

ISSN 1311-3321 (print)
ISSN 2535-1028 (CD-ROM)
ISSN 2603-4123 (on-line)

UNIVERSITY OF RUSE “Angel Kanchev”
РУСЕНСКИ УНИВЕРСИТЕТ “Ангел Кънчев”

BSc, MSc and PhD Students & Young Scientists
Студенти, докторанти и млади учени

PROCEEDINGS

Volume 57, book 3.4.

Electrical Engineering, Electronics and Automation
&
Communication and Computer Technologies

НАУЧНИ ТРУДОВЕ

Том 57, серия 3.4.

Електротехника, електроника и автоматика
&
Комуникационна и компютърна техника

Ruse
Русе
2018

Volume 57 of PROCEEDINGS includes the papers presented at the scientific conference RU & SU'18, organized and conducted by University of Ruse "Angel Kanchev" and the Union of Scientists - Ruse. Series 3.4. contains papers reported in the Electrical Engineering, Electronics and Automation & Communication and Computer Technologies section.

Book	Code	Faculty and Section
Faculty of Electrical Engineering Electronics and Automation		
3.1	FRI-10.326-1-EEEE	Electrical Engineering, Electronics and Automation
3.2	FRI-2G.303-1-CST	Communication Systems & Network Technologies
3.3	FRI-2G.302-1-CSNT	Computer Systems & Technologies
3.4	FRI-SSS-EEEE	Electrical Engineering, Electronics and Automation
	FRI-SSS-CCT	Communication and Computer Technologies

The papers have been reviewed.

ISSN 1311-3321 (print)

ISSN 2535-1028 (CD-ROM)

ISSN 2603-4123 (on-line) Copyright © authors

The issue was included in the international ISSN database, available at <https://portal.issn.org/>.

The online edition is registered in the portal ROAD scientific resources online open access



PROGRAMME COMMITTEE

- **Prof. Velizara Pencheva, PhD,**
University of Ruse, Bulgaria
- **Prof. Leon Rothkrantz**
Delft University of Technology, Netherlands
- **Assoc. Prof. Antonio Jose Mendes,**
University of Coimbra, Portugal
- **Prof. Ville Leppanen,**
University of Turku, Finland
- **Assoc. Prof. Marco Porta,**
University of Pavia, Italy
- **Prof. Douglas Harms,**
DePauw University, USA
- **Prof. Ismo Hakala, PhD,**
University of Jyväskylä, Finland
- **Prof. Dr. Artur Jutman,**
Tallinn University of Technology, Estonia
- **Prof. RNDr. Vladimír Tvarozek, PhD,**
Slovak University of Technology in Bratislava, Bratislava, Slovakia
- **Doc. Ing. Zuzana Palkova, PhD,**
Slovak University of Agriculture in Nitra, Nitra, Slovakia
- **Andrzej Tutaj, PhD,**
AGH University of Science and Technology, Krakow, Poland
- **Assoc. Prof. Behiç TEKİN, PhD,**
EGE University, Izmir, Turkey,
- **Prof. Valentin NEDEFF Dr. eng. Dr.h.c.,**
“Vasile Alecsandri” University of Bacău, Romania
- **Dr. Cătălin POPA,**
“Mircea cel Bătrân” Naval Academy, Constantza, Romania
- **Prof. dr Larisa Jovanović,**
Alfa University, Belgrade, Serbia
- **Prof. dr hab. Edmund LORENCOWICZ,**
University of Life Sciences in Lublin, Poland
- **Assoc. Prof. Ion MIERLUS - MAZILU, PhD,**
Technical University of Civil Engineering, Bucharest, Romania
- **Prof. Dojčić Vojvodić PhD,**
Faculty of Philosophy, University of Novi Sad, Serbi
- **Assoc. Prof. Alexandrache Carmen, PhD,**
Department of Teacher Training, “Dunarea de Jos”, Galati University, Romania
- **Prof. Alberto Cabada,**
University of Santiago de Compostela, Faculty of Mathematics, Santiago de Compostela, Spain
- **Assoc. Prof. Dr. Mehmet Şahin,**
Necmettin Erbakan University, Ahmet Keleşoğlu Faculty of Education, Konya, Turkey
- **Assoc. Prof. Erika Gyöngyösi Wiersum, PhD,**
Eszterházy Károly University, Comenius Campus in Sárospatak, Institute of Real Sciences, Sárospatak, Hungary
- **Anna Klimentova, PhD,**
Constantine the Philosopher University in Nitra, Slovakia
- **Prof. Igor Kevorkovich Danilov, DSc,**
Yuri Gagarin State Technical University of Saratov, Russia

- **Prof. Aleksander Valentinov Sladkowski, DSc,**
Silesian University of Technology, Poland
- **Prof. Pether Shulte, PhD,**
Institute for European Affairs (INEA), Dusseldorf, Germany
- **Prof. Aslitdin Nizamov, DSc., PhD,**
Bukhara Engineering-Technological Institute, Bukhara, Uzbekistan
- **Prof. Marina Sheresheva, PhD,**
Lomonosov Moscow State University, Russia
- **Prof. Erik Dahlquist, PhD,**
Mälardalen University, Sweden
- **Prof. Erik Lindhult, PhD,**
Mälardalen University, Sweden
- **Prof. Annika Kunnasvirta, PhD,**
Turku University of Applied Sciences, Finland
- **Prof. Walter Leal, Dr. (mult.) Dr.h.c. (mult.),**
Hamburg University of Applied Sciences, Germany
- **Prof. Dr. Gerhard Fiolka,**
University of Fribourg, Switzerland
- **Prof. Haluk Kabaalioglu, PhD,**
Yeditepe University, Turkey
- **Prof. Silva Alves, PhD,**
University of Lisbon, Portugal
- **Hanneke van Bruggen,**
Appeldoorn, The Netherlands
- **Nino Žganec,**
President of European Association of Schools of Social Work, Assoc. Prof. at the Department of Social Work,
University of Zagreb, Croatia
- **Prof. Violeta Jotova,**
Направление „Педиатрия“ в УМБАЛ „Св. Марина“ – Варна, България
- **Assoc. Prof. Tanya Timeva, MD, PhD,**
Obstetrics and Gynecology Hospital "Dr. Shterev", Sofia, Bulgaria
- **Prof. Kiril Stoychev, PhD,**
Institute of Metal Science, Equipment and Technologies “Acad. A. Balevsci” with Hydroaerodynamics centre
– BAS, Bulgaria
- **Assoc. Prof. Mark Shamtsyan, PhD,**
Technical University, Saint Petersburg, Russia
- **Assoc. Prof. Oleksii Gubenia, PhD,**
National University of Food Technologies, Kiev, Ukraine
- **Assoc. Prof. Olexandr Zaichuk, DSc,**
Ukrainian State University of Chemical Technology, Dnepropetrovsk, Ukraine
- **Prof. Eugene Stefanski, DSc,**
Samara University, Russia
- **Doc. Dr. Tatiana Strokovskaya,**
International University of Nature “Dubna”, Dubna, Russia
- **Prof. DSc. Petar Sotirow,**
Maria Curie-Skłodowska University of Lublin, Poland
- **Prof. Papken Ehasar Hovsepian,**
Sheffield Hallam University, Sheffield, UK
- **Assoc. Prof. Krassimir Dochev Dochev, PhD,**
University of Portsmouth School of Engineering, UK
- **Mariana Yordanova Docheva, PhD,**
University of Portsmouth School of Engineering, UK

- **Assoc. Prof. Ivan Antonov Lukanov, PhD,**
University of Botswana, Faculty of Engineering and Technology, Gaborone, Botswana
- **Assoc. Prof. Petko Vladev Petkov, PhD,**
Research Associate Cardiff University, UK
- **Prof. Stepan Terzian DSc,**
Bulgarian Academy of Science, Bulgaria
- **Prof. Dr. Gabriel Negreanu,**
University Politehnica of Bucharest, Romania

ORGANISING COMMITTEE

♦ **ORGANIZED BY: University of Ruse (UR) and
Union of Scientists (US) - Ruse**

♦ **ORGANISING COMMITTEE:**

• **Chairpersons:**

COR. MEM Prof. Hristo Beloev, DTSc – Rector of UR, Chairperson of US - Ruse

• **Scientific Secretary:**

Prof. Diana Antonova PhD, Vice-Rector Research,
dantonova@uni-ruse.bg, 082/888 249

♦ **MEMBERS:**

Assoc. Prof. Kaloyan Stoyanov, PhD
Assoc. prof. Velina Bozduganova, PhD,
Assoc. Prof. Kiril Sirakov, PhD,
Assoc. Prof. Milko Marinov, PhD,
Pr. Assist. Ivanka Tsvetkova, PhD,
Assoc. Prof. Simeon Iliev, PhD,
Assoc. Prof. Pavel Vitliemov, PhD,
Assoc. Prof. Mimi Kornazheva, PhD,
Boryana Stancheva, PhD,
Prof. Vladimir Chukov, DESc,
Pr. Assist. Krasimir Koev, PhD,
Prof. Juliana Popova, PhD,
Pr. Assist. Hristina Sokolova, PhD,
Pr. Assist. Magdalena Andreeva, PhD,
Assoc. Prof. Emilia Velikova, PhD,
Assoc. prof. Bagryana Ilieva, PhD,
Pr. Assist. Reneta Zlateva, PhD,
Pr. Assist. Velislava Doneva, PhD,
Assoc. Prof. Stefka Mindova,
Assoc. prof. Sasho Nunev, PhD,
Assoc. Prof. Despina Georgieva, PhD,
Pr. Assist. Vanya Panteleeva, PhD,
Assoc. Prof. Emil Trifonov, PhD,
Assoc. Prof. Galina Lecheva;
Assist. Prof. Milen Sapundzhiev, PhD;
Assoc. Prof. Tsvetan Dimitrov, PgD,
Assoc. Prof. Nastya Ivanova, PhD,

◆ **REVIEWERS:**

Prof. Ventsislav Valchev, PhD
Prof. Plamen Daskalov, PhD
Prof. Rozalina Dimova, PhD
Prof. Stanimir Sadinov, PhD
Prof. Borislav Bedzhev, DSc
Prof. Ivan Tsonev, PhD
Prof. Mihail Iliev, DSc
Assoc. Prof. Stefka Atanasova, PhD
Assoc. Prof. Petya Veleva, PhD
Assoc. Prof. Rusin Tsonev, PhD
Assoc. Prof. Lyudmil Mihailov, PhD
Assoc. Prof. Todorka Georgieva, PhD
Assoc. Prof. Hristo Valchanov, PhD
Assoc. Prof. Zheino Zheinov, PhD
Assoc. Prof. Venetka Aleksieva, PhD
Assoc. Prof. Milko Marinov, PhD

ELECTRICAL ENGINEERING, ELECTRONICS AND AUTOMATION

Content

1. FRI-SSS-EEEE-01	11
Development of Web-based Mobile System for Temperature and Humidity Monitoring in a Greenhouse	
<i>Georgi Petrov, Nadezhda Paskova</i>	
2. FRI-SSS-EEEE-02	16
Smart Irrigation System	
<i>Sechkin Remzi, Tsvetelina Georgieva</i>	
3. FRI-SSS-EEEE-03	19
Methods to Determine the Overflow or Deficiency of Chemical Elements in Plants	
<i>Valentin Velez, Nadezhda Paskova</i>	
4. FRI-SSS-EEEE-04	25
Software Applications for Simulation of Wireless Sensor Networks	
<i>Teodora Ivanova, Tsvetelina Georgieva</i>	
5. FRI-SSS-EEEE-05	31
Development of Measuring System Pulse and Blood Pressure of the Human Body	
<i>Svetoslav Petrov, Tsvetelina Georgieva</i>	
6. FRI-SSS-EEEE-06	36
Quality Evaluation of Dried Apricots	
<i>Darinka Ivanova Ilieva-Stefanova, Donka Ivanova, Nikolay Valov</i>	
7. FRI-SSS-EEEE-07	41
Generator for Ensuring of Alternative Power Supply in Buildings	
<i>Bojidar Tasev, Daniel Kajtsanov, Seher Kadirova</i>	
8. FRI-SSS-EEEE-08	44
Maximum Power Point Tracking Unit for Battery Charging from a Photovoltaic	
<i>Penko Binkov, Seher Kadirova</i>	
9. FRI-SSS-EEEE-09	49
IR/R Oxymeter	
<i>Aleksandr Furmanov, Aneliya Manukova</i>	

COMMUNICATION AND COMPUTER TECHNOLOGIES

Content

1. FRI-SSS-CCT-01	54
Web-Based Project Management System	
<i>Tsvetelina Mladenova, Yordan Kalmukov</i>	
2. FRI-SSS-CCT-02	60
Web-Based System for Making Doctor's Appointments	
<i>Tsvetelina Mladenova, Birdzhan Hasan, Yordan Kalmukov</i>	
3. FRI-SSS-CCT-03	64
Combinational Circuits Design Using Logisim 2.7.1	
<i>Milutin Barakov, Adriana Borodzhieva</i>	
4. FRI-SSS-CCT-04	71
Developing Web Services in JAVA with Spring Framework	
<i>Teodor Todorov, Tsvetelina Mladenova, Yordan Kalmukov</i>	
5. FRI-SSS-CCT-05	76
Android Mobile Application for Managing Accounts with ERC20 Cryptocurren	
<i>Teodor Todorov, Ivaylo Rusev</i>	
6. FRI-SSS-CCT-06	79
An Interactive Educational Tool for Studying the Hamming Code Algorithm for Encoding and Decoding Data Using Matrix Method	
<i>Ivan Todorov, Galina Ivanova</i>	
7. FRI-SSS-CCT-07	85
An Interactive Software for Studying the Algorithm for Data Encoding Using Cyclic Code	
<i>Velizar Todorov, Galina Ivanova</i>	
8. FRI-SSS-CCT-08	91
An Android Mobile Application for Planning an Effective Control of Automobile Service Periods	
<i>Aleksandar Dzhrekos, Galina Ivanova</i>	
9. FRI-SSS-CCT-09	97
An Educational Software for Solving Tasks to Control Arithmetic and Logical Operations	
<i>Tsvetomir Goranov, Galina Ivanova</i>	
10. FRI-SSS-CCT-10	103
An Innovative Method for Studying the Components in Computer Motherboard Using Augmented Reality	
<i>Lenimir Zdravkov, Galina Ivanova</i>	
11. FRI-SSS-CCT-11	109
A Survey of Web Based Educational Platforms	
<i>Plamen Dimitrov, Galina Ivanova</i>	
12. FRI-SSS-CCT-12	115
Modified Ring-Based Mutual Exclusion Algorithm with Failover Recovery	
<i>Ivaylo Rashkov, Milen Lukanchevski</i>	
13. FRI-SSS-CCT-13	121
Online System for Erasmus Student Application and Monitoring Incoming Mobilities	
<i>Viktor Bratov, Galina Ivanova</i>	
14. FRI-SSS-CCT-14	127
GoLang – Introduction and Basic Features	

	<i>Teodora Yordanova, George Stefanov, Yordan Kalmukov</i>	
15.	FRI-SSS-CCT-15 REST Services Written in GoLang <i>Teodora Yordanova, George Stefanov, Yordan Kalmukov</i>	133
16.	FRI-SSS-CCT-16 Time Protocol (RFC-868) Server Workload Source <i>Adelina Koleva, Milen Lukanchevski</i>	139
17.	FRI-SSS-CCT-17 Photo Scanned Textures <i>Nikolay Marinov, Irena Valova</i>	145
18.	FRI-SSS-CCT-18 3D Game “Inside the Dream” <i>Denislav Kolev, Irena Valova</i>	150
19.	FRI-SSS-CCT-19 The Knight 3D Game <i>Nikola Rashkov, Irena Valova</i>	155

DEVELOPMENT OF WEB BASED MOBILE SYSTEM FOR TEMPERATURE AND HUMIDITY MONITORING IN A GREENHOUSE¹

Georgi Petrov – Student

Department of Automatics and Mechatronics,
University of Ruse “Angel Kanchev”
Tel.: +359 89 960 0992
E-mail: joropetroov@abv.bg

Nadezhda Paskova – PhD Student

Department of Automatics and Mechatronics,
University of Ruse “Angel Kanchev”
Tel.: +359 888 676
E-mail: npaskova@uni-ruse.bg

Abstract: Each season in the greenhouses are increasingly paying attention to the quality of the microclimate. One of the main components that increase productivity is the appropriate microclimate control technology. The efficient and economical use of energy resources provides an additional opportunity to reduce production costs.

Keywords: Web-based mobile system, microclimate, temperature and humidity measurement

ВЪВЕДЕНИЕ

Има много видове сеизмична защита в областта на гражданското строителство: Съществуват големи разлики между отглеждането на растения в оранжерии и на полето. Отглежданите в оранжерии култури образуват по-голяма растителна маса и дават много по-високи добиви в сравнение с отглежданите на полето. Поради това и начинът на отглеждане и грижа, се диференцира от този, който се прилага на открито. Специално внимание трябва да се обръща на температурата и влажността в оранжерията.

