Thermostat

Name: Vintilescu Stefania-Madalina
FILS

Introduction

This project aims to provide a minimal implementation of a thermostat.

- The initial idea was to create a simple and affordable thermostat using commonly available components and an Arduino board.
- The purpose of this project is to maintain a consistent temperature in a room by turning on the fan or heating element based on the difference between the current temperature and the desired temperature set by the potentiometer.
- Using a thermostat can lead to energy savings and more consistent temperature control. Additionally, this type of project can be a great learning experience for beginners who are interested in learning about electronics and programming with an Arduino.

==== UPDATE ====

- The potentiometer is now utilized to adjust the brightness of the LCD, and the desired temperature value is changed using push buttons. A green led will also light up when the thermostat is in **cooling mode**. Otherwise, the red led will light up while in **heating mode**. This feature has been implemented so that the user can observe the current state of the thermostat.

General description

- We use a BMP280 temperature sensor to constantly monitor the current temperature of the room. The desired temperature is set using a potentiometer, and the thermostat compares this value to the current temperature.
- Depending on the difference between the current and desired temperature, the thermostat will activate either a heating element or a fan. These components are connected to a 2-channel relay

module, which in turn is connected to the appropriate pins on the Arduino board.
- The current temperature and the desired temperature are displayed on the I2C LCD, helping us decide whether to increase or decrease it.

=====UPDATED SCHEME=====

Hardware Design

The components used in this project are:

- 1x 1K Potentiometer
- 1x Breadboard +2x Mini BreadBoard
- 1x 16×2 I2C LCD
- 1x BMP280 temperature and pressure sensor
- 1x 12V charger(compatible with Arduino UNO R3 - will be used for the alimentation of the fan and the heating element)
- 1x Arduino UNO R3
- jumper wires
- 1x 12V 60x60x15mm fan
- 1x 12V heating element for 3D printers
- 1x 2-channel module relay
- 1x green led, 1x red led
- 4x resistors, different resistances

I will provide a legend for the wires as follows:

- red - VCC
- black - GND
- yellow - Vin (from the charger + NO- normally open- for both relays)
- pink - VCC for the fan (C2 of the 2nd relay)
- purple - VCC for the heating element (C1 of the 1st relay)
- gray - for the potentiometer (setting the luminosity of the LCD)
- brown - SCL (analog pin A5, connected for the I2C LCD and for the BMP280 sensor)
- white - SDA (analog pin A4, used for the same components as SCL)
- blue - cooling system (digital pins 8 - for the fan and 6 - for the blue push button which decreases the temperature)
- orange - heating system (digital pins 7 - for the heating element and 5 - red push button which increases the temperature)

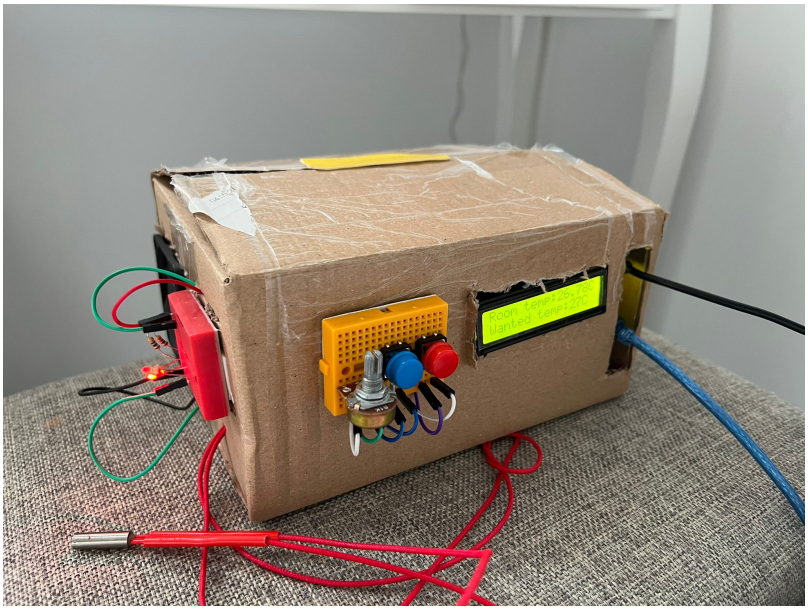
I didn't find the same-looking heating element in my design application, but I used an element with similar functionality(red heating pad).

Software Design

Description of my code (firmware):

- I used the Arduino app to develop my code
- This code utilizes the following libraries:
  1. Wire.h: This library provides functions for I2C communication, which is used to communicate with the BME280 sensor and for the I2C LCD.
  2. LiquidCrystal_I2C.h: This library allows easy control of I2C-based LCD displays. It is used to initialize and interact with the LCD display.
  3. SparkFunBME280.h: This library provides functions for interacting with the BME280 sensor, which is used to measure the room temperature.
- The code starts by including the necessary libraries and defining the pin connections for the fan, heating element, buttons, and LCD display. It also initializes the target temperature variable.
- In the setup() function, the LCD display is initialized and the I2C communication is started. The BME280 sensor's I2C address is set, and if the sensor fails to connect, an error message is printed.
- The loop() function is where the main functionality of the code resides. It starts by displaying the current room temperature from the BME280 sensor on the LCD display. It then displays the target temperature on the second line of the display.
- The target temperature can be adjusted using the blue and red buttons. If the blue button is pressed, the target temperature is decreased by 1 degree, and if the red button is pressed, the target temperature is increased by 1 degree. There is a delay after each button press for debounce.
- After adjusting the target temperature, the code compares the target temperature with the current room temperature. If the target temperature is higher, it turns on the heating element and turns off the fan. If the target temperature is lower or equal, it turns off the heating element and turns on the fan.
- There is a delay of 1 second before the loop repeats.

Results



Here is a demonstration video with the results obtained after this project.
https://youtu.be/h-x47DTjOQE

Conclusions

Building and working on this temperature control project with Arduino was an exciting and fulfilling experience. It allowed me to learn and apply electronics and programming skills while creating a practical solution. Overcoming challenges and completing the project brought a sense of accomplishment and inspiration for further exploration and future projects. The benefits include creating a comfortable environment and showcasing the power of open-source platforms like Arduino.

Download

Here are all my files regarding this project:
vintilescu_stefania-madalina_fils_1221a_project_materials.zip

Journal

- *27.04 → choosing the theme project*
- *02.05 → ordering the necessary components*
- *05.05 → making the first functional prototype of the software + hardware project*
- *07.05 → ordering other several components for new functionalities*
- *10.05 → making the second functional prototype of the software + hardware project (including new*

functionalities)
- *11.05 → solding the pins for the BMP280 sensor at the lab*
- *21.05 → finishing the hardware design in Fritzing app*
- *28.05 → adding the code information and making the last adjustments to the page*

Bibliography/ Resources

Hardware+Software Resources

- https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/ → how to connect and I2C LCD
- https://www.jamestharpe.com/arduino-thermostat/#:~:text=This%20Arduino%20Uno%20based%20thermostat,the%20temperature%20is%20too%20high. →inspirarion/starting idea
- https://www.youtube.com/watch?v=xA-vExF6ChI&t=283s → BMP280 sensor pins + inspiration code
- https://forum.fritzing.org/ → from where I imported some components for my hardware design(for Fritzing app)

Export to PDF