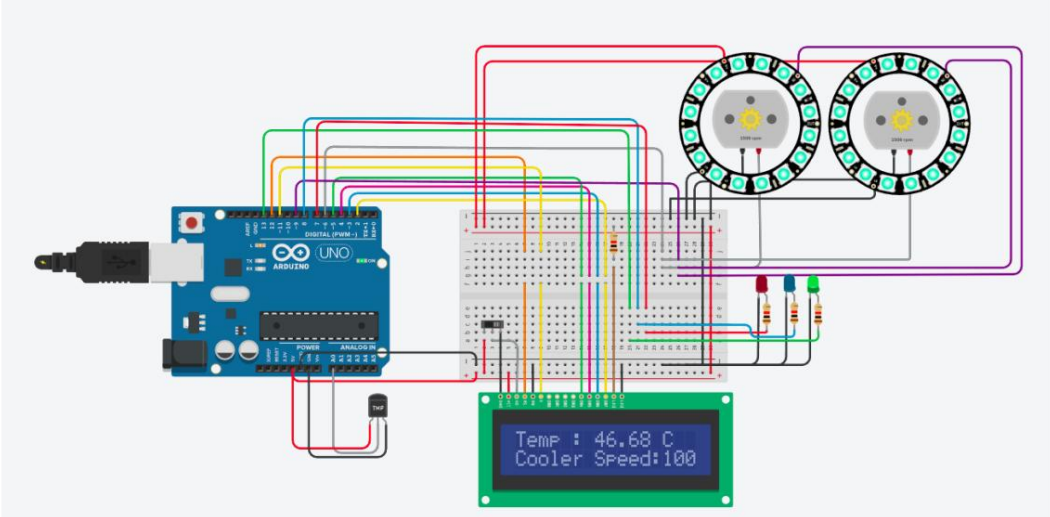
3.2 Implementation

- For the implementation of the circuit we used the Tinkercad platform
- To make the wire connections I used a breadboard and I try to use a color convention, red for connecting to In (5V) and black for connecting to Ground
- Components are powered from the Arduino board

3.3 Interpretation of data

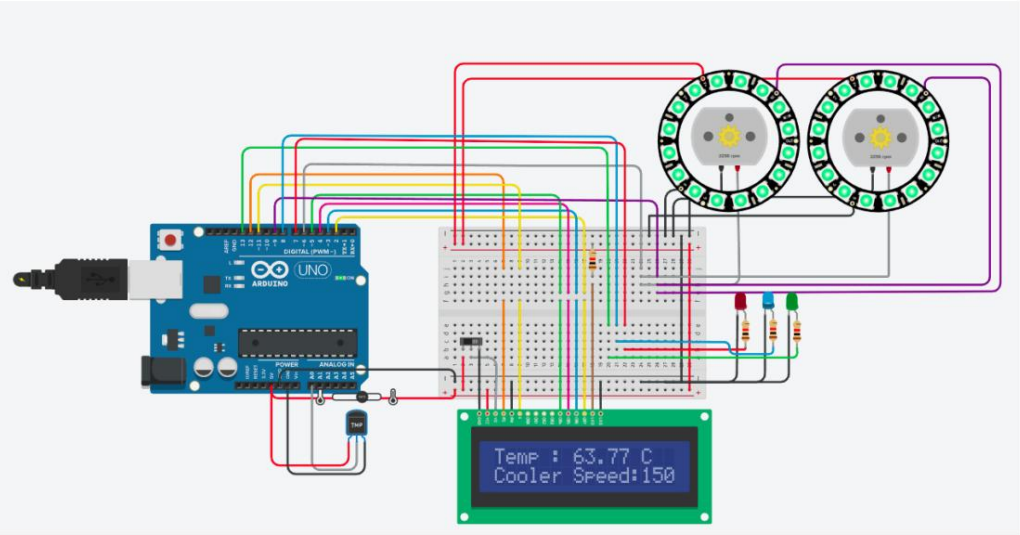


At the start of the circuit, the temperature indicated by the sensor is 24.71 degrees Celsius. For a temperature value lower than 40 degrees Celsius, the motors are in the OFF state, and the green bulb lights up, signaling a low temperature value. You can see the LED lighting of the motors made with the help of the neo ring.

