A picture containing calendar

Description automatically generated

Институт за математику и информатику

Природно-математички факултет

Универзитет у Крагујевцу

**Завршни пројекат из предмета Микропроцесорски системи**

**Тема: Паметни радијатор који омогућава задавање температуре и временски период укључења и искључења**

Студент: Професор:

Лука Ристовић 83/2018 др Александар Пеулић

Фебруар 2023.

Садржај

[1. Увод 2](#_Toc126801555)

[2. Објашњавање алгоритма и кода 2](#_Toc126801556)

[3. Симулација програма у Proteus 8 програму 13](#_Toc126801557)

[4. Корисничко упутство 14](#_Toc126801558)

# 1. Увод

Живимо у времену где је веома важно штедети електричну као и било које друге енергију. Самим тим се производи све више уређаја са оптималном потрошњом истих. Циљ овог пројекта (Паметни радијатор који омогућава задавање температуре и временски период укључења и искључења) је концептуални развој и реализација радијатора зарад оптималнијег коришћења електричне енергије а притом да на што ефикаснији начин људе згреју у својим домовима у хладним данима.

За реализацију овог пројекта је коришћен програм Proteus који служи за симулацију и софтвер STM32CubeIDE у коме је писан код за микроконтролер STM32F103C6.

A picture containing indoor, person

Description automatically generated Слика 1. Илустрован приказ паметниг радијатора

# 2. Објашњавање алгоритма и кода

У овом поглављу ћемо детаљно описати логику и сам код програма. Овај радиатор у себи садржи један LCD, један термостат, један сензор који мери тренутну температуру просторије,дугме за паљење и гашење самог радијатора и један тимер за паљење/гашење радијатора у одређеном тренутку.

Кренућемо од кода за LCD (Liquid Crystal Display).

Text

Description automatically generated with medium confidence

Слика 2. Изглед LCD функција

У овом делу кода смо направили две функције: LCD\_Init и LCD. LCD\_Init функција служи за иницијализацију само екрана. Као што је на пример чишћење екрана, подешавање курсора на почетну позицију за неки испис, итд. Друга функција служи да извршава послате наредбе што од малопређашње функције тако и касније у коду за разне исписе.

Сам радијатор се може укључити на стандардан ON/OFF прекидач и када се он притисне, покреће се такозвани прекид у програму који извршава следећи код.

A screenshot of a computer

Description automatically generated with medium confidence

Слика 3. Функција која је задужена за прекид када се притисне дугме

На почетку програма променљива prekidac има вредност 0, што значи када се први пут дугме притисне, радијатор се пали, симулирано зеленом диодом која је повезана на микроконтполер преко порта В на пин 10. Та зелена диода се укључује као и малопре описан екран и радијатор креће са радом. Поред зелене ту је и црвена диода која симулира угашен систем ( на почетку извршавања програма оно је укључена).

У главом делу програма у главној беспоначној петљи **while**(1) се налази следећи код:

A picture containing table

Description automatically generated

Слика 4. Први део петље у главном делу програма

Први део петље у главном делу програма почиње тако што смо убацили код у бесконачно чекање. То бесконачно чекање се прекида када корисник покрене систем притиском на дугме. Променљива flag има на почетку вредност 1 и та вредност се мења тек покретањем функције за прекид (Слика 3.). Након тога програм прво чита вредност са термостата (у нашем случају симулације то је потенциометар). Ту се налази температура коју је корисник задао радијатору да достигне и одржава. Минимална температура је 16 а максимална 34 степени докле радијатор може грејати. Затим се та задата температура исписује на екран, са следећом поруком: „Задата Темп: Х“, где је Х прочитана температура са термометра. Након тога систем мери тренутну температуру просторије и смешта је у променљиву trenutnaTemperatura (због ограниченог броја пинова који раде у програму Протеус, ова температура је ручно постављена на 23 степена целзијуса).