Разнообразието от конструкции и използвани материали е голямо, а изборът им е продиктуван от различни предпоставки. В доклада са представени по-често използваните конструкции и материали за изработка на оранжерии (Фармланд, 2009).

Оранжерия от поликарбонат е показана на Фиг. 1. Поликарбонатът има висока устойчивост на удар. Устойчив е на външни влияния. Материалът не се изменя от атмосферните въздействия, тъй като има специален UV-защитен слой. Не изменя механичните и оптичните си свойства в температурни граници от -40 до $+120^{\circ}\text{C}$. Макар и първоначалната висока цена за материала, инвестицията е препоръчителна поради дългия живот на материала – от около 10 до 15 години.

Най-често поликарбонатните листове се предлагат с дебелина 4mm. В зависимост от дебелината на листа има пропускливост от 84% до 90% и често превишава светлопропускливостта на силикатните стъкла. За приложения, като зимни градини например, плоскостите се изработват със слънчево-бронзов оттенък и полу-прозрачен бял опал. И двата вида поглъщат отблясъците и слънчевата топлина, а плоскостите с бял опал създават разсеяна естествена светлина.

¹ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: РАЗРАБОТВАНЕ НА WEB БАЗИРАНА МОБИЛНА СИСТЕМА ЗА МОНИТОРИНГ НА ТЕМПЕРАТУРА И ВЛАЖНОСТ В ОРАНЖЕРИЯ.



Фиг. 1. Оранжерия от поликарбонат

Оранжерии от полиетилен са представени на Фиг. 2. Полиетиленовите оранжерии приличат на полиетиленовите тунели, но са с по-големи размери. В надлъжна посока също се монтират стоманени пръти. В повечето случаи тези оранжерии са с широчина 6 метра, височина до билото – 2.4 метра и дължина - около 30-40 метра. Този вид оранжерии имат редица предимства: полиетиленовото покритие се поставя и закрепва лесно, подходящи са за малки терени, дворни пространства, а също и за райони с повече валежи, защото водата се оттича много лесно.



Фиг. 2. Оранжерия от полиетилен

Стъклото също е предпочитан и традиционен материал за покритие на оранжерии, един от най-често използваните. Оранжериите със стъклено покритие (Фиг. 3) са от т. нар. “холандски тип”. Това е най-често прилагания тип оранжерии. Подходящи са за отглеждане на зеленчуци и цветя. Стандартната височина е около 4.5 - 5 метра, но има и по-високи. Стъклото е един от често използваните материали за дълготрайна употреба. Позволява максимално пропускане на светлина, има отлични отразяващи свойства и е лесно за почистване. Осигурява много добра защита на културите при лоши метеорологични условия. За остъкляване се използват безцветни, светлопропускливи, гладки стъкла с дебелина минимум 4мм.

Стъклото е скъпо, крехко и чупливо и това предполага честа смяна на покритието. Поради тази причина, а и с напредването на технологиите, все повече пластмасови изделия намират приложение в строителството на оранжерии последните 20 години.



Фиг. 3. Стъклена оранжерия

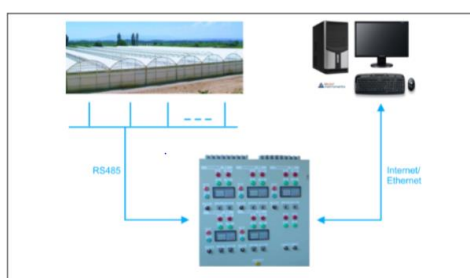
ИЗЛОЖЕНИЕ

Необходимост от следене на температура, влажност и осветеност в оранжерии

С намаляването на количеството получена светлина растенията не могат да изработват такова количества храна, както при по-силна осветеност. По-високите дневни температури ще предизвикат разграждане на по-голямо количество храна, отколкото се изгражда, т.е. процесите на разграждане ще преобладават над процесите на изграждане. Затова е добре и температурата да бъде съобразена, т.е. малко по-ниска. Слънцето затопля с естествена топлина оранжерията, което е за предпочитане пред изкуственото отопление. От друга страна доброто осветяване позволява на растенията да изработват и достатъчно количество храна. Зимните оранжерии обикновено, задържат влага и проветряването спомага да се избегнат болести.

Съществуващи системи за следене на температура и влажност в оранжерия

Съществуват множество системи за мониторинг на температура и влажност в оранжерии. На фиг. 4 е показана система за регулиране на микроклимат в оранжерии. Системата има възможност за регулиране на температурата на въздуха в оранжерията, на относителната влажност на въздуха, на температурата на подаващия тръбопровод, на връщащия тръбопровод, температурата на външния въздух и скоростта на вятъра. Използва се един контролер за една оранжерия. Реализираният проект е за пет оранжерии и състоянието на наблюдаваните параметри се визуализира на операторско табло. Системата има възможност за отдалечено управление, за архив на данните по температура и влажност, експортирането им до Excel формат, графично или таблично представяне, възможност за достъп на софтуера по интернет.



Фиг. 4. Система за регулиране на микроклимат в оранжерии

Разработване на web-базирана система за мониторинг на температура и влажност в оранжерия.

За изграждането на система за измерване на температура и относителна влажност в оранжерия са избрани модули на Ардуино (Karvinen K. & Karvinen T., 2011). Към микроконтролерът Ардуино е свързан модул за безжична комуникация и сензор за измерване на температура и влажност.

Ардуино Uno е микроконтролерна развойна платка с ATmega328P AVR микроконтролер. Има 14 цифрови входно-изходни (I/O) порта, 6 аналогови входа, 16 MHz кварцов резонатор, четири светодиода (един потребителски, свързан на 13-ти цифров I/O порт и три, които индикират работата на платката: ON, Tx и Rx), USB конектор, захранващ куплунг, бутон за рестартиране и ICSP 41 конектор. Шест от цифровите I/O порта могат да се използват като PWM (ШИМ) изходи. Свързването с компютър се осъществява чрез USB кабел USB A – USB B. Може да се захранва през USB порта на компютъра или от външен източник, като превключването между различните начини за захранване е автоматично.



Фиг. 5. Микроконтролер Arduino Uno



Фиг. 6. Модул XBee

Характеристиките на микроконтролера са следните:

- микроконтролер: ATmega328P;
- работно напрежение: 5 V;
- захранващо напрежение (препоръчително): 7-12 V;
- цифрови I / O порта: 14 броя (от които 6 могат да са PWM изходи);
- аналогови входове: 6 броя;
- максимален ток на I / O порт: 40 mA;
- програмируема памет: 32 KB, от които 0.5 KB заети от буутлоудъра;
- SRAM: 2 KB;
- EEPROM: 1 KB;
- тактова честота: 16 MHz.

Радио модулет е от типа XBee 60mW с PCB антена – Серия 1 (802.15.4) Това е 2.4GHz Xbee модул с антена, който използва 802.15.4 протокола, което го прави лесен за използване с набор от серийни команди. Този модул прави комуникацията между микроконтролери, компютри и всякакви други устройства със серийен порт много лесна. Поддържа мрежи с два или повече модула.

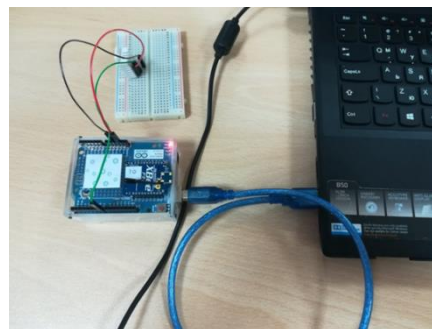
Основните характеристики на радио модула са:

- работно напрежение 3.3V;
- консумация 50mA;
- 60mW изход;
- обхват 100m;
- PCB антена;
- 6 бр. 10-битови АЦП пина;
- 8 бр. цифрови входно-изходни пина;
- 128-битова кодировка;
- възможност за жично или безжично конфигуриране;
- AT или API команди.

DHT-22 е избрания сензор за температура и относителна влажност. Интерфейсът за комуникация е по един проводник. Сензорът е напълно калибриран и не са необходими допълнителни компоненти, за да се получат точни показания за влажност и

температура. Захранването е от 3,3 V до 6,0 V, обхват на влажността от 0 до 100 % с точност до 2% и обхват за температура -20 до 80 °C с точност до 0,5°C.

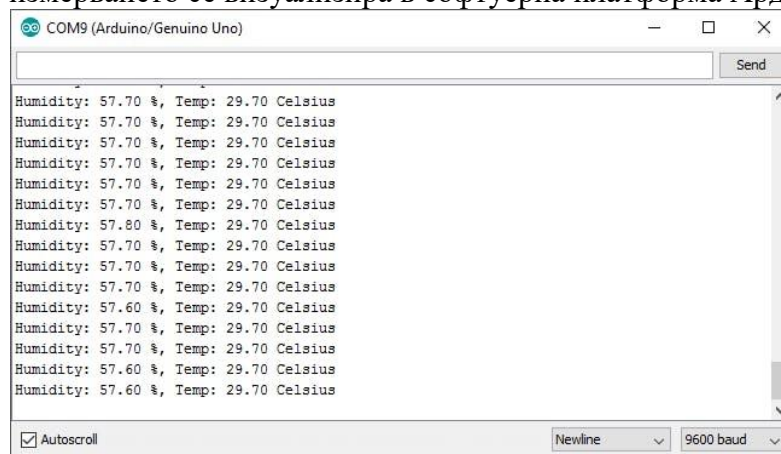
На фиг.8. е представена част от разработената web – базирана система за измерване на основни параметри (температура и влажност) в оранжерия.



Фиг. 7. Сензор за температура DHT-22

Фиг. 8. Web базирана система за оранжерия

Информация от измерването се визуализира в софтуерна платформа Ардуино (фиг. 9).



Фиг. 9. Визуализация на получените данни

ИЗВОДИ

Проектирана е и е разработена web базирана мобилна система за мониторинг на температура и влажност на въздуха в оранжерия.

Избрани са подходящ контролер, сензор и безжичен модул за предаване на данните.

Разработено е и е тествано програмното осигуряване на системата. Предстои да се разработи web базираното приложение.

БЛАГОДАРНОСТ

Изследванията са подкрепени по договор с № 2018-ЕЕА-03 „Изследване и моделиране на процесите отглеждане, преработване и окачествяване на хранителни продукти“, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

Karvinen K., Karvinen T., (2011). Arduino Bots and Gadgets, Published by O'Reilly Media, 250-300

Фармланд България, (2009). Грижа за културите в оранжерии. https://farmland.bg/gruji_za_kulturite_v_oranjerii.html

SMART IRRIGATION SYSTEM²

Sechkin Remzi - Student

Department of Automation and Mechatronics

University of Ruse "Angel Kanchev"

Tel.: +359 87 86 86 953

E-mail: seckinremzi@abv.bg

Assoc. Prof. Tsvetelina Georgieva, PhD

Department of Automation and Mechatronics

University of Ruse "Angel Kanchev"

Phone: +359 82 888 668

E-mail: cgeorgieva@uni.ruse.bg

Abstract: : In order to meet the growing demand in agriculture, intelligent recovery has been made to facilitate production in a healthy way. In this study, the data obtained from the soil moisture sensors controlled by the Arduino Uno controller and according to these data the system is triggered and the pump releases the necessary water for irrigation of the plants. The water pump closes automatically when the soil meets the required water requirements. In addition, data processed in the Arduino Uno controller is transferred via wireless network modules to the interface created on the computer. According to these data, irrigation can be controlled by the remote computer at any time. With this project, the goal is to optimize the use of water resources, to achieve maximum energy savings, to eliminate excessive and unnecessary irrigation problems at the appropriate moment by irrigation.

Keywords: Smart Recovery, Arduino Uno Controller, Interface

ВЪВЕДЕНИЕ

С развитието на технологиите, ползотворния ефект на системите за автоматизация върху човешкия живот се увеличава от ден на ден. Този ефект е най-съществен при технологичните подобрения в селското стопанство, поради това, че то е едно от основните производства в човешкия живот. В това производство напояването е на първо място. За по - здравословно отглеждане на културите е необходимо хомогенно и икономично напояване. Това хомогенно напояване може да бъде реализирано с контролни и измервателни елементи без човешка намеса.

Влажността в почвата се определя от водния капацитет на почвата. Водата е незаменима както за почвата така и за другите живи същества по почвата.

Желанията за влажност на растенията варират между 40-80% в зависимост от вида.

Това е предпоставка за проектиране и създаване на smart система за напояване.

ИЗЛОЖЕНИЕ

Компоненти на smart системата за напояване

Основните компоненти на smart системата за напояване са управляващо устройство, сензори и изпълнителни механизми.

Характеристиките на микроконтролера, който е избран за реализиране на системата са следните:

- микроконтролер: ATmega328P;
- работно напрежение: 5 V;

² Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: ИНТЕЛИГЕНТНА СИСТЕМА ЗА НАПОЯВАНЕ.

- захранващо напрежение (препоръчително): 7-12 V;
- цифрови I / O порта: 14 броя (от които 6 могат да са PWM изходи);
- аналогови входове: 6 броя;
- максимален ток на I / O порт: 40 mA;
- програмируема памет: 32 KB, от които 0.5 KB заети от буутлоудъра;
- SRAM: 2 KB;
- EEPROM: 1 KB;
- тактова честота: 16 MHz.

Капко-разпръскватели в системите за напояване

Водата за капково напояване трябва да достигне корените на растението с помощта на капко-разпръскватели, разположени в тръбата за вода, които са поставени на определени места (Фиг. 1). Капко-разпръсквателите са разположени само по корените на растенията. По този начин се подобрява само растението, което е нужно и се предотвратява растежа на плевели. Корените на растенията се навлажняват и повече пространство се напоява с по-малко вода чрез капково поливане. Тъй като растенията по почвата не са омокрят изцяло, образуването на болести в растението е предотвратено. Предотвратявайки болестите по растенията, се намалява и използването на допълнителни лекарства.

В този доклад е представена разработена смарт напоителна система, общата ѝ структура, системните компоненти и реалната разработка.



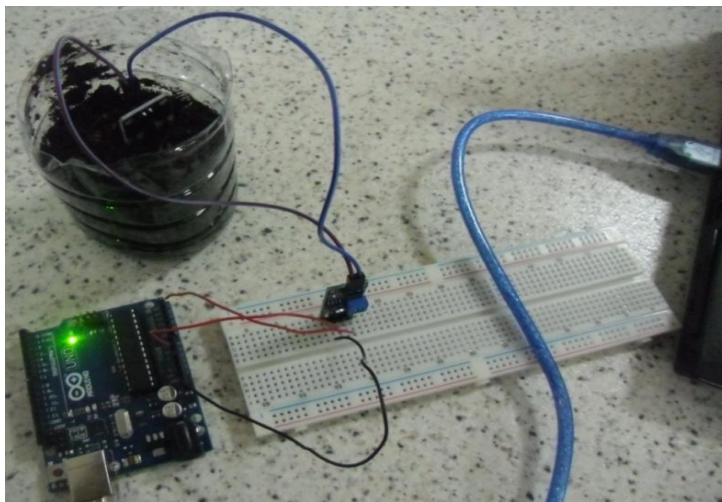
Фиг. 1. Водна тръба и капко-разпръсквателите по тръбата

Налягането по тръбата е свързано с входното налягане на водата, постъпваща в капко-разпръсквателите.

Сензор за влажност

Проведени са експериментални изследвания, за да се определят необходимите диапазони на влажност със сензор, който открива стойността на влагата в почвата. Експериментът се проведе в съд с напоена почва, нормално влажна почва и суха почва. Използва се микроконтролер Arduino Uno за информацията получена от сензора за влагата в почвата. Изходното напрежение на сензора за влага в почвата е между 0 V и 5 V.

На фиг. 2. е представена системата за експериментално изследване за необходимата влага.



Фиг. 2. Експериментално изследване за необходимата влага

ИЗВОДИ

Проектирана е и е разработена web базирана мобилна система за мониторинг на температура и влажност на въздуха в оранжерия.

Избрани са подходящ контролер, сензор и безжичен модул за предаване на данните.

Разработено е и е тествано програмното осигуряване на системата. Предстои да се разработи web базираното приложение.

