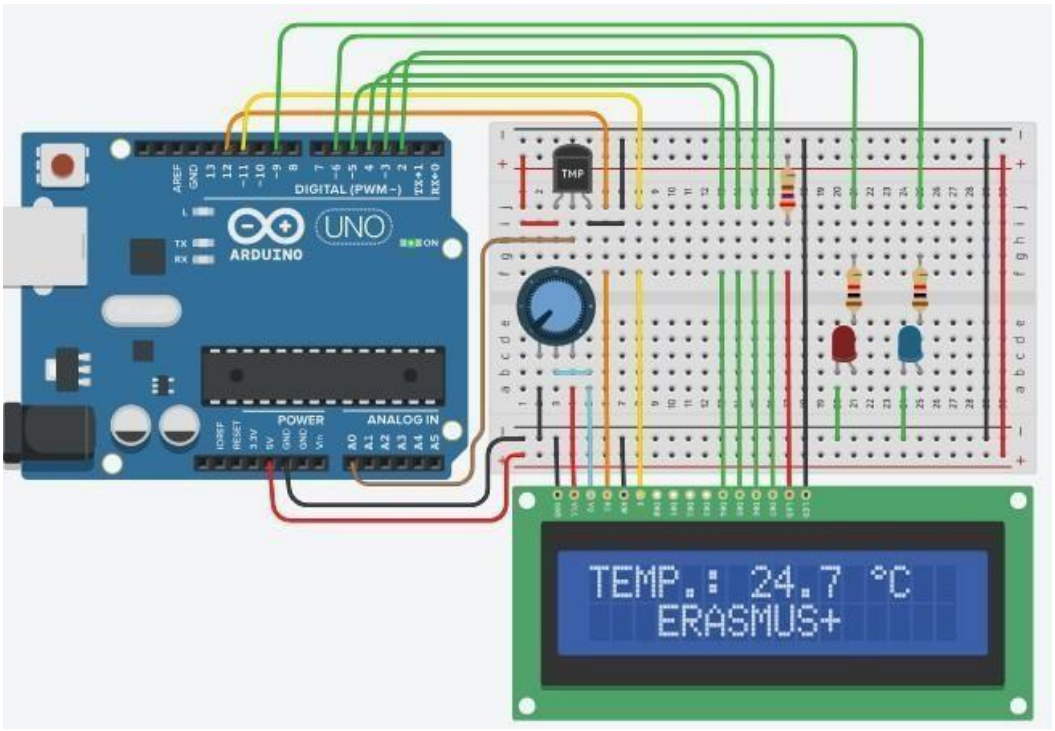
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

Heating/Cooling Project Tinkercad

Creation of a Temperature Control Simulator using Arduino with Tinkercad

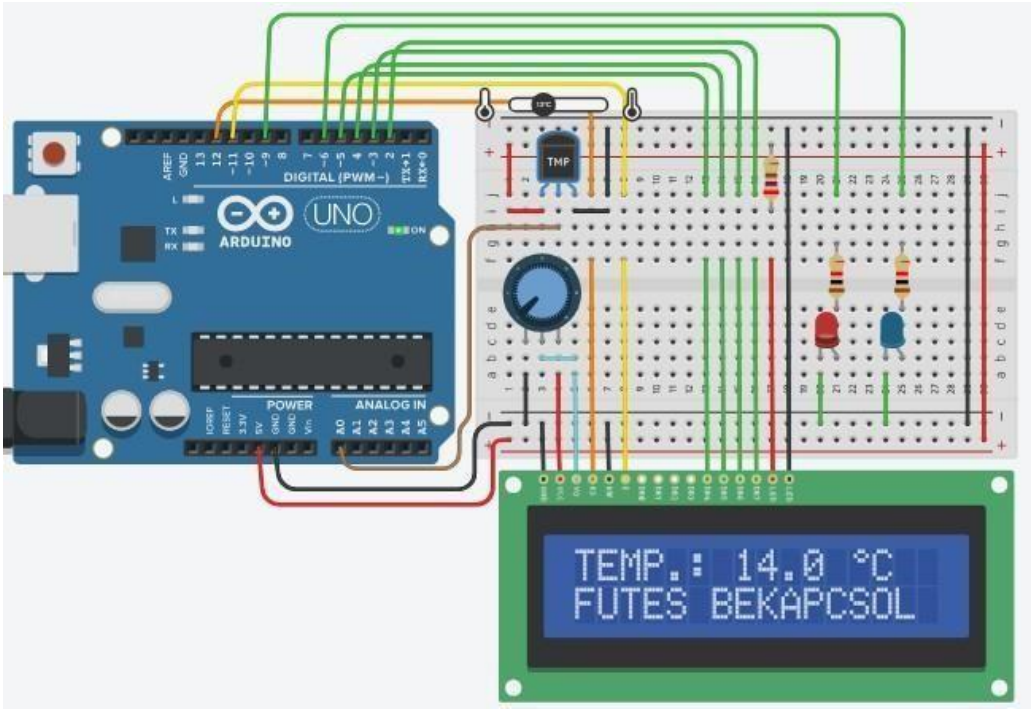
- **Project Title:** Discover the Green Life with Robots.
- **Project ID:** 2023-1-HU01-KA210-VET-000156243.