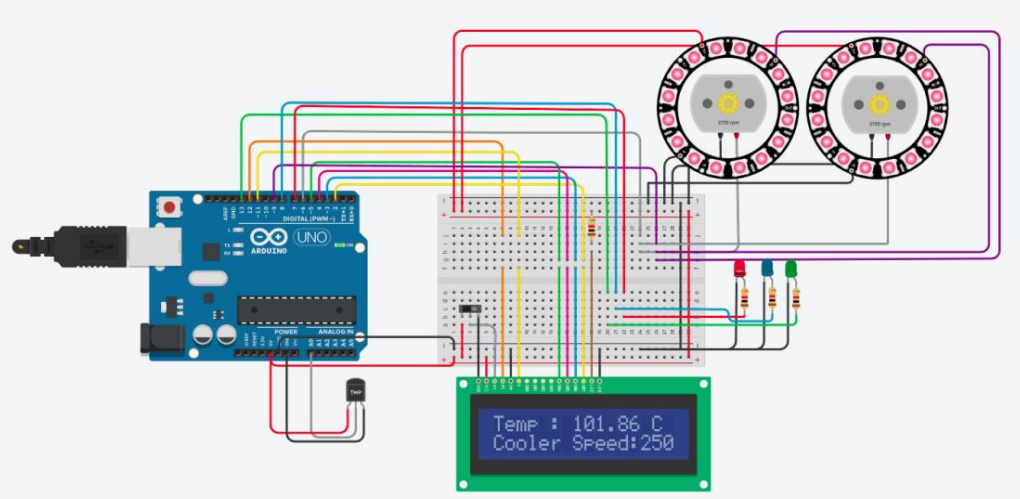


By changing the temperature value using the sensor at 46.7 degrees Celsius, the correct display of the value within the display is observed. The green light is on

continued, but the two engines start at one speed, with a speed of 100,
because the temperature value is between 40 and 60 degrees Celsius.

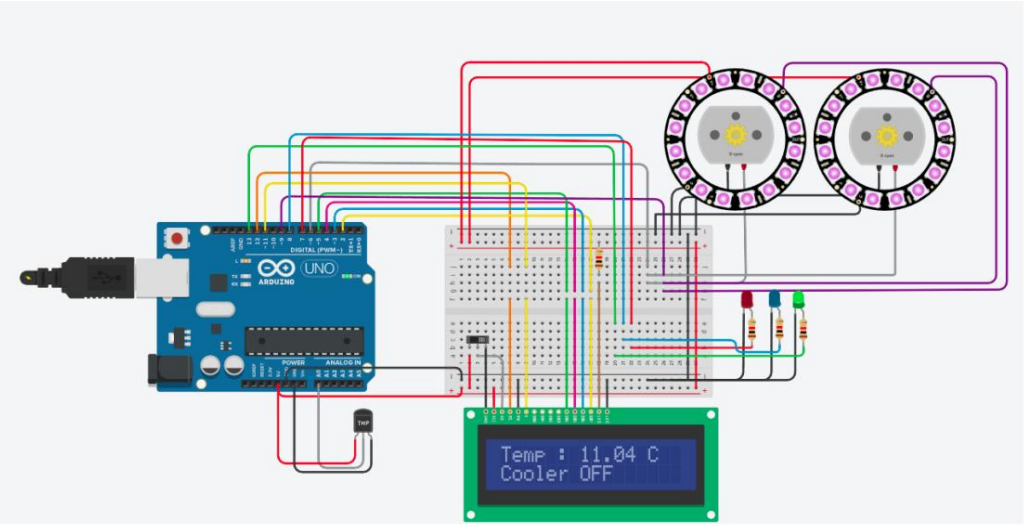


When the temperature indicated by the sensor has a value
between 60 and 80 degrees Celsius is observed the new functionality of the engine,
whose speed increases to 150, and the green light stops shining
to the detriment of the blue one which indicates an average temperature value.



At a temperature value between 80 and 125, a core message is displayed with the
help of the display and the motors start at maximum speed, with

a speed of 250. The blue light goes out, and the red light illuminates a very high temperature value.