БЛАГОДАРНОСТ

Изследванията са подкрепени по договор с № 2018-EEA-03 „Изследване и моделиране на процесите отглеждане, преработване и окачествяване на хранителни продукти“, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

Duran U., Demir Ö., Yavuz C., (2012). DHT11 Sıcaklık ve Nem Sensörü İle bilgisayar Destekli Ölçüm ve Kontrol Devresinin İmalatı ve Deneysel Olarak İncelenmesi”, Proje Tabanlı Mekatronik Eğitim Çalıştayı, Çankırı-Ilgaz

Karvinen K., Karvinen T., (2011). Arduino Bots and Gadgets, Published by O'Reilly Media, 250-300

Melgar E.R., Diez C.C., (2012). Arduino and kinect projects, Springer Science+Business Media, 450-500

Wheat D., (2011). Arduino Internals, Springer Science+Business Media, LLC, 342-440

METHODS TO DETERMINE THE OVERFLOW OR DEFICIENCY OF CHEMICAL ELEMENTS IN PLANTS³

Valentin Velev– Student

Department of Automatics and Mechatronics,
University of Ruse “Angel Kanchev”
Tel.: +359 89 783 8456
E-mail: rival393@gmail.com

Nadezhda Paskova – PhD Student

Department of Automatics and Mechatronics,
University of Ruse “Angel Kanchev”
Tel.: +359 888 676
E-mail: npaskova@uni-ruse.bg

Abstract: Under the conditions of precision farming, fertilization is one of the most important agro-technical measures to obtain high yields in quality and quantity and to maintain and increase soil fertility. For the normal growth, development and fructification of vegetable crops and for realization of rich and quality production from them, around 20 chemical elements are needed, which are called nutritional elements.

Keywords: Tomato, macronutrients, image processing

ВЪВЕДЕНИЕ

Производството на зеленчуци е традиционно за България. Климатичните условията на страната ни определят сезонния характер на зеленчукопроизводството и основната цел и значение на оранжерийното производство е да реши проблема с целогодишното снабдяване на пазара с пресни зеленчуци.

У нас доматиите са основна зеленчукова култура с най-голям дял на производство в Пловдивски, Пазарджишки, Петричко-Сандански, Плевенски, Русенски, Великотърновски, Ямболски, Бургаски район. Масовото използване на доматиите се дължи на високата продуктивност и многообразието на приложение на продукцията за прясна консумация и в преработвателната промишленост. И най-вече на отличните хранително-вкусови качества. Хранителната ценност на плодовете се определя от наличието в тях на въглехидрати, органични киселини, витамини, минерални соли, ароматични вещества.

ИЗЛОЖЕНИЕ

Не винаги за нездравият външния вид на доматиите са виновни болести или паразити. В някои случаи сухи листа, бледи растения и забавяне на растежа на културата са в резултат на недостатъчно количество хранителни елементи в почвата. Важно е да се установи, кой химичен елемент липсва на растението. Недостатък на хранителен елемент се определя от външния им вид на растението домати.

Дефицит на хранителни вещества при доматиите

Много често растенията страдат от дефицит или излишък на хранителни вещества. При недостиг на растенията на хранителни вещества се появяват обезцветяване или изкривяване на листата и стъблата (Yuan Z., 2016).

³ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: МЕТОДИ ЗА ОПРЕДЕЛЯНЕ НАЛИЧИЕ ИЛИ ЛИПСА НА ХИМИЧНИ ЕЛЕМЕНТИ В РАСТЕНИЯ.

За да бъдат здрави растенията те се нуждаят от комбинация от растителни вещества. Веществата които са нужни в сравнително големи количества са: азот, калий, фосфор, калций, магнезий и сяра. Останалите хранителни вещества са необходими в по-малки количества като: мед, бор, манган, желязо, цинк и молибден.



Фиг. 1. Влияние на химичните елементи върху растението

В растенията се срещат всички химични елементи, намиращи се в почвата. По-голяма част от тях са абсолютно необходими за нормалното протичане на физиологичните процеси.

Таблица 1. Химичен състав на растителни клетки и тъкани (в % от свежото тегло).

Химичен елемент	%	Химичен елемент	%
Кислород	65,0	Калций	0,15
Водород	10,0	Желязо	0,01
Въглерод	18,0	Йод	0,0001
Азот	3,0	Натрий	0,03
Фосфор	1,0	Цинк	0,0003
Калий	2,0	Сяра	0,12
Магнзий	0,02	Мед	0,0002

Таблица 2. Признаци на дефицит на дефицит при различните химични елементи

Наименование	Означение	Признаци на дефицит
Азот	N	Наблюдава се леко червено оцветяване на жилките и дръжките на листата. Засегнатите първи от този дефицит са по-старите и зрели листа, които постепенно променят нормалния си характерен зелен цвят към много по-бледо зелено. При напредване на дефицита по-старите листа стават равномерно жълти, а в крайния стадий листата придобиват жълтеникаво бял цвят. Младите листа в

		горната част на растението поддържат зелен, но блед цвят и са склонни да намалят размера си. Този дефицит намалява разклоненията, което води до къси и тънки растения. При предоставяне на нужния азот, възстановяването на дефицитните растения е незабавно (дни) и забележително.
Фосфор	P	Симптомите при фосфорен дефицит не са много отличителни и поради тази причина трудно се идентифицират. Важен отличителен белег е слабия и бавен растеж , спрямо култури отглеждани при сходни условия на околната среда, но без липси (Sen A., 2011). При този дефицит малките листа придобиват червеникаво-лилав оттенък, ръбовете на листата могат да изглеждат изгорени (маргинална некроза), а старите листа стават почти черни. Наблюдава се намалено производство на плодове или семена.
Калий	K	Характерни симптоми на този дефицит са изгаряния по ръбовете на листата (маргинална некроза) и в по-напреднал етап се наблюдава пожълтявания (междужилкова хлороза) и изгаряния (некроза) между жилките на листата.
Калций	Ca	Наблюдава се некроза около основата на листата. Симптомите на некротична тъкан в бързо растящи области се дължи на калциев дефицит и се наблюдават изгаряния по върховете на марулите и вързиво гниена при домати.

Магнезий	Mg	Признак за магнезиев дефицит светло зелено или жълто оцветяване на ръбовете на долните и по-старите листа. Също така могат да се появят и други цветове като кафяво, лилаво и червено. Самото растение извлича магнезият от старите листа, за да го достави на по-новите.
Желязо	Fe	Хлороза между жилките на листата.
Бор	B	Окапване на връхните пъпки, корените и листата.
Манган	Mn	Просветляване на листата от недостатъчно хлорофил, пожълтяване на листата и почервяване между жилките и по краищата.

Методи за измерване азот в растенията

В началните фази на развитието си растенията се нуждаят от много малко азот, но той им е крайно необходим, за да не се получи задържане в растежа на листата и корените, с което да се ограничи усвояването на други хранителни вещества. Най-голяма нужда от азотна храна растенията имат в периода на интензивното нарастване, а също така при залагане на репродуктивните органи. През време на узряването листата започват да стареят и натрупаните в тях вещества се придвижват към репродуктивните органи. Прекомерното и едностранчиво азотно торене усилва растежа, листата нарастват и придобиват тъмозелен цвят, вегетационният период се удължава и узряването се забавя. Поради натрупването на повече белтъчини в растенията се задържа голямо количество вода и тъканите стават меки и сочни. Такива растения са предразположени към гъбни болести, по-силно се нападат от вредители, а житните зърнени култури полягат.

При недостиг на азот листата изменят цвета си от бледозелен до жълто-зелен и жълт. Пожълтелите листа по-късно придобиват оранжев или кафяв оттенък и преждевременно окапват. Растенията остават дребни, със слаба фотосинтетична дейност и се вдървесняват, в резултат на което не се осигурява висок и качествен добив на зелена маса, плодове и семена.

Количеството азот в растенията може да бъде измерено, чрез няколко метода, а измерване на вегетационен индекс, SPAD метър именно химически тест, и диаграма на цветните листа. Всеки метод има своя предимства и недостатъци.

- За определяне на количеството азот в растението се използва измерването на вегетационният индекс NDVI (Normalized Difference Vegetation Index). Това е цифров индикатор, при който се използват вълни бизло до инфрачервеният спектър. Характерно за него, е че светлината която растението поглъща или абсорбира непряко служи за измерване на количеството азот в растението (Jaroenseng S., 2010). Този метод (NDVI) може да се получи от сателитно изображение, но не е подходящо за определяне на азот в малка площ.

Този индекс се основава на наблюдението, че различните повърхности отразяват различно различните видове светлина.

Чрез съотношението на червените и близките инфрачервени ленти от отдалечено изображение, може да се определи индекс на растителност "greenness". Индексът на нормализираната разлика в растителността (NDVI) е може би най-честият от тези индекси на

съотношението растителност. NDVI се изчислява на базата на пиксел като нормализирана разлика между червените и близките инфрачервени ленти от изображението:

$$NDVI = \frac{(NIR-RED)}{(NIR+RED)}, \quad (1.1)$$

където NIR е стойността на близката инфрачервена лента за клетка и RED е стойността на червената лента за клетката. NDVI може да се изчисли за всяко изображение, което има червена и близка инфрачервена лента. Биофизичната интерпретация на NDVI е част от абсорбираната фотосинтетично активна радиация.

- SPAD метър

Това е устройство за измерване а количеството хлорофил в растението. Хлорофилът е зелен пигмент, който позволява на растенията да фотосинтезират. Този процес използва слънчевата светлина, за да превърне въглеродния диоксид и водата в градивните елементи на растенията. Тъй като азотът е част от хлорофила, чрез измерване на хлорофил може индиректно да се измери количеството азот в растението. Това позволява по-ефективно планиране на приложенията на торове.

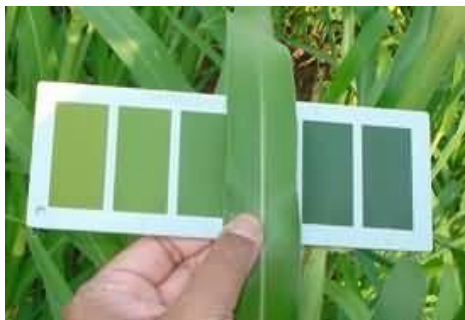


Фиг.2. SPAD-метър

Измервателният уред SPAD измерва разликата между пропускателната способност на червената (650 nm) и инфрачервената (940 nm) светлина през листата, генерирайки трицифрена стойност на SPAD. Измерванията с този уред могат да бъдат силно повлияни от специфичната част на листата, където се правят измерванията, тъй като хлорофилът не е равномерно разпределен по листовото острие.

- Графична диаграма на листа

Графичната диаграма на листата представлява цветна диаграма с много оттенъци на цвета на самото листото от светлозелено до тъмнозелено. Този метод е характерен с това че пробният лист се срязва и се сравнява с цвета от цветната диаграма. Графиката ще даде възможност за определяне на количеството азот в това листо. При различните видове култури се използват различни диаграми.



Фиг.3. Диаграма за определяне на количеството азот в растението.

- Анализ на цветни изображения на листо на растение

Чрез анзлиз на цветни изображения може да се определи недостигът и излишъкът на азот в растението. За обработка на изображението най-често се използват RGB и HIS модели. Цветните компоненти се извличат от предварително обработените изображения, които включват R, G, B компонентите на RGB модела и H, I компонентата на HIS модела.

ИЗВОДИ

Направен е обзор на методите за определяне наличие или липса на химични елементи в растения.

1. Методите чрез директно измерване на определените показатели могат да се използват само за конкретни химични елементи. Цената им е висока и има наличие на смущения. Някои от тези методи са субективни.

2. Цифровата обработка на изображения позволява използването на много по-сложни алгоритми и следователно може да се приложи за определяне на наличие или дефицит на няколко химични елемента едновременно. Тази група методи са широко приложими през последните години за оценяване на различни качествени показатели.

3. Методи чрез спектрален анализ. Методите от тази група са съвременни, сравнително висока точност се постига с тях, но уредите за спектрален анализ са предимно лабораторни (стационарни) и имат сравнително висока цена.

БЛАГОДАРНОСТ

Изследванията са подкрепени по договор с № 2018-ЕЕА-03 „Изследване и моделиране на процесите отглеждане, преработване и окачествяване на хранителни продукти“, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

Jaroenseng S., (2010). Results of Studies the Possible Correlations between SPAD Value and Total, Journal of ISSAAS, 16(1), 123-161

Sen A., (2011). Leaf Colour Chart vis-a-vis Nitrogen Management, American Journal of Plant Sciences in Different Rice Genotypes , 2, 223-236

Yuan Z., Cao Q., Zhang K., Tahir Ata-Ul-Karim S., Tian Y., Zhu Y., Cao W., Liu X., (2016). Optimal Leaf Positions for SPAD Meter Measurement in Rice

SOFTWARE APPLICATIONS FOR SIMULATION OF WIRELESS SENSOR NETWORKS⁴

Teodora Ivanova - student

Department of Automatics and Mechatronics,
University of Ruse "Angel Kanchev"

Assoc. Prof. Tsvetelina Georgieva, PhD

Department of Automatics and Mechatronics,
University of Ruse "Angel Kanchev"

Tel.: +359 82 888 668

E-mail: cgeorgieva@uni-ruse.bg

Abstract: *The paper presents the types of wireless sensor networks and their interfaces with application in precise agriculture. Wireless sensor networks are used for remote data transmission and wireless communication. The report is in the field of automation. It covers all aspects in the field of application, graphical interface in the field of wireless communication. The report is a study of the wireless sensor network.*

Keywords: *wireless sensor network*

ВЪВЕДЕНИЕ

Безжичната сензорна мрежа се състои от разпределени в пространството автономни, безжични сензорни платформи.

Използват се за измерване на различни физични величини като температура, влажност на въздуха, инфрачервена светлина, вибрации, налягане, химически сензори и други. Безжичната им комуникация и малкият им размер позволяват покриване на големи площи като гори, мостове и сгради.

Малък размер, ниска цена и безжична мрежова функционалност.

Те включват в себе си един или няколко сензора, приемопредавател, даващ възможност за безжична комуникация, и управляващ изчислителен блок.

ИЗЛОЖЕНИЕ

Симуляционно изследване на безжична сензорна мрежа

Безжичните сензорни мрежи намират огромно приложение в интелигентни сгради, регионно наблюдение, откриване и сигнализиране за горски пожари.

Интелигентни сгради

Множество сгради имат огромно количество разсип на енергия поради своето неефективно отопление, вентилация и климатизация (ОВК). Многовъзловите безжични сензорни мрежи позволяват наблюдението и контролирането на тези величини в реално време и с висока резолюция, което води до енергийно ефикасни помещения. Освен това, решенията базирани на тези безжични мрежи изключват скъпо и много често неудобно окабеляване на тези помещения.

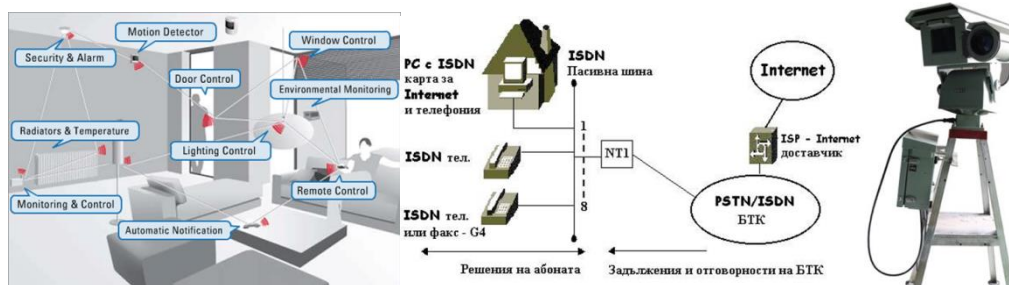
Регионно наблюдение

Множество сензори се монтират в регион, където определено явление ще се наблюдава. Типичен пример е използването на безжична сензорна мрежа за наблюдаване на военно вражеско нашествие в определен регион.

⁴ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: ПРОГРАМНИ СРЕДСТВА ЗА СИМУЛАЦИЯ НА БЕЗЖИЧНИ СЕНЗОРНИ МРЕЖИ.

Откриване и сигнализиране на горски пожари

Множество сензорни възли се монтират в гора, за да може своевременно да се засече избухването на пожар, да се локализира и съответно да бъдат сигнализирани съответните власти. За разкриването на самия пожар, сензорните възли използват температурни сензори, сензори за влажност и сензори за газове, които се създават от огъня в гората.



Фиг. 1. Приложение на безжичните сензорни мрежи

На разположение има няколко комуникационни интерфейса.

Интерфейс на базата на RS-485

Служи за събиране на данни и управление, позволяващ връзката на множество устройства по обща шина. Предаването на данни е чрез диференциални сигнали по две линии. Дава се възможност да се свързват по обща шина множество приемо-предаватели. Могат да се свържат до 32 устройства към една обща шина.

CAN интерфейс

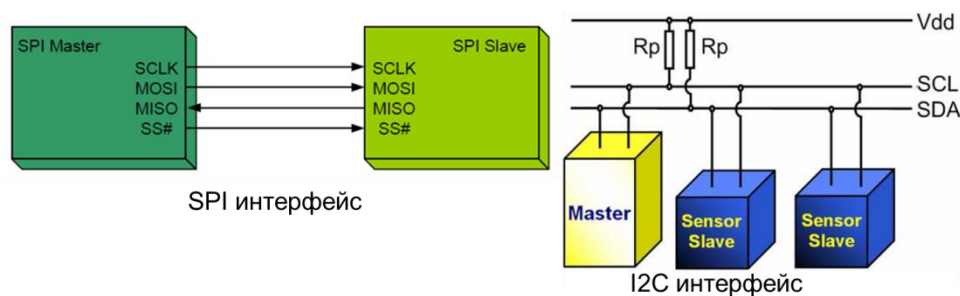
Намира приложение в автомобилостроене и автоматизация на производството. Има протокол за обмен на данни. Устройствата се свързват към обща шина (bus) протокол от по-високо ниво CANopen.