I. Circuit Implementation and Components



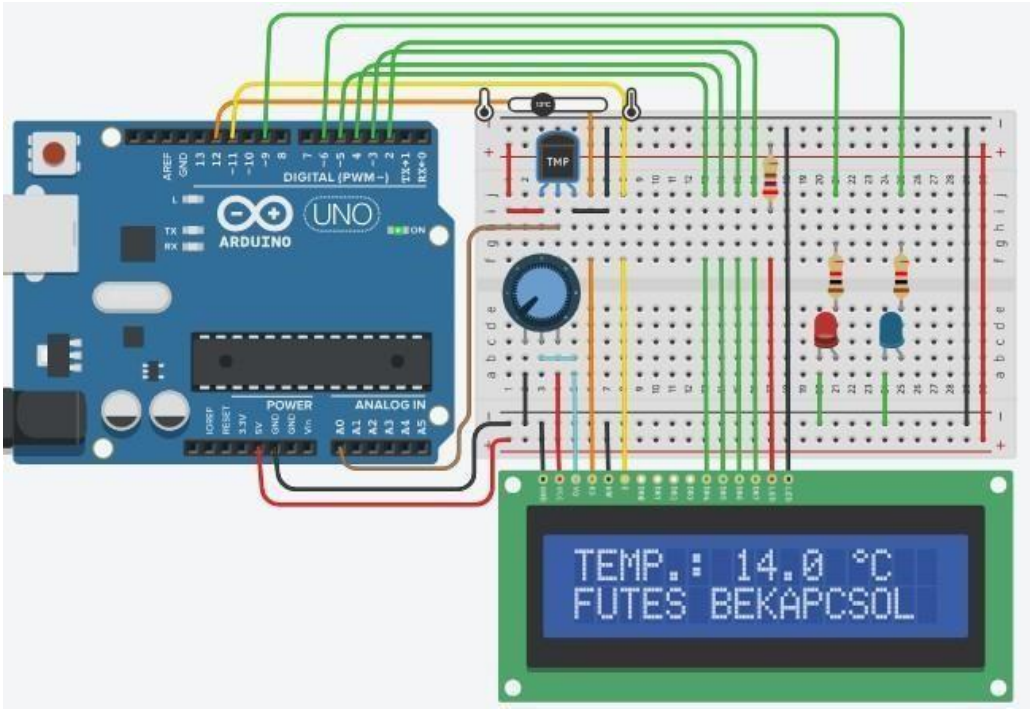
This circuit implements a temperature control circuit using **Arduino UNO**. The circuit was wired and programmed using the **Tinkercad simulator**.

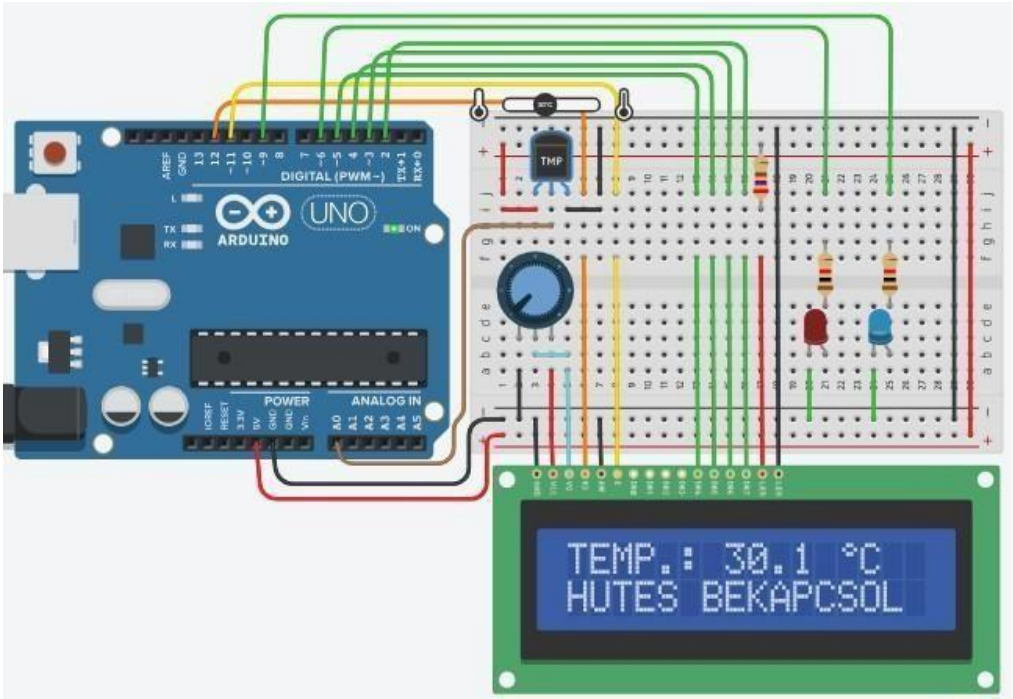
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.



