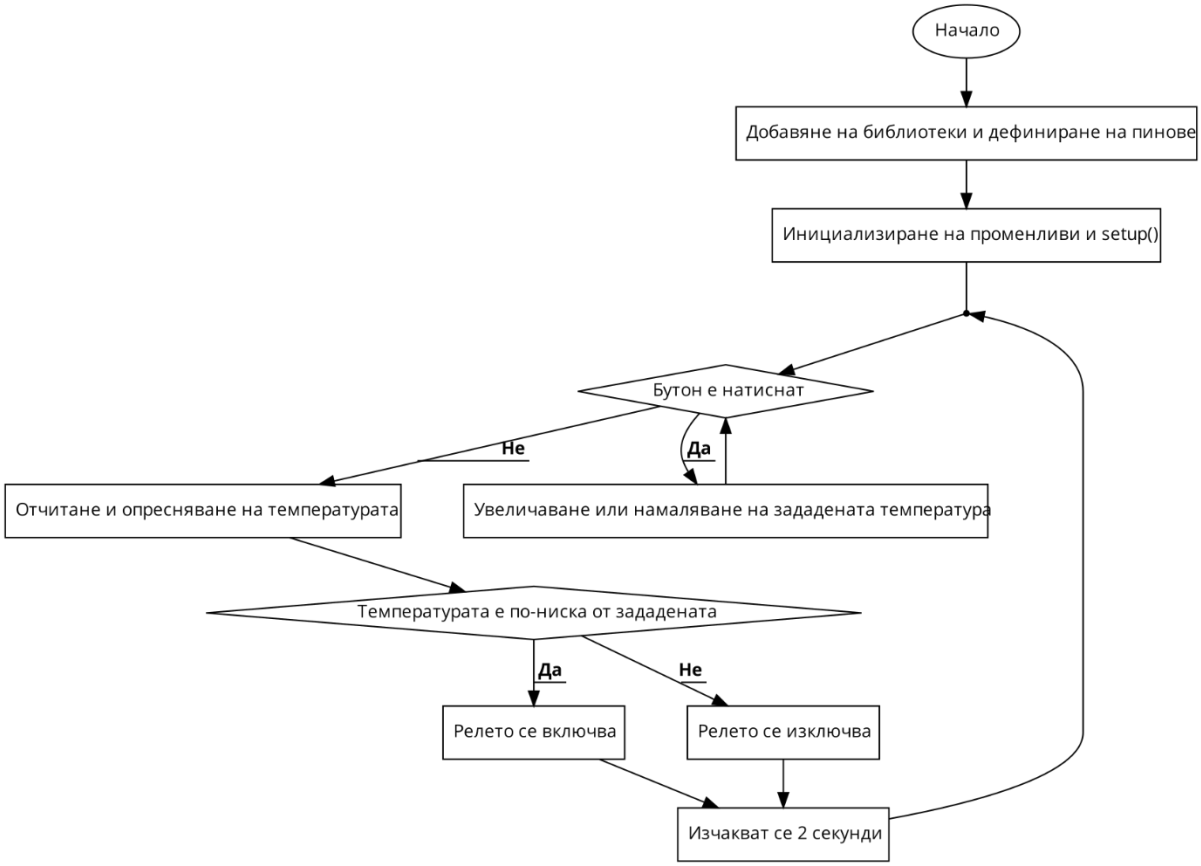
Курсов проект – Вградени системи
Термостат