Фиг.2. Устройства за свързване

SPI и I2C интерфейси за сензори с интегрално изпълнение

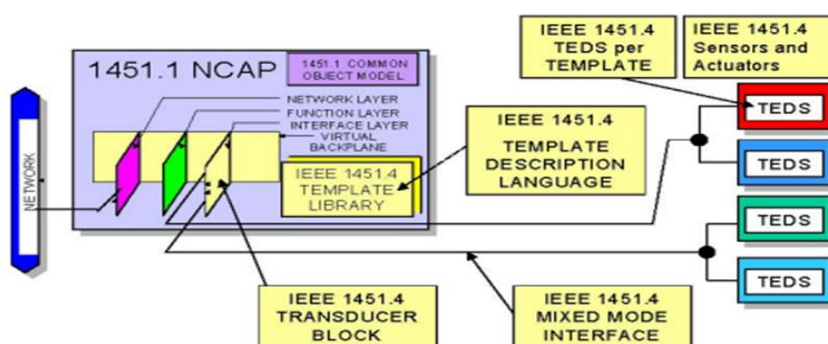
Има едно главно (master) устройство и свързани с него едно или повече подчинени (slave) устройства. При SPI интерфейса комуникацията се осъществява посредством четири линии. I2C интерфейсът използва две линии за сериен обмен на информация – SDA и SCL.



Фиг. 3. SPI и I2C интерфейси

IEEE1451.4

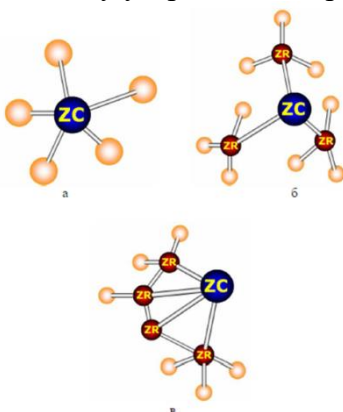
За описване начина на свързване на интелигентни сензори (Шопов, 2010). Дава се възможност сензорът да съхранява информация за самия него и запазва аналоговия сигнал от сензора.



Фиг.4. IEEE1451.4

ZigBee

Като надстройка на стандарта IEEE 802.15.4 (Shopov M., 2012). Опростява се разработката на безжични устройства. Възможност за комуникация между продукти на различни производители. Стандартът IEEE 802.15.4 специфицира физическия и каналния слоеве за множество радио-честотни ленти, включително 868MHz, 915MHz и 2,4GHz. Маршрутизира съобщения от съседните му устройства в мрежата към други устройства.



Фиг.5. Мрежова архитектура

Acrylic Wi-Fi (Windows)

Acrylic Wi-Fi Home е Wi-Fi скенер, който е опростена и безплатна версия на професионалния Acrylic Wi-Fi. Интерфейса е разбираем и показва списък от SSID-та заедно с детайлите им. Друг тип информация, която дава са dBm стойностите (сила на сигнала изразена от 0 до -100, като "0" е възможно най-добрият сигнал) за RSSI (Received Signal Strength Indication), може да именува 802.11 стандарта (включително 802.11ac), разпознава по-големи

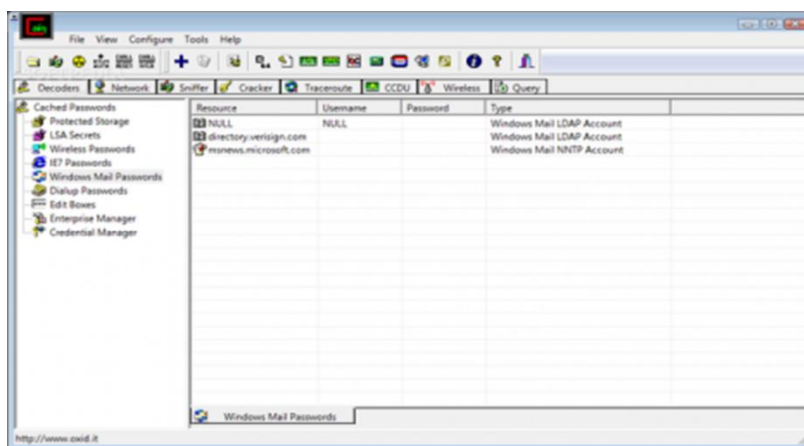
мрежи и идентифицира различните канали. Тъй като е безплатна версия, не може да разкрива скритите SSID-та, но показва какво знае мрежата за тях.



Фиг.6. Acrylic Wi-Fi Home е Wi-Fi скенер

Can & Abel (Windows)

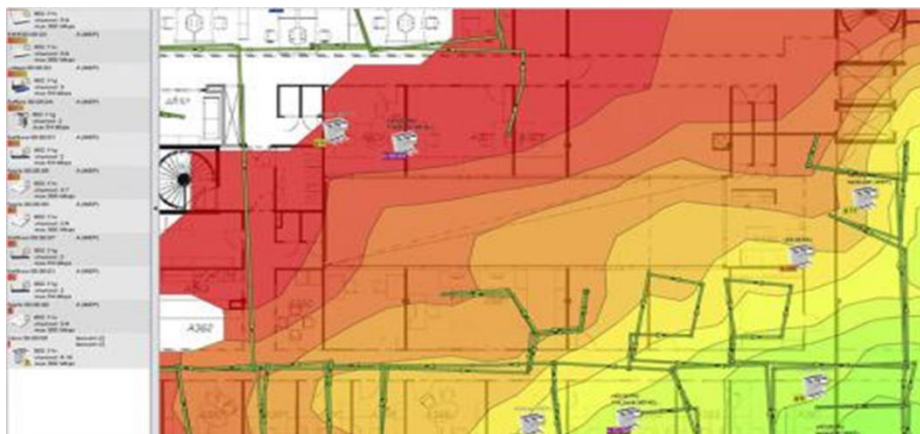
Този инструмент е с широкоспектърна насоченост. От една страна възстановява пароли, от друга има способност да краква пароли. Също така може съвсем успешно да анализира Wi-Fi мрежа. С идеята да е възможно най-опростен, софтуера не предоставя графики и таблици. Не може да засича 802.11ac стандарта, както и не анализира каналите.



Фиг.7. Can & Abel (Windows)

EkahauHeatMapper (Windows)

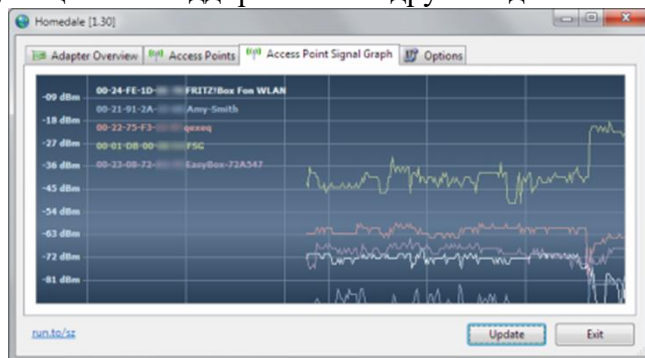
Както подсказва името, това е софтуер за картографиране на мрежи. HeatMapper всъщност, е безплатната версия на Wi-Fi Site and Survey Planner на Ekahau. Предоставя атрактивна графика на покритието и дава информация за настройките по сигурността на засечените Wi-Fi мрежи. Спокойно анализира a/b/g/n мрежи, но има известни затруднения с 802.11ac.



Фиг.8. EkahauHeatMapper (Windows)

Homedale (Windows)

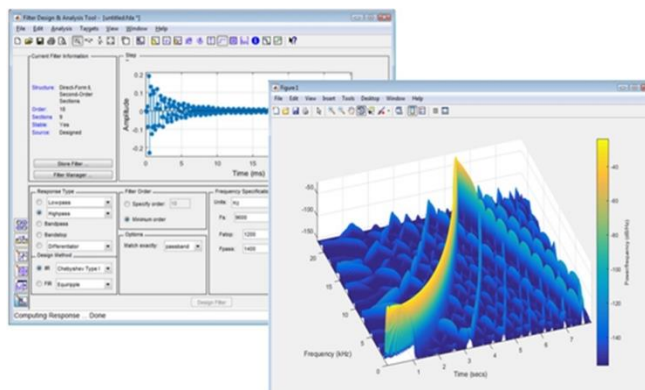
Homedale е относително прости чък и портативен Windows-базиран скенер с възможност за включване на интерфейс с команден ред. Освен, че дава данни за сила на сигнала и основна информация за мрежата, също така поддържа GPS и други видове геозасичане.



Фиг.9. Homedale (Windows)

MATLAB

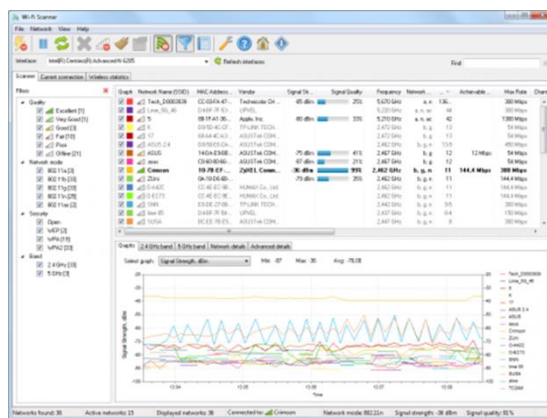
Графични потребителски интерфейси за симулация на невронни мрежи. Позволяват вграждането на множество допълнителни функции, които да реализират параметри на безжичните сензорни мрежи за различните интерфейси за комуникация. Позволяват определяне на разположението на модулите върху изследваната площ, както и определяне на състоянието на пакетите, които се предават по мрежата.



Фиг.10. Графични потребителски интерфейси за симулация на невронни мрежи в Matlab

LizardSystems Wi-Fi Scanner (Windows)

LizardSystems предлага безплатна версия на техния Wi-Fi Scanner, ако не е за комерсиална употреба. Освен, че поддържа всички версии на 802.11 стандарта, разкрива и множеството канали.



Фиг. 11. LizardSystems Wi-Fi Scanner

ЗАКЛЮЧЕНИЕ

Безжичните сензорни системи намират все по-голямо приложение в ежедневието на човека. Цифровите им интерфейси спомагат за по-добра комуникация.

Безжичната комуникация дава възможност за разработване на малки и евтини сензори.

Посочените в презентацията софтуерни инструменти предоставят базови данни за безжичните мрежи: сила на сигнала, канали, MAC адреси и статус на сигурността.

Някои от тях показват ниво на шума, или споделят статистика свързана с успешно/неуспешно прехвърлените пакети по мрежата.

За конкретна приложна задача е необходимо създаване на софтуерен инструмент, който да позволява допълнително да се вграждат модули, които да изследват допълнителни параметри на мрежата. За целта най-подходяща е средата MATLAB.

БЛАГОДАРНОСТ

Изследванията са подкрепени по договор с № 2018-БЕА-03 „Изследване и моделиране на процесите отглеждане, преработване и окачествяване на хранителни продукти“, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

Shopov M., Raghunathan V., Schurgers C., Park S., Srivastava S., (2012). Energy Aware Wireless Sensor Networks.ACM Transactions on Embedded Computing Systems

Шопов, М., Н. Каканакон, Г. Петрова, Г. Спасов, (2010). Анализ на методите за вход/изход и протоколи за обмен в мрежи от био-медицински сензори и персонални монитори, Научни трудове на СУБ – Пловдив, Серия Б: Естествени и хуманитарни науки, 12, 114-119

DEVELOPMENT OF MEASURING SYSTEM PULSE AND BLOOD PRESSURE OF THE HUMAN BODY⁵

Svetoslav Petrov - student

Department of Automatics and Mechatronics,
University of Ruse "Angel Kanchev"

Assoc. Prof. Tsvetelina Georgieva, PhD

Department of Automatics and Mechatronics,
University of Ruse "Angel Kanchev"

Tel.: +359 82 888 668

E-mail: cgeorgieva@uni-ruse.bg

Abstract: *Biometric sensors are becoming increasingly widely used in various industries. The deployment of wireless biometric sensors to measure pulse and blood pressure in the lymph node is an excellent opportunity to monitor the health of patients in real time.*

Keywords: *Biometric sensors*

ВЪВЕДЕНИЕ

Факт е, че биометричните технологии вече са неизменна част от нашето ежедневие. Те са модерно и надеждно решение на комплексни проблеми, осигурявайки високо ниво на сигурност, управление на достъпа до сгради и отделни помещения със специален режим, контрол на работното време, достъп до работа с апарати, компютри и съоръжения, достъп до информация, трансакции или друг вид ресурс. Новите европейски документи и визи, включително и новите лични карти в България са с вграден електронен чип с биометрична информация.

Биометричните системи навлизат все по-широко на обществено и потребителско ниво. Тъй като биологичните обекти притежават редица уникални особености, те могат да бъдат използвани за тяхната идентификация. Биометричните системи за контрол и управление на достъпа работят на базата на разпознаване на хората именно чрез техните специфични физиологични и поведенчески черти. Физиологичните биометрични характеристики на човека са свързани с особеностите на тялото – пръстови отпечатаци, форма на лицето, ДНК, особености на ириса или ретината на окото, геометрия на ръката и линии на дланта.

ИЗЛОЖЕНИЕ

Биометрията комбинира методи за уникалното разпознаване на хората базирано на присъщите им физически или поведенчески черти. В информационните технологии в частност, биометрията се използва за идентифициране на хората като контрол за достъп. Биометрията се използва също за следене на индивиди които са под полицейско наблюдение.

Проектиране на система за измерване на биометрични показатели.

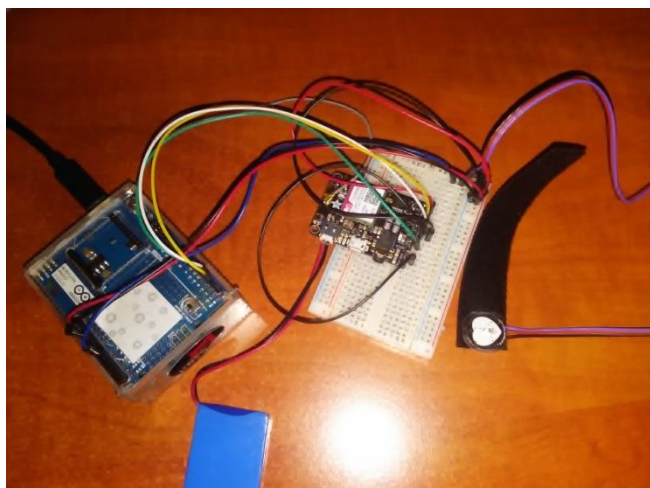
За проектирането на системата са използвани микроконтролер Arduino. Използван е и сензор за пулс и кръвно налягане.

Ардуино е платформа, базирана на входно-изходна платка и среда за програмиране, използваща езика Processing. Платката Ардуино е снабдена с мощни ATmega микроконтролери и обикновено изпълнява ролята на "мозъка" на работи и интерактивни проекти. Ардуино може да "усеща" заобикалящия го свят с помощта на различни сензори (напр. за звук, светлина,

⁵ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: РАЗРАБОТВАНЕ НА СИСТЕМА ЗА ИЗМЕРВАНЕ ПУЛС И КРЪВНО НАЛЯГАНЕ НА ЧОВЕШКОТО ТЯЛО.

докосване, движение и др.) и да реагира на промени в заобикалящата среда (напр. като задвижва електромотори и командва уреди издаващи звук или светлина).

Ардуино Uno (Karvinen K. & Karvinen T., 2011) е микроконтролерна развойна платка с ATmega328P AVR микроконтролер. Има 14 цифрови входно-изходни (I/O) порта, 6 аналогови входа, 16 MHz резонатор, четири светодиода (един потребителски, свързан на 13-ти цифров I/O порт и три, които индикират работата на платката: ON, Tx и Rx), USB конектор, захранващ куплунг, бутон за рестартиране и ICSP конектор. Шест от цифровите I/O порта могат да се използват като PWM (ШИМ) изходи. Свързването с компютър се осъществява чрез USB кабел - USB A to USB B.



Фиг.1. Система за измерване на пулс и кръвно налягане на човешкото тяло.

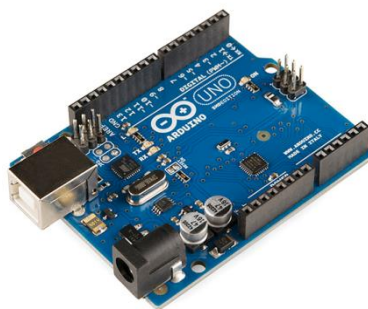
Uno може да се захранва през USB порта на компютъра или от външен източник, като превключването между различните начини за захранване става автоматично. Външният източник на захранване може да е AC-DCадаптер (7 до 12V) или батерия.