Graphical user interface, text, application

Description automatically generated

Слика 5. Други део петље у главном делу програма

Потом се тренутна температура собе исписује на екран. И последњи део while петље се односи на проверу да ли је тренутна температура просторије већа или једнака од задате температуре на термостату. Уколико је мања, радијатор наставља са грејањем и исписивањем тренутне и задате температуре у просторији, док уколико је једна или виша, радијатор исписује на екрану поруку о гашењу која гласи „Gasim se...“ и почиње са одбројавањем (10секунди) до гашења, након чега се и угаси.

Овај радијатор поседује и тајмерски прекид, што значи да на одређено време проверава да ли треба укључити/искључити радијатор. Он се активирао у подешавањима које ћемо ускоро описати, имплементацијом функције:

void HAL\_TIM\_PeriodElapsedCallback(TIM\_HandleTypeDef \*htim)

као и дозвољавањем његовог коришћења следећом линијом кода у главном делу програма изнад главне while(1) петље:

HAL\_TIM\_Base\_Start\_IT(&htim2);

Подешавања везана за сам хардвер микроконтролера смо обавили на следећи начин:

Graphical user interface

Description automatically generated

Слика 6. Поглед дефинисаних пинова за микроконтролер

На самом почетку ћемо истакнути да смо користили екстерни клок, због тајмера и то се може видети на слици испод:

Graphical user interface, text, application

Description automatically generated

Слика 7. RCC Mode and Configuration

Радни такт износи максималних 72МHz за овај процесор, као и за ABP1 Timer.

Diagram

Description automatically generated

Слика 8. Clock Configuration

За аналогно-дигиталну конверзију смо користили три пина, PA0, PB1 и PА7. У подешавањима смо дозволили сталну конверзију и број конверзија наместили на три.

Graphical user interface, application

Description automatically generated

Слика 9.

Конфигурација за АДЦ

За наше On/Off дугме смо користили пин 0 на порту Б.

Graphical user interface, application, table

Description automatically generated

Слика 10. Конфигурација за дугме које је задужено за укључивање/искључивање

Такође смо у подешавањима дозволили и прекид за дугме као и за тимер.

Graphical user interface, text, application

Description automatically generated

Слика 11. Конфигурација за оба прекида

Што се тиче временског прекида, ту смо поставили следеће вредности:

Graphical user interface, text, application, email

Description automatically generated

Слика 12. Конфигурација за временски прекид

Clock Source смо поставили на Internal Clock, док смо Prescaler поставили на 39999 а Counter Period на 1999. Додатно смо изменили још три подешавања која су обележена на слици 12.

На овај начин смо добили временски прекид на сваких 1,1111 секунду.

# 3. Симулација програма у Proteus 8 програму

Graphical user interface, text, application

Description automatically generatedОд електричних компоненти смо користили следеће:

1. Једно дугме
2. Лед диоде (црвену и зелену)
3. Три отпорника (два од 330 ома и један од 4,7к)
4. Три потенциометра (симулирају сензор топлоте, термостат и тајмер)
5. Један LCD
6. STM32F103C6 плочу

Слика 13. Компоненте

Chart

Description automatically generated with medium confidence

Слика 14. Изглед електричне шеме

# 4. **Корисничко упутство**

При самом покретању симулације, радијатор је искључен док светли црвена лед диода.

Радијатор се покреће као што смо већ и описали притиском на дугме које је повезано на PB0.

Након притиска на дугме, радијатор очитава задату температуру на термостату као и тренутну температуру са сензона и исписује их на екран.

Diagram

Description automatically generated with medium confidence

Слика 15. Испис задата температуре са термостата

Уколико радијатор достигне жељену температуру, исписује поруку за гашење и након 10 секунди се гаси.

Diagram

Description automatically generated

Слика 16. Радијатор је достигао жељену температуру

На самом крају, корисник има могућност да подеси који жељени дан у недељи жели да радијатор буде укључен тако што ће на потенциометру ТИМЕР да подеси жељени опсег. У зависности од подешеног радијатор ће се палити у то време уколико је тада искључен.

**НАПОМЕНА: Због ограничења тимер интерупта у Протеусу, ову могућност није могуће визуелно имплементирати. Остале споменуте могућности се могу тестирати уколико се искључе опције за тимер прекид у самом STM32CubeIDE програму.**