If the temperature indicated by the sensor returns to a value of 12 degrees Celsius, the motors will be stopped, the corresponding message will be displayed using its display. The green light illuminates again indicating a low temperature value.

```
#in cl ude <A dafr ui t N e oPi x el. h>
#in cl ude <Li q ui d C ryst al. h>

// initialize the library with the numbers of the interface pi ns
Li q ui d C ryst al lcd (1 2 3  ,  1 1,  5 ,  4 ,  ,  2 ) ;

#in cl ude <A dafr ui t N e oPi x el. h>

#define PIN 9                    // Pine tree
#define N LEDS 32 // Number ofledintotalstrips

A dafr ui t N e oPi x el strip = A dafr ui t N e oPi x el (N LEDs, PIN  ,  NEO_GRB + NEO KHZ800);

floatv al ue;
float tempe ra tu re;
int motor = 6;
int speed = 0;
between dC ol or = 0;
intgree nC ol or = 0;
int bl u eC ol or = 0;
intfa nC o ol e rPi n = 6;

v oid se tup () {
        // cooler
        pinMode (f anC o ole rPin, OUTPUT);
        // led red
        pinMode (7, OUTPUT);
        // leg bl ue
        pinMode (8, OUTPUT);
        // ledgr ee n
        pinMode (1 3, OUTPUT);
        strip. be gi n ();
        randomSeed (analogRead (0));

        lcd. be gi n (1 6 2);
        pinMode (motor, OUTPUT);

        an al o gW ri te (motor, speed);
        lcd. clear ();
        speed = 0;
        an al o gW ri te (motor, speed);
        d el ay (1 0 0 0);
        lcd. set C ursor (30);  ,
        lcd. p ri nt (“”);  Temperature
        lcd. set C ursor (2 1); ,
        lcd. p ri nt (“C o ole r System”);
```

```
        d el ay (3 0 0 0);
        lcd. clear ();
}

v oid loop () {
        tempe ra tu re = analogRead (0) ÿ 0. 0 0 4 8 8 2 8 1 4;
        tempe ra tu re = (tempe ra tu re - 0. 5) ÿ 1 0 0. 0;
        lcd. set C ursor (0 0); ,
        lcd. p ri nt (“Temp:”);
        lcd. p ri nt (tempe ra tu re);
        lcd. p ri nt (“C”);
        d el ay (1 0 0 0);
        lcd. clear ();

        if (tempe ra tu re <4 0) {
                digita lW rite (f anC o ole rPin, LOW);
                digit lW rite (1 3 HIGH);
                digit lW rite (8, LOW);
                digit lW rite (7, LOW);
                lcd. set C ursor (0 1); ,
                lcd. p ri nt (“C o ole r OFF”);
                d el ay (1 0 0 0);
        }

        if (tempe ra tu re> 40 && tempe ra tu re <6 0) {
                digita lW rite (f anC o ole rPin HIGH);
                digit lW rite (1 3 HIGH);
                digit lW rite (8, LOW);
                digit lW rite (7, LOW);
                speed = 1 0 0;
                lcd. set C ursor (0 1); ,
                lcd. p ri nt (“C o ole r Speed: 1 0 0”);
                an al o gW ri te (motor, speed);
                d el ay (1 0 0 0);
        }

        if (tempe ra tu re> 60 && tempe ra tu re <8 0) {
                digit lW rite (7, LOW);
                digit lW rite (8 HIGH);
                digit lW rite (1 3, LOW);
                speed = 1 5 0;
                lcd. set C ursor (0 1); ,
                lcd. p ri nt (“C o ole r Speed: 1 5 0”);
                an al o gW ri te (motor, speed);
                d el ay (1 0 0 0);
        }
```

```
        if (tempe ra tu re> 80 && tempe ra tu re <1 2 5) {
                digit lW rite (7 HIGH);
                digit lW rite (8, LOW);
                digit lW rite (1 3, LOW);
                speed = 2 5 0;
                lcd. set C ursor (0 1); ,
                lcd. p ri nt (“C o ole r Speed: 2 5 0”);
                an al o gW ri te (motor, speed);
                d el ay (1 0 0 0);
        }

        color ();
}

// colorstrip
v oid color () {
        set C ol or ();
        for (inti = 0; i <1 6; i ++) {
                strip. set P ixel C olor (istrip. C, ol or (redC ol or, g reenC ol orstrip. show ();     ,   bl u eC ol or));

        }
}

// pi cks random v al uestosetfor RGB
v oid set C ol or () {
        re dC ol or = random (0 2 5,5);
        gree nC ol or = random (0 2 5,5);
        bl u eC ol or = random (0 2 5,5);
}
```

4 Bibliography

- https://www.tinkercad.com/
- https://www.youtube.com/watch?v=LrOM2GABK1g
- https://www.diagrameditor.com/
- https://www.overleaf.com/