В Uno Rev.3 микроконтролера Atmega8 U2 (Melgar E.R. & Diez C.C., 2012), програмиран като USB-to-Serial адаптер, служещ за USB връзката, е сменен с Atmega16U2. След пина Aref са добавени два допълнителни пина за сигналите SDA и SCL (I2C протокола). Бутонът за рестартиране е преместен до USB конектора, а след пина reset има нов порт IOREF.

Техническите характеристики са следните:

- Микроконтролер: ATmega328P
- Работно напрежение: 5 V
- Захранващо напрежение (препоръчително): 7-12 V
- Цифрови I / O порта: 14 (от които 6 могат да са PWM изходи)
- Аналогови входове: 6
- Максимален ток на I / O порт: 40 mA
- Програмируема памет: 32 KB, от които 0.5 KB заети от буутлоудъра
- SRAM: 2 KB
- EEPROM: 1 KB
- Тактова честота: 16 MHz.

С помощта на Ардуино може много лесно да се навлезе в света на микроконтролерите. Програмирането му става с безплатния java базиран софтуер ArduinoIDE, който е изключително удобен и лесен за използване. Ардуино има голяма обществена подкрепа в цял свят и обширен набор от библиотеки и примери за използването на платформата.



Фиг.2. Микроконтролер Ардуино UNO

Ардуино представляват микроконтролерни развойни платки с отворен код, базирани на принципа „лесен за използване“ хардуер и софтуер. Те взаимодействат с външното им обкръжение чрез различни сензори, бутони, електромотори, светодиоди и други, което позволява на разработчиците да програмират създават широк набор от приложения. Голямо предимство на Ардуино е, че конекторите им са стандартни (като например USB), което позволява лесно свързване с други устройства и системи. Важно от финансова гледна точка е да отбележим, че тези електронни платки могат както да се закупят готови, така и да бъдат сглобени от потребители с достатъчно познания в тази област. Серията Ардуино включва микроконтролери и процесори на различни производители – Atmel, ARM, Intel.

Управлението на платките става посредством набор от инструкции на програмния език за програмиране Arduino (базиран на Wiring) и средата за разработка Arduino Software (базирана на Processing). Ардуино възниква като платформа за разработка на проекти от студенти без особен опит в областта на електрониката и програмирането, но впоследствие, след като достига по-широк кръг от потребители и набира голяма популярност, тя започва да се използва в много по-сложни проекти в сфери като IoT, 3D принтиране, вградени системи и др. През годините разработчици от цял свят (студенти, любители, артисти, програмисти, специалисти в различни области) допринасят за натрупването на огромен обем информация и проекти, които са със свободен достъп и помагат в работата, както на хора, които тепърва навлизат в света на Ардуино, така и на експерти.

Данните за сърдечния ритъм може да са изключително важни. На фиг.3. е представен оптичен сензор за измерване на сърдечния ритъм, заедно със схема за усилване и премахване на шума от сигнала. Подходящ за мобилни приложения заради малката си консумация от едва 4mA при 5V захранване.



Фиг.3. Сензор за измерване на пулс и кръвно налягане

Adafruit FONA MiniGSM е GSM модул с който може да се провеждат разговори, да се изпращат текстови съобщения и да се обменя информация в Internet. Ядрото на платката е най-новият SIM800 GSM модул, който е с размер на пощенска марка.

Основни характеристики

- Четири бандов 850/900/1800/1900MHz – може да работи с всяка GSM мрежа и всяка 2G SIM карта
- Провеждане на гласови повиквания. Може да използвате слушалки с микрофон или външен 8ohm говорител + електрен (кондензаторен) микрофон
- Изпращане и получаване на SMS съобщения
- Изпращане и получаване на GPRS данни (TCP/IP, HTTP)
- Сканиране и възпроизвеждане на FM радио (забавна опция, която работи отлично)
- ШИМ драйвер за вибрационен мотор
- Интерфейс с AT команди и автоматично разпознаване на baud rate.



Фиг.4. FONA MiniGSM модул

Програмен код за измерване на биометрични показатели.

За програмиране и настройка на основните параметри на модула е използвана софтуерната платформа Arduino IDE. Тя представлява интегрирана развойна среда, предназначена за лесна връзка и комуникация с контролерите. Поддържа програмните езици C и C++, като също така разполага и с вграден компилатор. За базовата станция е разработена програма, която да осигурява събирането на информация от безжичен сензорен модул и визуализиране на стойностите.

Предимството на повечето Ардуино платформи е, че имат светлинен индикатор и специфичен резистор, свързани между (pin 13 и земя), което е доста удобно за прости тестове. Горният код не може да бъде видян от стандартен C++ компилатор като валидна програма, затова, когато потребителят кликне бутона „Upload to I/O board“ в средата за разработка, копие на кода се записва във временен файл, в който се добавя функцията main(), за да се валидира кода. Средата за разработка на Ардуино използва GNU toolchain и AVR Libc, за да компилира програми и използва Avrdude, за да качва програмите на устройството. Вследствие на това, че Ардуино използва Atmel микроконтролери, средата за разработка на Atmel AVR Studio може също да се използва за писане на софтуер за Ардуино.

Ардуино (Wheat D., 2011) е свободен хардуер: хардуерната референция на проекта са разпространени под Creative Commons Attribution Share-Alike 2.5 лиценз и са достъпни на сайта на Ардуино. Планове и производствени досиета за някои версии на Ардуино са също достъпни. Изходният код за интегрираната среда за разработка (IDE) е достъпен и издаден под GNU General Public License, версия 2.



Фиг.5. Софтуерна среда Ардуино

Въпреки че хардуерните и софтуерните скици са достъпни безплатно под copyleft лицензи, разработчиците са поискали името „Ардуино“ да бъде уникално само за официалния продукт и да не се използва за производни работи без разрешение. Официалният политически документ по използването на Ардуино името гласи, че проектът е отворен да обединява работата на други в официалния продукт.

ЗАКЛЮЧЕНИЕ

Проектирана е и е разработена система за измерване пулса и кръвното налягане на човешкото тяло. Избрани са подходящ контролер, сензор и модул за предаване на данните. Разработено е и е тествано програмното осигуряване на системата. Предстои да се разработи web базираното приложение.

БЛАГОДАРНОСТ

Изследванията са подкрепени по договор с № 2018-РУ-09, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

- Karvinen K., Karvinen T., (2011). Arduino Bots and Gadgets, Published by O'Reilly Media, 250-300
- Melgar E.R., Diez C.C., (2012). Arduino and kinect projects, Springer Science+Business Media, 450-500
- Wheat D., (2011). Arduino Internals, Springer Science+Business Media, LLC, 342-440

QUALITY EVALUATION OF DRIED APRICOTS⁶

Darinka Ivanova Ilieva-Stefanova – PhD Student / Young Scientist

Department of Automatics and Mechatronics,

University of Ruse “Angel Kanchev”

Tel.: +359896124791

E-mail: distefanova@uni-ruse.bg

Assoc. Prof. Donka Ivanova, PhD

Department of Automatics and Mechatronics,

University of Ruse “Angel Kanchev”

Tel.: +359 82 888 266

E-mail: divanova@uni-ruse.bg

Assist. Prof. Nikolay Valov, PhD

Department of Automatics and Mechatronics,

University of Ruse “Angel Kanchev”

Tel.: +359 82 888 668

E-mail: npvalov@uni-ruse.bg

Abstract: *The etiology of the drying process is very varied and geared towards a comprehensive assessment of the main indicators for the quality of the dried fruits and vegetables. The article explains the various methods and technical devices for evaluation of the physical, chemical and biological indicators of quality in the process of dehydration.*

Key words: *apricots, drying temperature, unified drying, food drying process, indicators for quality.*

ВЪВЕДЕНИЕ

За оценката на качеството на сушените продукти, цветът и свиването са най-важните аспекти, тъй като те играят важна роля при приемането на продукта. Потребителите са склонни да свързват цвят и други визуални свойства с вкус, безопасност, време за съхранение, хранително ниво на удовлетворение (Pedresche et al., 2006). Крайното качество на изсушените продукти значително зависи и от температурата на сушене.

В поддържането на качеството на изсушения продукт са включени много фактори като среда и уред за сушене, процеси на предварителна обработка, но измежду тях температурата на сушене е най-критичният. Много е важно да се контролира скоростта на отстраняване на влагата, за да се постигне добро качество на изсушените материали. В процеса на сушене на храни, ефектът от температурата на сушене върху качествените параметри е описан от много изследователи (Farrise et al., 2008, Orikasa et al., 2008, Diamante, 2010; Elamin и Akoy, 2014). Krokida et al. (2003 г.) проучват влиянието на условията на въздуха за сушене върху поведението на сушенето на картофи, моркови, черен пипер и чесън. Те откриват, че температурата на сушене е най-важният фактор, влияещ върху скоростта на сушене. Babalis et al. (2004) също установяват, че температурата на въздуха за сушене значително влияе върху кинетиката на сушенето по време на сушене на тънък слой смокини. Следователно, може да се заключи, че температурата на сушенето е основният повлияващ фактор за определяне качеството на крайния продукт (Farias and Ratti, 2008; Elamin, 2014).

Процесът на сушене с горещ въздух е един от най-често използваните методи в хранително-вкусовата промишленост за производство на сушени плодове. Основните

⁶ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: КАЧЕСТВЕНА ОЦЕНКА НА СУШЕНИ КАЙСИИ.

недостатъците на метода за сушене с горещ въздух са нежелани физически, структурни, химични, органолептични и хранителни промени по време на сушенето, което влошава качеството на крайните продукти и намалява приемането от страна на крайните потребители (ScalaandCrapiste, 2008; ArslanandOzcan, 2011). За щастие, външния вид на изсушените храни като форма, цвят и текстура са сериозен показател за тези неблагоприятни промени и могат да бъдат използвани като индикатори за оценка на качеството на храните. Освен това цветът на хранителните продукти се оценява от крайните потребители като първокачествен параметър за качество (Hatcheretal., 2004; Abdullahetal., 2004).

ИЗЛОЖЕНИЕ

Целта на конкретното проучване е оптимизиране на конвективното сушене на кайсии по критерии за качеството на крайния продукт, след определяне промяната на цвета и съдържанието на β -каротин и витамин С.

Проби

Бяха закупени пресни кайсии (*Prunusarmeniaca* L.), избрани без дефекти и със сходен размер и цвят. Преди да бъдат изсушени, един килограм кайсии се измива, подсушиха и се разделиха на две половини. Кайсиите бяха предварително третирани чрез накисване за 10 минути в 0,2% разтвор на аскорбинова киселина.

Проучвания при сушене

Експерименталната установка за сушенето се състои от камера с рафтове, вентилатор, нагреватели, компютър, контролер, цифров мащаб, сензори за относителната влажност и температура и анемометър. По време на експериментите температурата и влажността на атмосферния въздух са записани, както и температурата и скоростта на сушилният агент са контролирани. Експериментите се провеждат с кайсии, разположени на рафтовете и се изсушават при температури 50 и 80 ° C и скорости на сушилния 0,5 и 2 m/s. По време на сушенето, масата на кайсии се записва на всеки 15 минути, по електронен път с клас на точност III. Първоначалното съдържание на влага в кайсиите е 86%, като те са изсушени, докато не се достигне състояние, при което няма промяна в масата, а съдържанието на влага е около 16 ÷ 18%.

Анализ на качеството

Набор от експерименти е извършен през лятото на 2016 г., за да се определи промяната на цвета и съдържанието на β -каротин и витамин С при изсушени кайсии.

Цветовите стойности L^* , a^* и b^* на пресните и изсушените кайсии бяха измерени, използвайки камера DFK 31AU03. Общата цветова разлика ΔE е определена както следва:

$$\Delta E = \sqrt{(L_f^* - L_i^*)^2 + (a_f^* - a_i^*)^2 + (b_f^* - b_i^*)^2}$$

където L_f^* , a_f^* и b_f^* са цветовите стойности на свежите проби и L_i^* , a_i^* и b_i^* са цветовите стойности на изсушените проби.

Общото съдържание на β -каротин се определя, като се използва методът на Park (Ahmed, M., Akter, M., & Eun, J. B., 2011). Съдържанието на витамин С се определя съгласно метода на Klein & Perry (Ahmed, M., Akter, M., & Eun, J. B., 2011). Анализът се провежда с тройно измерване на проба от сушени кайсии.

Статистически анализ и оптимизация

Процесите на сушене по план В2, симетричен план на състава с централна точка, са изпълнени през лятото на 2016 г. за целите на качествения анализ. Стойностите на температурата и скоростта са показани в Таблица 1.

Регресионните уравнения са получени в резултат на статистическия анализ със StatSoft пакета на STATISTIKA. Оптимизирането на математическия модел на кривата на изсушаване и оптимизирането на условията на обработка са реализирани с GA и GAMULTIOBJ в MATLAB.

Ефект от сушенето върху показателите за качество

Оценката на промяната в цвета на изсушените кайсии се извършва, като се използват цветните стойности L^* , a^* и b^* . Измерените цветови стойности на пресните и изсушените кайсии са показани в Таблица 1 и са близки до публикуваните от Ihns, R., Diamante, L. M., Savage, G. P., & Vanhanen, L. (2011).

Влиянието на температурата на сухия агент върху цветовата стойност L^* е значимо – при пресни кайсии е 69,0, а за изсушени – от 43,8 при $T=65^\circ\text{C}$ до 30,2 при $T=50^\circ\text{C}$.

Скоростта на сушилния агент беше значима при по-ниски температури на сушене. Стойността a^* се увеличава повече от два пъти с повишаването на температурата на сушилния агент. Ihns et al. твърдят, че стойността a^* намалява с повишаването на температурата, докато Karabulut et al. твърди, че a^* се увеличава с повишаването на температурата на сушилния агент, което потвърждава потъмняването на изсушените кайсии. Стойността b^* се променя в диапазона от 9 при $T=50^\circ\text{C}$ и $V=0.5\text{ m/s}$ до 25.1 при $T=65^\circ\text{C}$ и $V=0.5\text{ m/s}$.

В последната колона на таблица 1. показва общата промяна на цвета ΔE . Очевидно е, че с повишаването на температурата на сухия агент до $T=65^\circ\text{C}$ загубите на цвят в изсушените кайсии намаляват.

Промяната на цвета на сухите кайсии може да бъде описана с математическия модел

$$\Delta E = 283.3708 - 7.165 T + 1.793 v + 0.0526 T^2$$

Моделът е адекватен.

Таблица 1. Цветови стойности, β -каротин и витамин С на сушени кайсии.

	T	v	$T, ^\circ\text{C}$	$v, \text{ m/s}$	$\beta\text{-c.}$	vit.C	L^*	a^*	b^*	ΔE
пресни					41	17	69.0	6.2	51.8	85.50
1	-1	-1	50	0.5	24	5.28	30.2	6	9	57.77
2	-1	0	50	1.25	26	7.2	34.4	9.7	9.6	54.68
3	-1	1	50	2	26	8.54	37.8	10.3	10.2	52.16
4	0	-1	65	0.5	29	5.06	41.2	12.2	25.1	39.01
5	0	0	65	1.25	30	6.05	43.8	12.8	24.8	37.52
6	0	1	65	2	31	6.78	43.4	12.6	24.6	37.90
7	1	-1	80	0.5	35	4.75	35.2	11.9	21	46.08
8	1	0	80	1.25	35	4.4	36.5	12.2	22.2	44.37
9	1	1	80	2	37	4.4	36.4	12.4	21.8	44.73

Съдържанието на β -каротин в кайсии, изсушени при различни температури и скорости на изсушаващия агент, е показано в Таблица 1. Съдържанието на β -каротин в пресните плодове е 41 mg/100 g, което е по-голямо от стойностите за различните сортове, публикувани от Ruiz et al. Стойността на β -каротин при температура $T=50^\circ\text{C}$ и скорост $V=0.5\text{ m/s}$ е 24 mg/100 g, а при $T=80^\circ\text{C}$ и $V=0.5\text{ m/s}$ е 35 mg/100 g, т.е. повишаването на температурата на сушилния агент, увеличава съдържанието на β -каротин. В проучванията на Ihnsi Karabulut също е показано, че съдържанието на β -каротин в изсушените кайсии се увеличава с повишаването на температурата на сушилния агент. Това се потвърждава и от таблица 1., където, макар и в по-

малки граници, нарастването на скоростта на сушилния агент при една и съща температура повишава съдържанието на β -каротин.

Моделът, описващ зависимостта на съдържанието на β -каротин в изсушените кайсии и температурата и скоростта на сушилния агент, е следният:

$$\beta = 15.3332 + 0.0555 T + 1.3333 v + 0.0022 T^2$$

Моделът е адекватен.