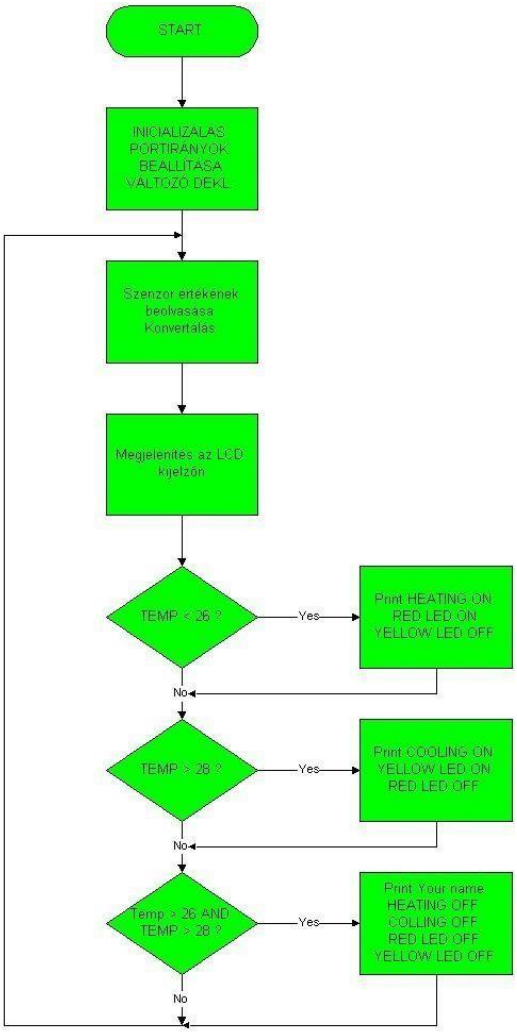
- **Indicators:** Cooling is indicated by a **blue LED**. Heating is indicated by a **red LED**.
- **Sensor:** Temperature sensing is performed by the **TMP36 sensor**.
- **Display:** The readable appearance of the display can be adjusted with a **potentiometer**.
- **Control Logic:**
    - Heating is switched ON when the temperature falls **below 18 degrees Celsius**.
    - Cooling is switched ON **above 28 degrees Celsius**.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.





Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.



II. Source Code (Arduino C++)

The source code was written based on the flowchart for the Erasmus+ project.

- **Author:** Kiss Antal István.
- **Project Description:** Heating-Cooling Project.
- **Libraries:** Includes <LiquidCrystal.h>.

```
#include <LiquidCrystal.h>

// LCD pins: RS(12), Enable(11), D4(5), D5(4), D6(3), D7(2)

LiquidCrystal lcd(12,11,5,4,3,2);

// Pin Definitions

int TMP36 = A0; // Temperature sensor connected to Analog pin A0
```

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

```
int hut = 9;    // 'Cooling' output connected to Digital pin 9
int fut = 6;    // 'Heating' output connected to Digital pin 6


// Custom character array definition for the degree symbol (°)
byte fok[]=
{
   B00110,
   B01001,
   B00110,
   B00000,
   B00000,
   B00000,
   B00000,
   B00000,
};


void setup()
{
   // Set 'Cooling' and 'Heating' pins as output
   pinMode(hut, OUTPUT);
   pinMode(fut, OUTPUT);


   // Initialize the LCD (16 columns, 2 rows)
   lcd.begin(16, 2);
   lcd.print("TEMP.:");
```

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

```
  // Create custom character for the degree symbol at index 0
  lcd.createChar(0,fok);
}

void loop()
{
  // Read Analog input A0
  int reading = analogRead(TMP36);
  // Convert reading to voltage
  float volt = reading * (5.0 / 1024.0);
  // Convert voltage to Celsius temperature
  float Temp = (volt-0.5) * 100;

  // Display Temperature value
  lcd.setCursor(7,0);
  lcd.print(Temp,1);       // Write Temp variable value
  lcd.print(" ");            // Space
  lcd.write(byte(0));      // Write the custom degree symbol
  lcd.print("C");
  lcd.setCursor(0,1);      // Set cursor to the 1st character of the 2nd row

  // Control Logic
  if (Temp < 18){
    lcd.print ("HEATING ON     "); // Translation: FUTES BEKAPCSOL
    digitalWrite(fut, HIGH); // Turn Heating LED/Actuator ON
  }
```

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

```
  else if (Temp > 28)
  {
    lcd.print ("COOLING ON     "); // Translation: HUTES BEKAPCSOL
    digitalWrite(hut, HIGH); // Turn Cooling LED/Actuator ON
  }

  // Print project label (This print statement may overwrite the previous status messages)
  lcd.print ("   ERASMUS+    ");

  // Switch Off Actuators (Note: these lines turn outputs LOW immediately after control check)
  digitalWrite(fut, LOW);
  digitalWrite(hut, LOW);

}
```