Боян Белчев, 2022

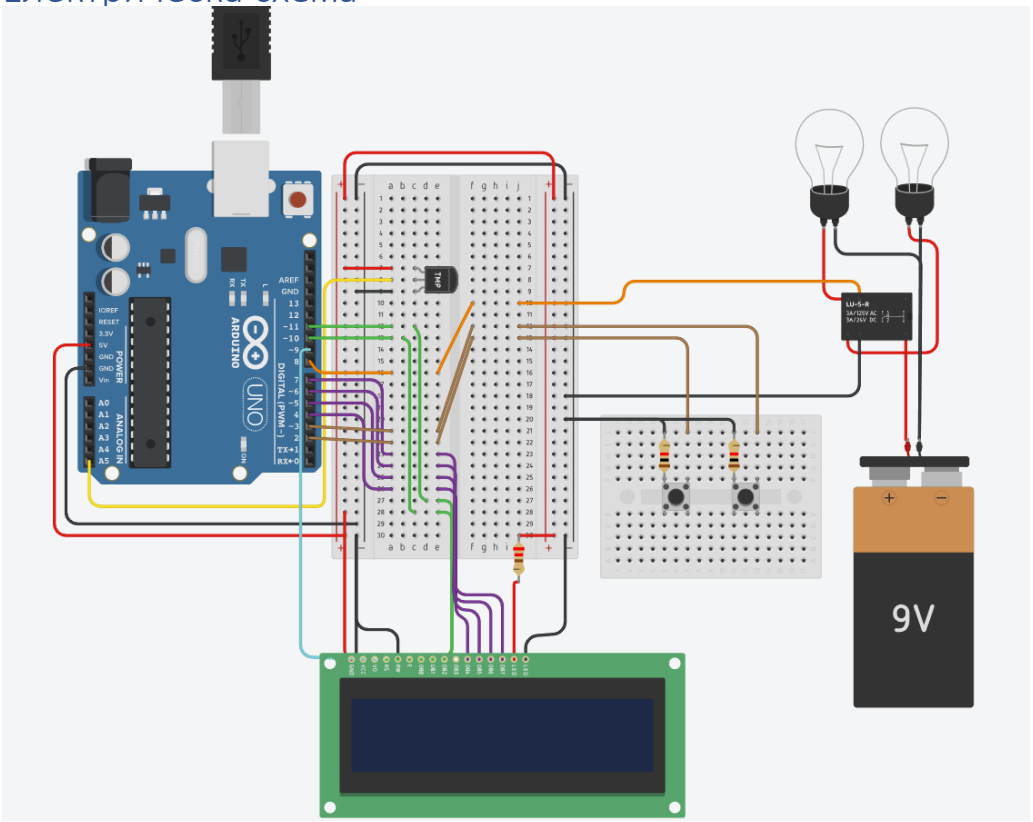
Описание

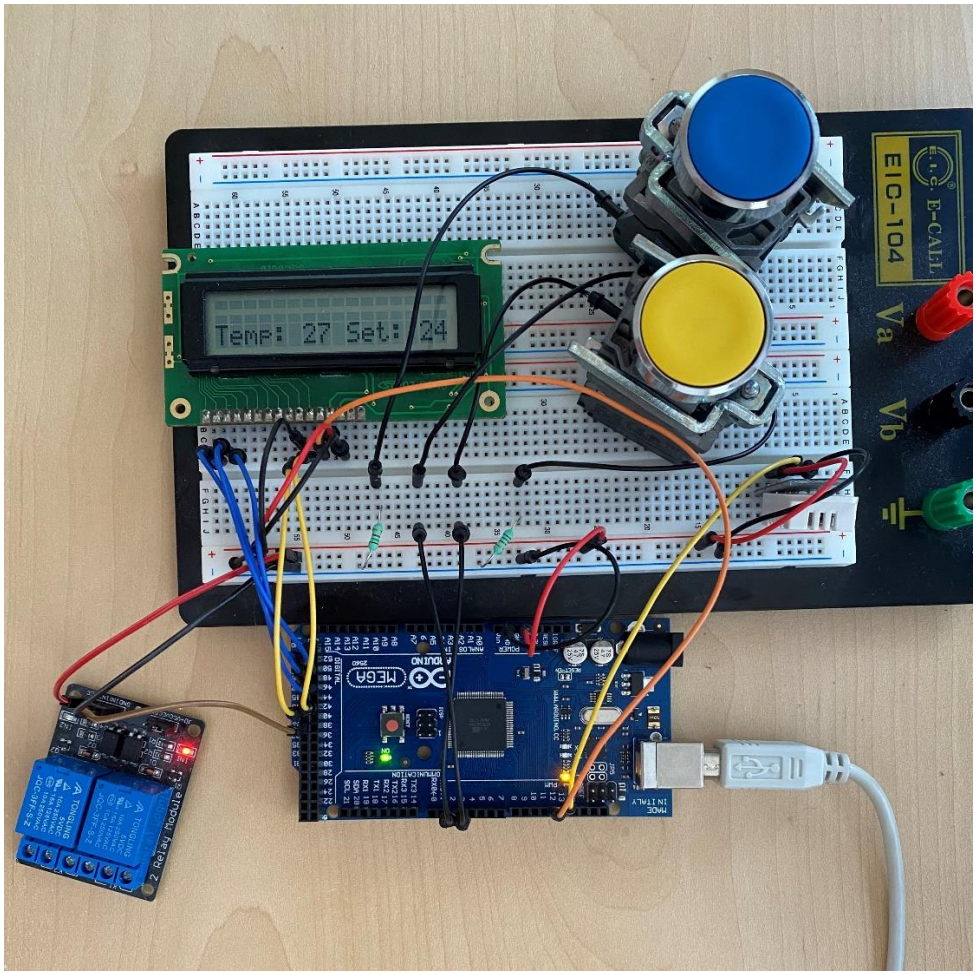
Термостатът е устройство, което има за цел да улесни поддържането на постоянна температура в дадено помещение. С помощта на температурни сензори измерва текущата температура, сравнява я със зададената от потребителя температура и чрез един регулиращ елемент я настройва до зададената стойност.