Съдържанието на витамин С в пресните кайсии е 17 mg/100g. Витамин С стойностите в сушените при различни температури и скорости кайсии са показани в Таблица 1. Те варират от 4.4 mg/100g до 8.54 mg/100g. Съдържанието на витамин С намалява с повишаването на температурата на сушилния агент. При $T=50^\circ\text{C}$, влиянието на скоростта на сушилния агент е значително – намаляването на скоростта понижава съдържанието на витамин С. При $T=80^\circ\text{C}$ влиянието на скоростта е много слабо.

Моделът на съдържанието на витамин С в изсушени кайсии в зависимост от температурата и скоростта на сушилния агент е:

$$C = 3.6085 + 0.0173 T + 6.2433 v + 0.08022 v T.$$

Моделът е адекватен.

ЗАКЛЮЧЕНИЯ

Беше изследвано въздействието на параметрите на изсушаващия агент върху загубата на качествените показатели за цвят, съдържание на β -каротин и съдържание на витамин С. С увеличаването на температурата на сушилния агент до 65°C загубата на цвят на изсушените кайсии намалява. Увеличаването на температурата на сушилния агент е свързано с увеличаването на съдържанието на β -каротин и с намаляването на съдържанието на витамин С.

Оптималните стойности на температурата и скоростта на сушилния агент според критериите за качество на крайния продукт и продължителността на процеса са $T=63,5^\circ\text{C}$ и $V=0,5\text{ m/s}$.

Подходът за оптимизиране на процеса на сушене на кайсии, представен по-горе, може да бъде приложен при сушенето на различни плодове и зеленчуци.

БЛАГОДАРНОСТ

Изследванията са подкрепени по договор с № 2018-ЕЕА-03 „Изследване и моделиране на процесите отглеждане, преработване и окачествяване на хранителни продукти“, финансиран от Фонд „Научни изследвания“ при Русенски университет „Ангел Кънчев“.

REFERENCES

- Abdullah, M.Z., Guan, L.C., Lim, K.C., Karim, A.A., (2004). The application of computer vision and tomographic radar imaging for assessing physical properties of food. J. Food Eng. 61 (1), 125-135.
- Ahmed, M., Akter, M., & Eun, J. B. (2011). Optimisation of drying conditions for the extraction of β -carotene, phenolic and ascorbic acid content from yellow-fleshed sweet potato using response surface methodology. International Journal of Food Science & Technology, 46(7), 1356-1362.
- Albanese, D., Cinquanta, L., Cuccurullo, G., & DiMatteo, M. (2013). Effects of microwave and hot-air drying methods on colour, β -carotene and radical scavenging activity of apricots. International Journal of Food Science & Technology, 48(6), 1327-1333.
- Arslan, D., Ozcan, M.M., (2011). Dehydration of red bell-pepper (*Capsicum annuum* L.): change in drying behavior, colour and antioxidant content. Food Bioprod. Process. 89 (4), 504-513.

Babalís, S. J., and V.G. Belessiotis. (2004). Influence of the drying conditions on the drying constants and moisture diffusivity during the thin-layer drying of figs. *Journal of Food Engineering*, 2004(65): 449-58.

Diamante, L., M. Durand, G. Savage, and L. Vanhanen. (2010). Effect of temperature on the drying characteristics, colour and ascorbic acid content of green and gold kiwifruits. *International Food Research Journal*, 17: 441-451.

Elamin, O. M., and Akoy. (2014). Effect of drying temperature on some quality attributes of mango slices. *International Journal of Innovation and Scientific Research*, 4(2): 91-99.

Farris, S., S. Gobbi, D. Torreggiani, and L. Piergiovanni. (2008). Assessment of two different rapid compression tests for the evaluation of texture differences in osmo-airdried apple rings. *Journal of Food Engineering*, 88(4): 484-491.

Farias, A.M., and C. Ratti. (2008). Advances in food dehydration. *Contemporary food engineering*. CRC Press, Boca Raton, pp 1-36
 Fluent 6.3 user's Guide 2005. Fluent documentation. Fluent Inc, Lebanon.

Hatcher, D.W., Symons, S.J., Manivannan, U., (2004). Developments in the use of image analysis for the assessment of oriental noodle appearance and color. *J. Food Eng.* 61 (1), 109-117.

Ihns, R., Diamante, L. M., Savage, G. P., & Vanhanen, L. (2011). Effect of temperature on the drying characteristics, colour, antioxidant and beta-carotene contents of two pricot varieties. *International journal of food science & technology*, 46(2), 275-283.

Ivanova D., Valov N., Stojanov V. (2013). Dynamic models for apricots drying using genetic algorithm. *Information, Communication and Control Systems and Technologies*, 1: 29-33.

Karabulut, I., Topcu, A., Duran, A., Turan, S., & Ozturk, B. (2007). Effect of hot air drying and sun drying on color values and β -carotene content of apricot (*Prunus armeniaca* L.). *LWT-Food Science and Technology*, 40(5), 753-758.

Krokida, M. K., V. T. Karathanos, Z. B. Maroulis, and D. K. Marinou. (2003). Drying kinetics of some vegetables. *Journal of Food Engineering*, 59, 391-403.

Oríkasa, T., L. Wu, T. Shiina, and A. Tagawa. (2008). Drying characteristics of kiwifruit during hot air drying. *Journal of Food Engineering*, 85(2): 303-308.

Pedreschi, F., J. Leo'n, D. Mery, and P. Moyano. (2006). Development of a computer vision system to measure the color of potato chips. *Food Research International*, 39, 1092-1098.

Ruiz, D., Egea, J., Tomás-Barberán, F. A., & Gil, M.I. (2005). Carotenoids from new apricot (*Prunus armeniaca* L.) varieties and their relationship with flesh and skin color. *Journal of Agricultural and Food Chemistry*, 53(16), 6368-6374.

Scala, K.D., Crapiste, G., (2008). Drying kinetics and quality changes during drying of red pepper. *LWT-Food Sci. Technol.* 41 (5), 789-795.

GENERATOR FOR ENSURING OF ALTERNATIVE POWER SUPPLY IN BUILDINGS⁷

Bojidar Tasev

Department of Electronics,
“Angel Kanchev” Univesity of Ruse
E-mail:

Daniel Kajtsanov

Department of Electronics,
“Angel Kanchev” Univesity of Ruse
E-mail: office@didilight.com

Seher Kadirova, PhD

Department of Electronics,
“Angel Kanchev” Univesity of Ruse
Phone: +359 82 888 516
E-mail: skadirova@uni-ruse.bg

Abstract: *This paper presents alternative solutions to ensure power supply of critical consumes. A generator controlled by microprocessor capable of taking over power supply in case there are problems with power supply from the national power grid is presented in the paper. A block diagram and an electrical schematic have been presented.*

Keywords: *generator, alternative power supply, microcontroller*

ВЪВЕДЕНИЕ

Генераторът е основен компонент на плановете за подготовка за аварийни ситуации на много хора. Все пак много не успяват да си помислят как точно ще захранват елементите, които искат да работят, когато мрежата е прекъснала. Има три основни типа: автоматичен, предпазен панел за ръчно прехвърляне и блокировка на прекъсвача. Всеки от тях има различни степени на сложност, ползи и разходи.

Алтернативните методи за производство на електроенергия трябва да отговарят на следните технически и икономически съображения:

- достигане на текущата и номинална мощност за много кратко време (между 0,1 до 120 секунди), в зависимост от вида на алтернативния източник.
- могат да бъдат експлоатирани и / или върнати в експлоатация бързо от квалифициран персонал
- да осигури колкото е възможно повече оперативна самостоятелност
- осигуряват стабилност при експлоатация от техническа гледна точка и доставена електроенергия
- имат относително ниски разходи за изпълнение и работа

Аварийните генератори могат да се използват до 200 часа годишно по време на нуждите на пиковите мощности. В замяна на използването на аварийните генератори комуналните услуги ще предоставят гориво и оборудване, за да поддържат напълно генераторите, както и да изградят централизирано и паралелно захранване. Собствениците на сгради ще бъдат сигурни, че имат гарантирана, надеждна и непрекъсваема мощност от добре поддържани и изпитани аварийни генератори V V Mihut, M Panoiu, C Panoiu and I Baciú (2018).

⁷ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: ГЕНЕРАТОР ЗА ОСИГУРЯВАНЕ НА АЛТЕРНАТИВНО ЕЛЕКТРОЗАХРАНВАНЕ В СГРАДИ.

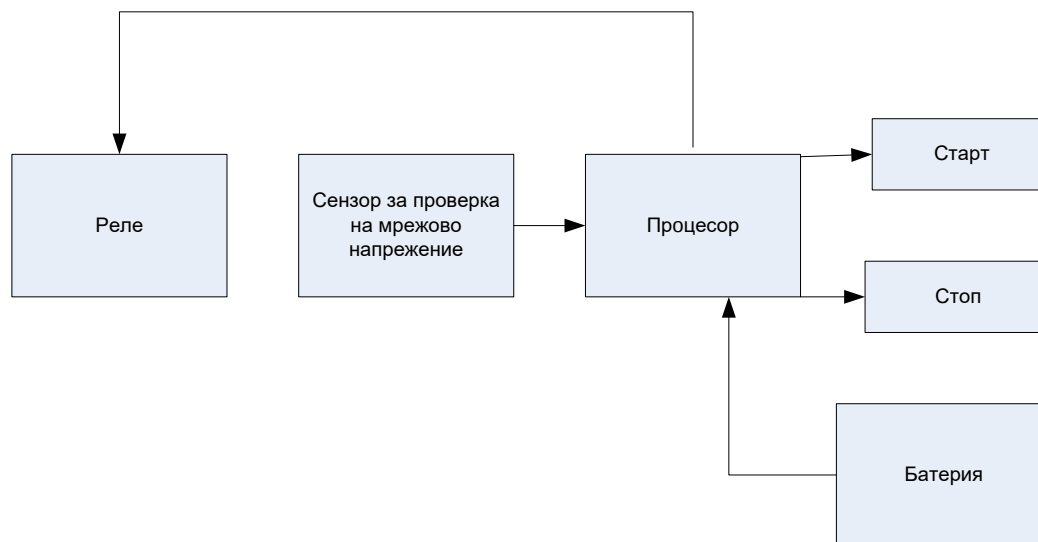
В случай на прекъсване на електрозахранването сградите разчитат на резервно захранване както за безопасността и здравето на населението, така и за защита на важни бизнес активи, които не могат да бъдат загубени при прекъсване на електрозахранването. Неочакваните бедствия често разстройват властта на стотици хиляди хора и фирми. Много сгради, като болници, летища, центрове за данни, съоръжения за водоснабдяване и канализация, бензиностанции и комуникационни и транспортни системи, изискват някаква алтернативна мощност, която в крайна сметка да спасява човешки животи по време на криза. Загубата на енергия за даден бизнес също може да създаде значително икономическо въздействие. Колкото по-дълъг е бизнесът без енергия, толкова по-голяма е икономическата загуба. Когато възникнат тези неочаквани ситуации, резервното захранване осигурява източник за поддръжка на натоварването на оборудването чрез непрекъсваеми източници на захранване, генератори или системи за съхранение на батерии (J, Piper (2010), M, Villalba, (2018), F. Markus, H. and G. Stieva, (2010)).

Една от най-важните задачи по поддръжката, които могат да се извършват с генераторни системи, е редовното упражняване на цялата система. Веднъж всяка седмица или две, в зависимост от размера на генератора и естеството на товарите, които трябва да се доставят, генераторите трябва да се пускат и да работят поне 30 минути. За да се тества цялата система, генераторите трябва да се движат под натоварване - в идеалния случай товарът, който нормално ще захранва. При работа на системата под товар в продължение на 30 минути, генераторът се изпитва заедно със стартовата си система, охладителната система и всички разпределителни устройства, необходими за захранване на товарите.

Целта на публикацията е да се разработят блокова схема и принципна електрическа схема на електронен модул за захранване на сграда.

ИЗЛОЖЕНИЕ

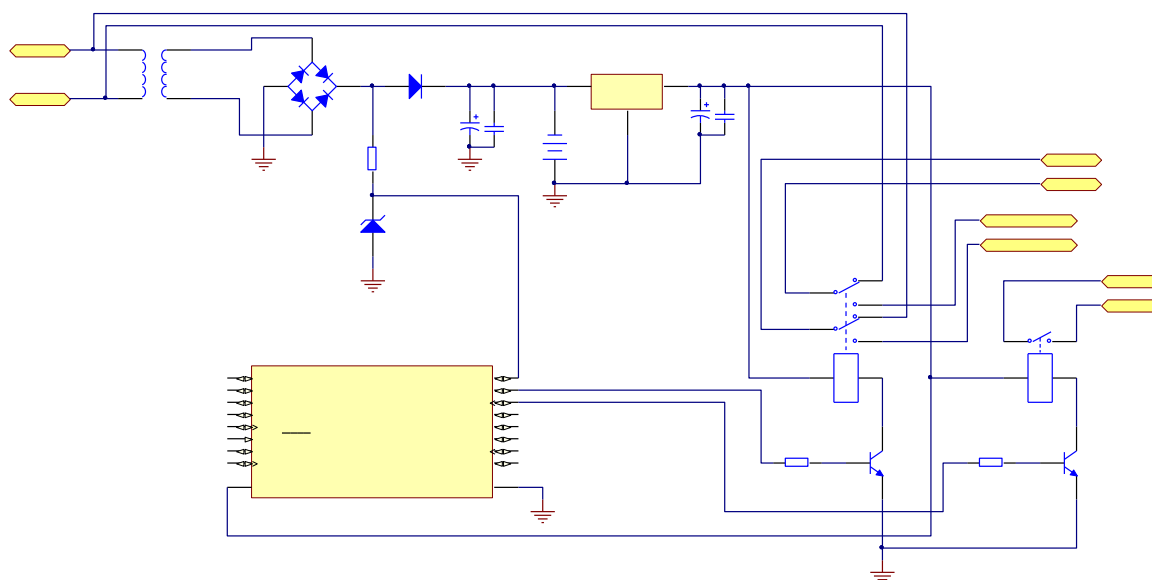
Блокова схема на електронен модул



Фиг. 1. Блокова схема на електронен модул на генератор за осигуряване на резервно захранване на сграда

Сензорът за проверка на мрежовото напрежение следи дали от батерията се подава напрежение към процесора. Следователно, ако към него има подадено напрежение той подава сигнал към релето и съответно то стартира резервното захранване на сградата. При спиране на входното напрежение сензорът за проверка на мрежовото напрежение отчита, че има подадено такова и предава информацията към процесора, който подава сигнал към релето и то спира да работи.

Разработване на принципна електрическа схема



Фиг. 2. Принципна електрическа схема на електронния модул

Чрез делителя се следи входното напрежение $1\text{ k}\Omega$ и ценов диод 5 V . Чрез RB0 на микроконтролера при отпадане на мрежовото напрежение микроконтролера задейства реле K2 и стартира генератора. При вече стартиран генератор, микроконтролера задейства реле K1 и подава напрежението от генератора към консуматора. При поява на входно напрежение микроконтролера детектира наличието на напрежение, при което превключва реле K1 и подава на консуматора входното напрежение и след това изключва реле K2 за СТОП на генератора.

ИЗВОДИ

Предложена е блокова схема на електронен модул за управление на генератор за резервно захранване на сграда.

Синтезирана е принципна електрическа схема на електронния модул и са оразмерени основните електрически елементи.

БЛАГОДАРНОСТ

The present document has been produced with the financial assistance of the Project 2018-RU-07 "Creation of a complex mobile laboratory platform for research and approbation of environmentally friendly technical solutions"

REFERENCES

- Mihut V. V., (2018). Methods for maintaining the continuity of power supply in hospitals, IOP Conference Series: Materials Science and Engineering
- Markus F. Hirschbold and Ginni Stieva, (2010). Automating Emergency Power Supply System Testing in Hospitals, White Paper
- Michael V., (2018). Designing backup, standby, and emergency power for high-performance buildings, FEA Consulting Engineers
- Piper J., (2010), Backup Generators, UPS Systems Provide Power in an Outage, Power & Communication

MAXIMUM POWER POINT TRACKING UNIT FOR BATTERY CHARGING FROM A PHOTOVOLTAIC⁸

Penko Binkov – Student

Department of Electronics,
University of Ruse “Angel Kanchev”
Tel.: +359 878 872 495
E-mail: penko_b@abv.bg

Seher Kadirova, PhD

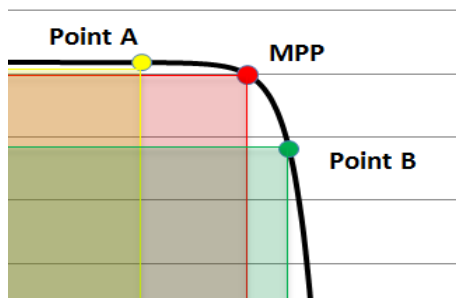
Department of Electronics,
“Angel Kanchev” University of Ruse
Phone: +359 82 888 516
E-mail: skadirova@uni-ruse.bg

Abstract: Solar energy is a potential energy source. A photovoltaic is necessary to harvest this kind of energy, and to be able to gather the most, the PV must have a good efficiency. The maximum efficiency can be achieved when PV works at its maximum power point. Maximum power point tracking controllers play an important role in photovoltaic systems. They maximize the output power of a PV array for a given set of conditions. The aim of the publication is to develop a block diagram and an electrical circuit diagram of the module for tracking the maximum power point for charging a battery from a photovoltaic panel.

