



КУРСОВ ПРОЕКТ – ТЕРМОСТАТ

БОЯН БЕЛЧЕВ, ИТ КАРИЕРА 2022

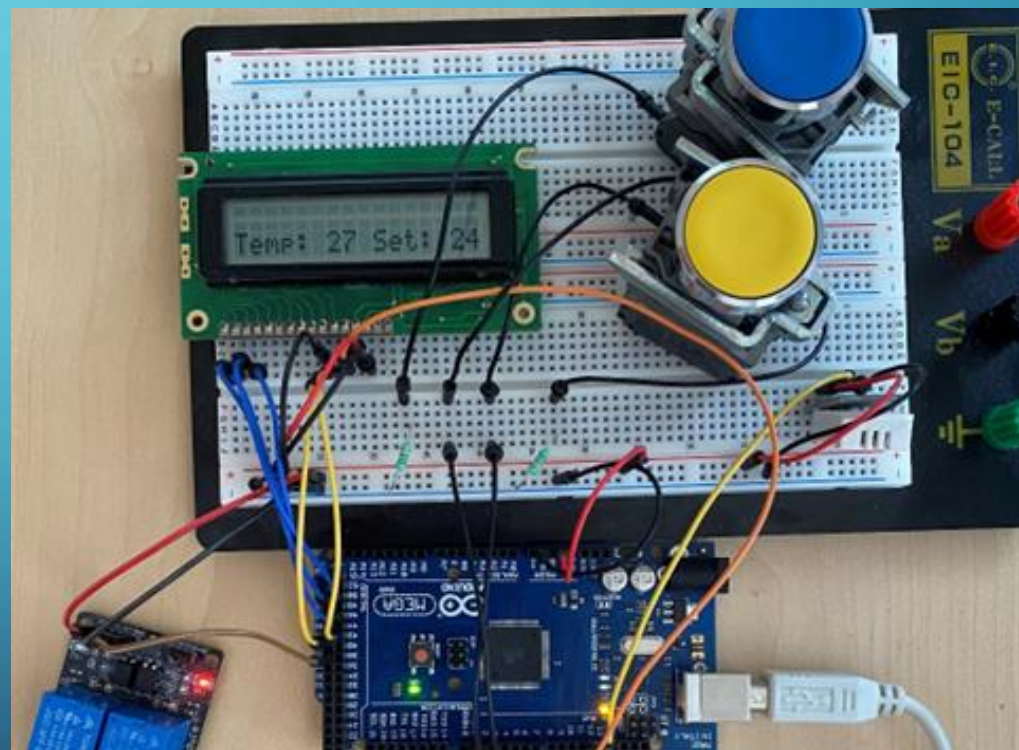
КАКВА Е ИДЕЯТА?

- 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.
- Термостатите често се използват в различни системи за автоматизация.



КАКВА Е РЕАЛИЗАЦИЯТА?

- Идеята е реализирана, използвайки микроконтролер Arduino, температурен сензор, електромагнитно реле, дисплей и два бутона.



КАК РАБОТИ?

```
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);
    }
}
```

- На всеки 5 секунди програмата отчита температурата, показва я на дисплея заедно със зададената, и ги сравнява. Ако текущата температура е по-ниска от зададената, то електромагнитното реле се включва. Ако е достигнала желаната температура, или я надвишава, релето е изключено.
- Потребителят може да променя зададената температура използвайки двата бутона.

ИНТЕРЕСНИ МОМЕНТИ НА КОДА...

- *Защо съм използвал хистерезис?*
- Хистерезисът представлява отклонение от желаната температура, което има за цел да предотврати прекомерното включване/изключване на механизмите за регулация на температурата.

ИНТЕРЕСНИ МОМЕНТИ НА КОДА

Защо съм използвал "input_pullup"?

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

Защо съм използвал "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.

```
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);
}
```

ИНТЕРЕСНИ МОМЕНТИ НА КОДА

```
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;
}
```

- *Debouncing?*
- За да се уверим, че програмата отчита всяко натискане само веднъж, се инициализира променлива, която има за цел да пази времето от последното натискане на бутон.
- В последните два метода се проверява последното време от извикването им, и само ако е повече от 300ms, се инкрементира или намалява променливата, в която се пази зададената от потребителя температура.

БЛАГОДАРЯ ЗА ВНИМАНИЕТО!