Блокова схема



Електрическа схема





Списък съставни части

- Микроконтролер Arduino Mega (работи и с Uno)
- Breadboard
- Сензор за температура DHT22
- LCD дисплей 16x2
- 2 пуш-бутона
- Електромагнитно реле
- Резистор 1.5 kΩ

Сорс код

```
#include <LiquidCrystal.h>

#include <DHT.h>

#define LCD_CONTRAST 12
#define DHTPIN 13
#define RELAYPIN 39
#define BTN1 2
#define BTN2 3

LiquidCrystal lcd(42, 43, 53, 50, 49, 46);
DHT dht(DHTPIN, DHT22);

int targetTemp = 20;
const int hysteresis = 1;
static unsigned long last_interrupt_time = 0;

void setup()
{
  pinMode(RELAYPIN, OUTPUT);
  pinMode(LCD_CONTRAST, OUTPUT);
  pinMode(BTN1, INPUT_PULLUP);
  pinMode(BTN2, INPUT_PULLUP);
  analogWrite(LCD_CONTRAST, 15);
  Serial.begin(9600);
  lcd.begin(16, 2);
  dht.begin();

  attachInterrupt(digitalPinToInterrupt(BTN1), RaiseTemp, RISING);
  attachInterrupt(digitalPinToInterrupt(BTN2), LowerTemp, RISING);
}

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

void UpdateTemp()
{
  int temp = dht.readTemperature();
  Serial.println(temp);
  lcd.clear();
  lcd.print("Current: ");
  lcd.print(temp);
  lcd.setCursor(0, 1);
  lcd.print("Target: ");
```

```
  lcd.print(targetTemp);

  if (temp < targetTemp - hysteresis)
  {
    digitalWrite(RELAYPIN, HIGH);
  }
  else if (temp > targetTemp)
  {
    digitalWrite(RELAYPIN, LOW);
  }
}

void RaiseTemp()
{
  unsigned long interrupt_time = millis();
  if (interrupt_time - last_interrupt_time > 300)
  {
    targetTemp++;
    UpdateTemp();
  }
  last_interrupt_time = interrupt_time;
}

void LowerTemp()
{
  unsigned long interrupt_time = millis();
  if (interrupt_time - last_interrupt_time > 300)
  {
    targetTemp--;
    UpdateTemp();
  }
  last_interrupt_time = interrupt_time;
}
```

Обяснения към кода

В началото на кода се добавят необходимите за функционирането му библиотеки, LiquidCrystal.h и DHT.h (само в реалния прототип, в Tinkercad сензорът е друг). След това се дефинират пиновете, към които са свързани компонентите (отново има разлика между симулацията и реалния проект), както и LCD дисплея и температурния сензор.

Инициализират се няколко променливи, зададената от потребителя температура (която по подразбиране е 20 градуса, и може да бъде променяна в последствие) и хистерезиса.

Използване на хистерезис

Хистерезисът представлява отклонение от желаната температура, което има за цел да предотврати прекомерното включване/изключване на механизмите за регулация на температурата.

След това се инициализира променлива, която има за цел да пази времето от последното натискане на бутон, за да се уверим, че програмата отчита всяко натискане само веднъж (debouncing).

В setup() се задава pinMode на всеки пин, след което чрез ШИМ се задава контраста на LCD дисплея, така, че да бъде лесно четим. Започва се серийна комуникация с цел дебъгване и се закача interrupt за бутоните, който работи с rising edge detection.

Използване на “input_pullup”?

Използва се вграденият pull-up резистор на Arduino при свързването на бутоните, за да се избегне „плуването“ на стойността на пиновете.

Използване на “interrupt”?

Използва се interrupt при бутоните, за да може веднага да бъде регистриран и обработен натискът на един от тях от потребителя, дори докато се извършва отчитане на температурата. В самите interrupt handler-и се проверява времето от последния interrupt, за да се избегне многократното инкрементиране на променливата с един натиск на бутона.

На всеки 5 секунди програмата извиква UpdateTemp(), който отчита температурата, показва я на дисплея заедно със зададената, и ги сравнява. (В Tinkercad присъства и метод GetTemp(), който отчита аналоговия сигнал от TMP36 сензора, и го преобразува в числена стойност.) Ако текущата температура е по-ниска от зададената (включвайки хистерезиса), то електромагнитното реле (което по презумпция е свързано към отоплителна система, например радиатор или духалка) се включва. Ако е достигнала желаната температура, или се надвишава, релето се изключва.

Последните два метода са свързани с бутоните, в тях се проверява последното време от извикването им, и ако е повече от 300ms, се инкрементира или намалява променливата, в която се пази зададената от потребителя температура.

Заключение

Проектът успешно изпълнява основната функционалност на термостата – отчитане на температура, съпоставяне на текущата температура със зададената от потребителя температура, включване или изключване на регулиращите елементи. Подобен проект би улеснил поддържането на постоянна температура в дадено помещение и би бил приложим в система за домашна автоматизация.

Линк към проекта в Tinkercad

https://www.tinkercad.com/things/fkQKvtYbK0N