Keywords: MPPT, Battery Charging, Photovoltaic

ВЪВЕДЕНИЕ

Стандартните фотоволтаични системи са изградени от фотоволтаични панели, контролер, инвертор, устройство за съхранение на енергия (батерия/батерии). Фотоволтаичните панели преобразуват слънчевата енергия в електричество благодарение на фотоволтаичния ефект. Инверторът служи за преобразуване на постоянното напрежение от панела в променливо напрежение, с параметрите на електричеството в електропреносната мрежа. Контролерът следи и управлява зарядния процес на батериите, така че да осигури максимален живот на батериите. Тяхната ефективност е от порядъка на 25%. За подобряване на ефективността е необходима система за проследяване на максималната точка на мощността. Точката на максимална мощност (ТММ) е точката, в която стойността на произведението на тока и напрежението е най-голямо (Фиг. 1) (R. Putri, S. Wibowo, M. Rifa. (2015), (S. Babaa, M. Armstrong, V. Pickert. (2014), (B. Subudhi, R. Pradhan (2013))).



Фиг. 1. Точка на максимална мощност.

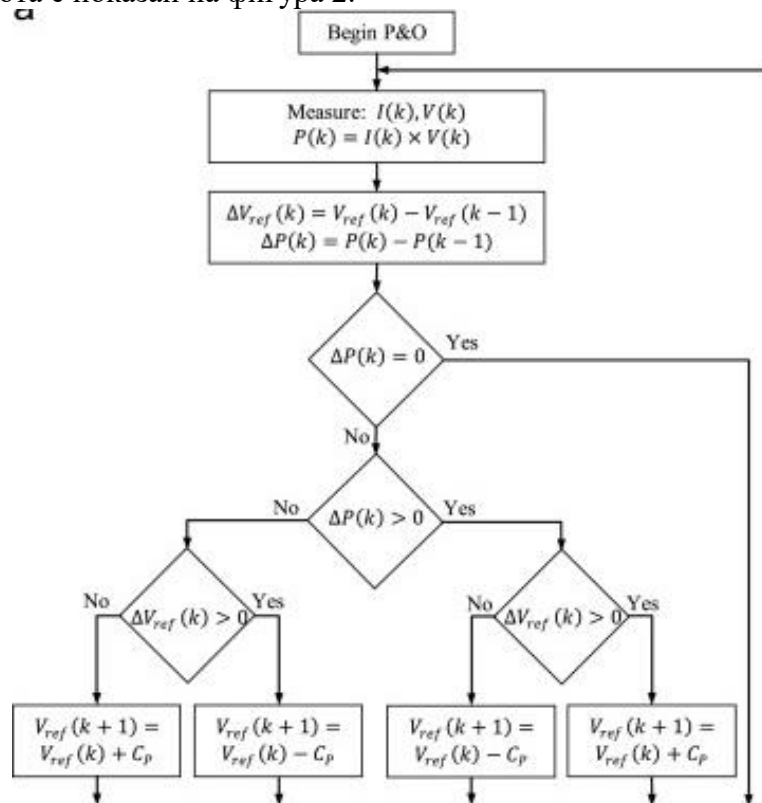
⁸ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: УСТРОЙСТВО ЗА ПРОСЛЕДЯВАНЕ НА МАКСИМАЛНАТА МОЩНОСТ ЗА ЗАРЕЖДАНЕ НА БАТЕРИЯТА ОТ ФОТОВОЛТАИЦИ.

Целта на публикацията е да се разработят блокова схема и принципна електрическа схема на електронен модул за проследяване на точката на максимална мощност при зареждане на батерия от фотоволтаичен панел.

ИЗЛОЖЕНИЕ – ОСНОВНИ МЕТОДИ

Смущение и наблюдение (Perturb and observe)

P&O методът се базира на промяна на напрежението, проверка на изходната мощност и сравняването ѝ с предходната стойност за мощността. Стартира се с измерване на тока и напрежението, и определяне на изходната мощност. Началната стойност на мощността се приема като референтна. При увеличаване на напрежението, ако мощността се е увеличила, това е индикатор за приближаване до ТММ и тази стойност се приема за референтна. Ако има спад в мощността е признак за отдалечаване от ТММ и следва напрежението да се намали. Алгоритъмът на работа е показан на фигура 2.



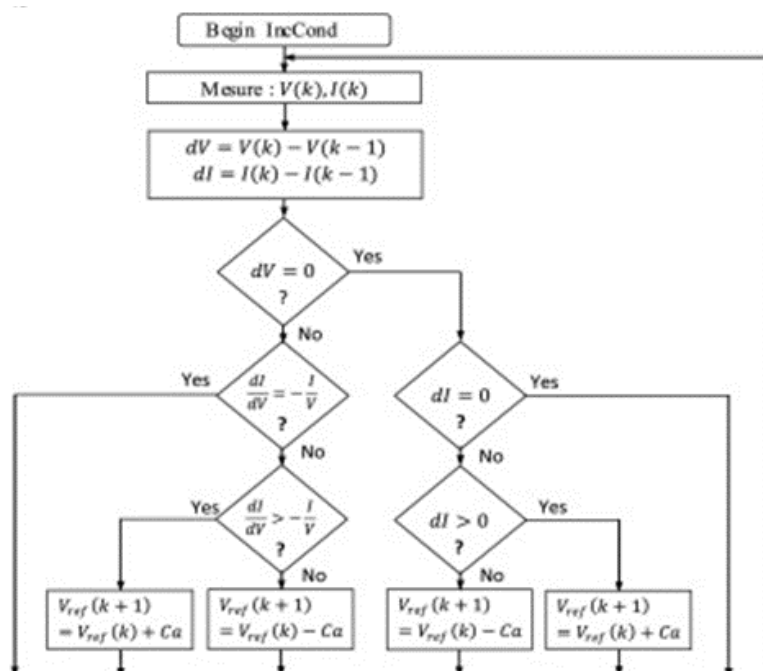
Фиг. 2. Алгоритъм на работа на метода P&O.

Предимства на метода: евтин, лесен за прилагане, широко разпространен, относително добра производителност.

Недостатъци: невъзможност да реагира на краткотрайни резки промени в степента на осветеността, винаги се колебае около точката на максимална мощност, без да я достига.

Нарастваща проводимост (Incremental conductance)

Методът Incremental conductance (IncCond) се основава на това, че наклона на мощностната крива е нула при ТММ в зависимост от напрежението. Изходното напрежение и тока на панела се следят непрекъснато във времето, при което ТММ контролерът изчислява проводимостта и инкременталната проводимост и да взема решение (да увеличи или намали съотношението на изхода). Предназначението на този алгоритъм е да открие работното напрежение на соларния панел, а моментната проводимост е равна на инкременталната проводимост. На фигура 3 е показан алгоритъмът на работа на метода.



Фиг. 3. Алгоритъм на работа на метода Incremental conductance.

Алгоритмът стартира с измерва напрежението и тока на панела. Инкременталните промени на напрежението и тока се изчисляват от микропроцесора.

Щом е достигната ТММ изходната мощност е постоянна, докато се забележи промяна в dI , което е индикация за промяна на метеорологичните условия и промяна в ТММ. Алгоритмът променя напрежението с постоянна стъпка, за да проследи новата ТММ.

За пресмятането на точката на максималната мощност се използват уравненията:

$$\frac{dP}{dV} = 0 \text{ ТММ} \quad (1)$$

$$\frac{dP}{dV} > 0 \text{ ляво от ТММ} \quad (2)$$

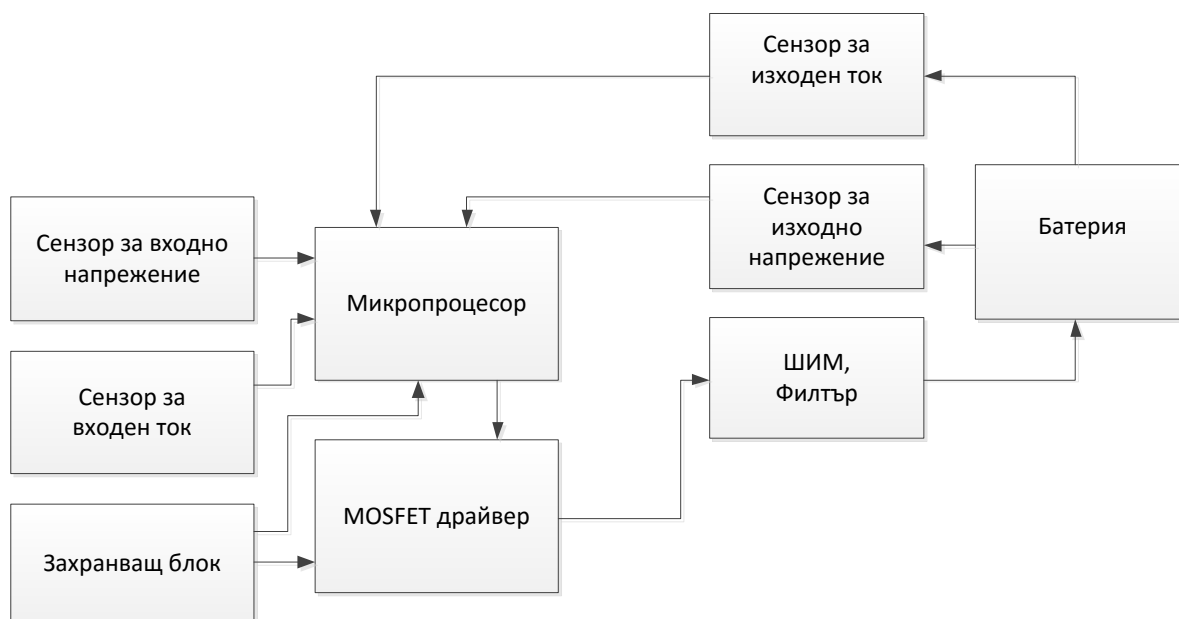
$$\frac{dP}{dV} < 0 \text{ дясно от ТММ} \quad (3)$$

Предимства: по-ефективен от Р&О, не греши в коя посока е ТММ и не се колебае около нея, а определя точно точката.

Недостатъци: по-скъп, по-сложен, изисква по-вече процесорна мощност.

Блокова схема

Той се състои от сензори за входен ток и напрежение и изходен ток и изходно напрежение, захранващ блок, микропроцесор, филтър и широчинно-импулсна модулация. Блоковата схема на устройството е показана на фигура 4.



Фиг. 4. Блок схема на електронен модул за следене на максимална точка на мощността.

Принципна електрическа схема

Захранващият блок е изграден от регулатор на напрежение и филтриращи кондензатори с изходно напрежение от 5V.

Принципната електрическа схема е изградена на основата на микропроцесор PIC16F, към входовете на който се подават сигналите от сензорите за напрежение и ток.

Сензорите за напрежение са делители на напрежение, като измерваното напрежение се получава пропорционално на реалната стойност. Измереното напрежение се подава към входовете на микропроцесора. Напрежението се изразява с уравнението:

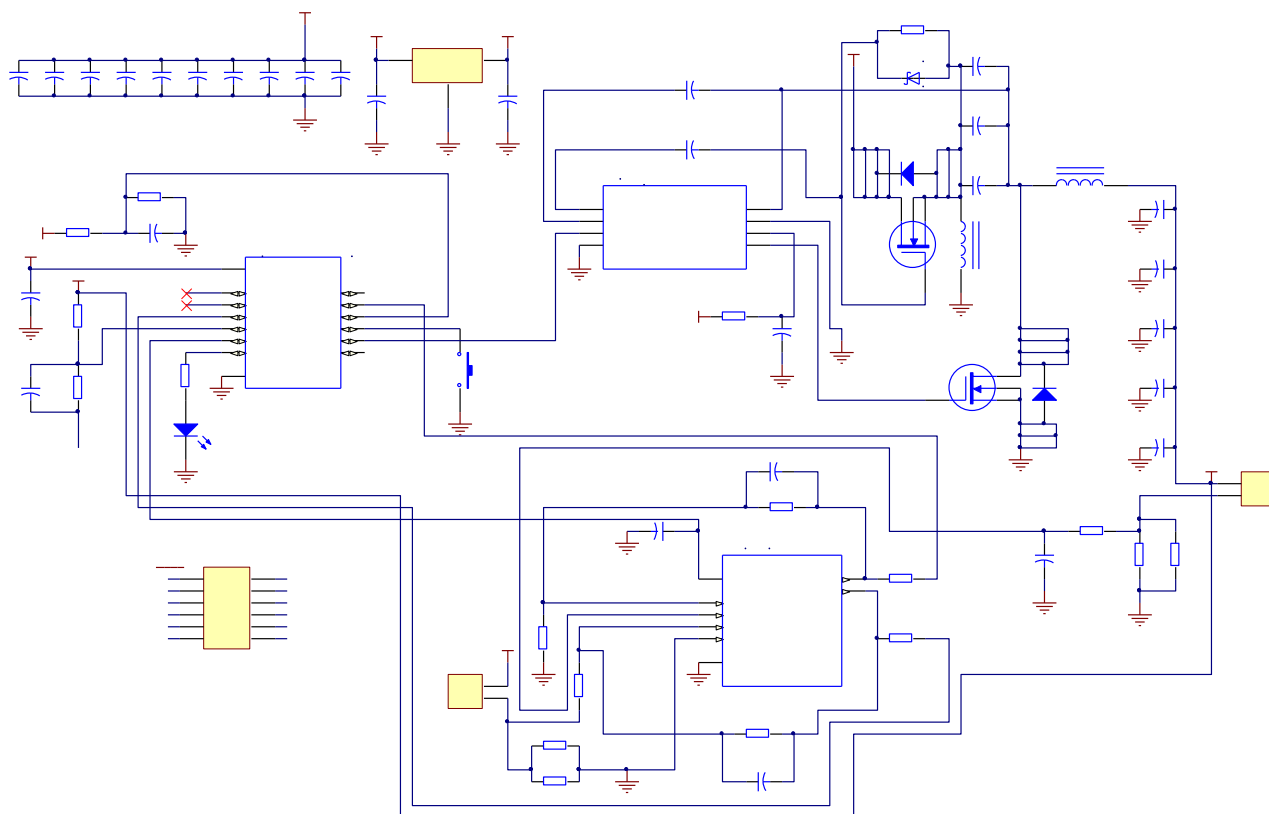
$$\frac{5}{1023} * 5 = 0.0244V \quad (4)$$

Сензорите за ток са изградени от делител на напрежение с нико омен резистор (шунт) и операционен усилвател, тъй като полученият сигнал след делителя е малък се налага усилването му от операционен усилвател. Вече усиленият сигналът се подава към входа на процесора. Максималният ток който може да се измери е 8A. Като на изхода на усилвателя се получават напрежения до 4V.

$$8 * 0.005 * 100 = 4V \quad (5)$$

MOSFET драйверът, който е свързан с процесора и с двата транзистора, е предназначен за реализиране на широчинно-импулсната модулация, с която се регулира изходното напрежение. Транзистор Q1 отговаря за продължителността на високата част на импулса, а транзистор Q2 за продължителността на ниската част на импулса. Филтър, изграден от кондензатори и бобина (LC), като изходът на филтъра е изход на модула.

Принципната електрическа схема на модула е показана на фигура 5.



Фиг. 5. Принципна електрическа схема на електронен модул за следене на максимална точка на мощността.

ЗАКЛЮЧЕНИЕ

Предложена е блокова схема на електронен модул за проследяване на точката на максимална мощност.

Синтезирана е принципна електрическа схема на електронния модул и е обоснован изборът на основните гравивни елементи, за осигуряване на нормални режими на работа на електронния модул.

Подбрани и обосновани са алгоритми на работа на модула за следене на точка на максимална мощност.

БЛАГОДАРНОСТ

The present document has been produced with the financial assistance of the Project 2018-RU-07 "Creation of a complex mobile laboratory platform for research and approbation of environmentally friendly technical solutions"

REFERENCES

- Putri R.I., Wibowo S., Rifa M., (2015), Maximum power point tracking for photovoltaic for incremental conductance method, *Energy Procedia*, 68, 22-30.
- Babaa S.E., Armstrong M., Pickert V., (2014), Overview of maximum power point tracking control methods for PV systems, *Journal of Power and Energy Engineering*, 2, 59-72.
- Subudhi B., Pradhan, R., (2013). A comparative study on maximum power point tracking techniques for photovoltaic power systems., *IEEE Transactions on Sustainable Energy*, 4, 89-98.

IR/R OXYMETER⁹

Aleksandr Furmanov, MsC Student

Department of Intelligent computer system and networks,
Odessa National Polytechnic University
E-mail: skyline_hr_31@outlook.com

Assoc. Prof. Aneliya Manukova, PhD

Department of Electronics,
University of Ruse „Angel Kanchev”
E-mail: amanukova@uni-ruse.bg

Abstract: *The article considers the existing version of the oxymeter, based on the idea of which describes the development of its own device. The article describes both the scientific side and the technical one. The development is based on Arduino microcontroller. The article deals with the development of its own electric circuit for the inclusion of infrared / red diodes, also the article deals with the algorithms of the system operation, software signal processing and time diagrams of the system operation. The report presents both ready-made versions of devices and personal development, and the report also contains publicly available materials from the Internet.*

Key words: *Arduino, oxymeter, system`s algorithm, program component of the system.*

ВЪВЕДЕНИЕ

Кислород свързва се с хемоглобин в еритроцитите при движение през белите дробове. Той се транспортира по цялото тяло във вида артериална кръв. Пулсов оксиметър използва две честоти на светлина (червена и инфрачервена) за определяне процента (%) хемоглобин в кръвта, наситен с кислород. Процентът се нарича наситеност с кислород на кръвта или SpO₂. Пулсов оксиметър също измерва и показва част от пулса, в същото време измерва нивото на SpO₂.

Кислородът в атмосферата влиза в белите дробове по пътя на дишане. Всяко легло съдържа около 300 милиона алвеоли, които са обградени с кръвоносни капиляри. Тъй като алвеоларните стени и капилярните стени са много тънки, кислородът, преминаващ през алвеолите, веднага се пренася в кръвоносните капиляри. (Обикновено при възрастни предаването отнема около 0,25 секунди при покой). Голямата част от кислорода, дифундираща в кръвта, свързва се с хемоглобин в еритроцитите, а част от кислорода се разтваря в плазмата на кръвта. Кръвта, обогатена с кислород (артериална кръв), тече през белодробните вени, след това в лявото предсърдие и левият вентрикул и, накрая, циркулира по всички органи на тялото и техните клетки. Количество кислород, транспортирано около кръга на тялото, се определя в основен смисъл, в каква степен хемоглобинът свързва се с кислород (фактор на белите дробове), концентрацията на хемоглобин (анемичен фактор) и сърдечен изхвърляне (сърдечен фактор).

Наситеност с кислород е показател за пренасянето на кислород в организма и показва, че в организма достига достатъчно количество кислород, особено в белите дробове.

Пулсов оксиметър също може да измерва част от пулса. Обемът на кръвта, изпомпван от сърцето за минута, се нарича сърдечен изход. Честота на качване за една минута се нарича честота на пулса. Тези показатели на сърдечна функция могат да бъдат определени с пулсов оксиметър. Идеята за създаване на собствен прототип на оксиметър е обусловена от идеята за преработка на електрическите схеми и въвеждане на програмна обработка на сигнали вместо

⁹ Докладът е представен на студентската научна сесия на 27.IV.2018 г. в секция „Електротехника, електроника, автоматика“ с оригинално заглавие на български език: IR/R ОКСИМЕТЪР.

аппаратной, что в перспективе может удешевить конструкцию оксигенометра до нескольких десятков долларов.

В данной статье будет изложен процесс построения устройства, а именно – электрическая принципиальная схема включения диодов и подключение к микроконтроллеру, тестирование устройства без программной обработки. Также в статье вкратце будут рассмотрены существующие аналоги-устройства. В завершение статьи будет представлен алгоритм работы всей системы и отрывки из кода программы, а именно – программная обработка сигналов, принятых с фотодиодов.

ИЗЛОЖЕНИЕ

Клетки организма нуждаются в кислороде, который переносится через кровь из легких в различные органы и ткани. В процессе дыхания кислород проникает в альвеолы в легком и присоединяется к клеткам, называемым эритроцитами.

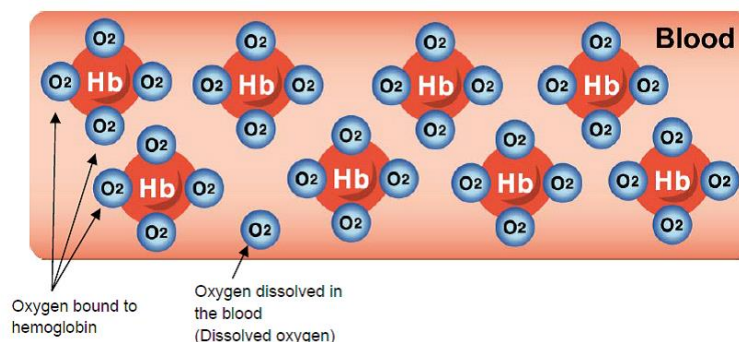


Рисунок 1 – Пример прикрепления кислорода к эритроцитам.

Гемоглобин связывает кислород и распространяет его по всему телу. В сочетании с кислородом образуется ярко-красное вещество, называемое оксигемоглобином. Как только гемоглобин отдает кислород он становится более темным и такое вещество называется дезоксигемоглобином. На рисунке 2 приведено соотношение коэффициента на сыщения крови оксигемоглобином и дезоксигемоглобином к длине волн красного и инфракрасного света.

Формула 1 – Вычисление на сыщения кислородом крови. Гемоглобин, связанный с кислородом, называется оксигенированным гемоглобином (HbO_2). Гемоглобин, не связанный с кислородом, называется дезоксигенированным гемоглобином (Hb). Насыщение кислородом представляет собой отношение оксигенированного гемоглобина к гемоглобину в крови, как определено следующим уравнением. Поскольку каждая молекула гемоглобина может связываться с 4 молекулами кислорода, она может связываться с 1-4 молекулами кислорода. Тем не менее, гемоглобин стабилен только тогда, когда он связан с 4 молекулами кислорода или не связан с каким-либо кислородом. Он очень нестабилен, когда связан с 1 - 3 молекулами кислорода. Поэтому, как показано на приведенном выше рисунке, гемоглобин существует в организме в виде дезоксигенированного гемоглобина (Hb) без связи с кислородом или в виде оксигенированного гемоглобина с 4 молекулами кислорода. Насыщение кислородом можно оценить с помощью SaO_2 или SpO_2 . SaO_2 представляет собой насыщенность кислородом артериальной крови, а SpO_2 – насыщенность кислородом, определяемое пульсоксиметром.

Они называются кислородной насыщенностью артериальной крови и перкутанной насыщенностью кислорода, соответственно.

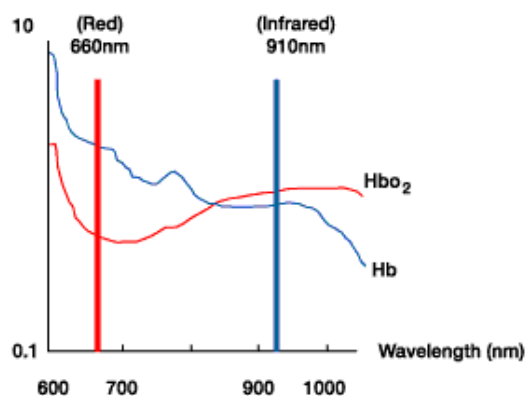


Рисунок 2. График абсорбции света гемоглобином

Вид у того что измерение насыщения кислородом крови жизненно необходимо для выживания человека в критических ситуациях в больницах и т.д., в данной статье был разработан собственный пример простейшего оксигенометра на основе микроконтроллера ArduinoYun, данный контроллер является избыточным и при желании и необходимости есть возможность переноса без особого изменения программной части проекта на более компактные микроконтроллеры поддерживающие язык программирования C/C++.

Метод измерения прибора основан на записи поглощения пропущенного через ткань (пальцы, уши и т. д.) света с двумя разны мидлинами волн, инфракрасный свет и красный. Чувствительность прибора составляет 0.0976% на бит АЦП, в данно ммикроконтроллере используется 10-битный АЦП и минимальное и максимальное цифровое представление в цифровой форме входящего аналогового сигнала составляет промежуток от 0 до 1023 бит. Исходя из того, что чем большесветового потока проходит через ткань и достигает фотодиода – тембольший процент насыщения крови кислородом мыиспользуем расчет – $100\%/1024$ в итогеполучаем 0.0986% на один бит (далееOxPerBit).

Для вычисления оксигенациимы используем: $OxPerBit * valRed$, где $valRed$ – получаемое значение с АЦП при работе красного диода, $OxPerBit * valIRed$, где $valIRed$ – получаемое значение с АЦП при работе инфракрасного диода, после этих вычислениймы проводим вычисление значений, полученных в ходе предыдущих расчетов – $valAll = (valRed / (valIRed + valRed)) * 100$, где $valAll$ – значение и того воеоксигенациикрови. Данный расчет выполняется по формуле 1.



Рисунок 3. Монитор порта Arduino с выводом результатов измерения оксигенации.

На рисунке 3 продемонстрирована работа оксигеномера при наличии толстой стенки, не пропускающей свет между диодами приемника и передатчика, как только поместили палец между диодами и убралиперегородку значения начали расти вплоть до 90-91 процента,

погрешность скорее всего присутствует из-за того, что вы полнялось тестирование на макетной плате без хорошей изоляции от источников внешнего освещения.

Но показания примерно правильны, т.к. было проведено сравнение с пульсоксиметром от компании TexasInstruments, отклонение наблюдалось около 1-3% оксигенации.

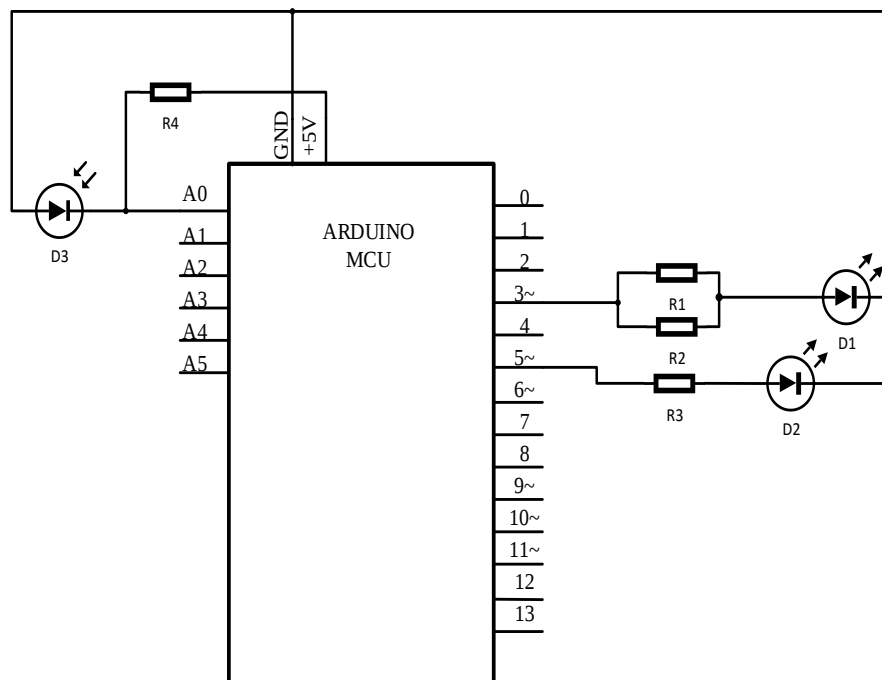


Рисунок 4. Принципиальная электрическая схема оксигенометра

На схеме на рисунке 4 представлена упрощенная схема микроконтроллера Arduino Yun, с изображением только цифровых входов/выходов и аналоговых пинов, также указаны пины питания +5V и пин заземления.

Также на рисунке представлены 2 диода, где D1 – инфракрасный диод, он подключен через 2 параллельно включенных резистора R1 и R2 номиналом 22 Ом каждый. В итоге на диод подается напряжение $\sim +5V$ через сопротивление общее 11 Ом. Данный диод является частью передатчика. Диод D2 является красным диодом и подключен уже через сопротивление R3 номиналом 220 Ом. Эти 2 диода составляют собой излучатель нужного нам света.

На рисунке 4 представлен также и фотодиод, включенный по стандартной схеме – анод заземлен, катод подключен к пину A0, через который мы получаем значение на АЦП и позже преобразовываем программно, также катод подключен к нагрузочному питанию через резистор R4 номиналом в 10 кОм. Принцип работы схемы: мы подаем прямой импульс на каждый из пинов передатчика и принимаем поток света на фотодиоде, который передает напряжение на пин A0.

Общий алгоритм работы системы представлен на рисунке 5.

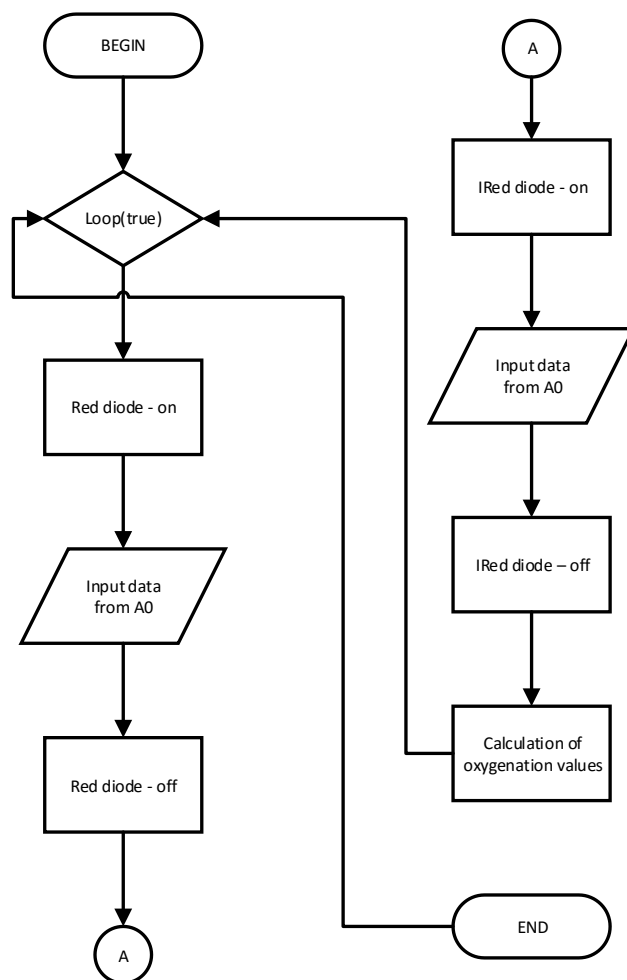


Рисунок 5. Блок-схема алгоритма работы оксигенометра

ВЫВОДЫ

В данной статье мы рассмотрели устройство, разработанное на основе принципов измерения оксигенации в крови, также было проведено тестирование устройства и сравнение с существующим аналогом от Texas Instruments. Устройство показало себя хорошо, имело небольшую погрешность в несколько процентов по сравнению с все тем же пульсоксиметром Texas Instruments.

REFERENCES

- Manukova A. (2011). Medicinskaelektronika. IzdatelskicentyrRuse (Оригинално заглавие: Манукова А., 2011, Медицинска електроника, 285 стр, inEnglish: Medicaelectronics, 285 pp.)
- Manukova A. ets (2016). Medicinskaelektronika – rykovodstvozaprakti`eskiupravneniya. IzdatelskicentyrpriRusenskiuniwersitet, 2016, 82 str., ISBN 978-954-712-701-2 (Оригинално заглавие: Манукова-Маринова А., И. Иванов, Л. Димитрова. Медицинска електроника – ръководство за практически упражнения. Издателски център при Русенския университет „Ангел Кънчев“, 2016 г., 82 стр., ISBN 978-954-712-701-2)

WEB-BASED PROJECT MANAGEMENT SYSTEM¹⁰

Tsvetelina Mladenova – MSc Student

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Tel.: +359 88 429 2155

E-mail: cvpmladenova@gmail.com

Assist. Prof. Yordan Kalmukov, PhD

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Phone: +359 82 888827

E-mail: jkalmukov@ecs.uni-ruse.bg

Abstract: *The paper presents a study of existing project management systems. After a thorough comparison is made a web based application is proposed. Some of the most important requirements are listed and the results of the system design are shown in the end. The paper presents only a fraction of an ERP system, which consist of the described Project Management System here.*

Keywords: *project management, work efficiency, ERP system*

ВЪВЕДЕНИЕ

Съвременните компании ежедневно са изправени пред проблеми свързани с ръководенето и ефективното разпределение на ресурси. Много често това е проблемът, който проваля до 75% от проектите на компанията (Burger, 2016).

Във време на силно развиващи се технологии и все по-комплексни проекти, най-важна е ролята на проектовия мениджър. За да бъде един мениджър характеризирен като добър, той трябва да може да предвижда рисковете и проблемите преди тяхната поява, и да разпределя задачите, така че екипът му да бъде оптимално натоварен и конкурентноспособен.

Този доклад представя система за управление на проекти, съобразена с изискванията и нуждите на малкия и среден бизнес. Тя представя модел на работа, целящ прозрачност, централизация на входните и изходните данни, контрол и самоконтрол на служителите и не на последно място – ефективна комуникация с клиентите.

Изложени са проблемите и трудностите, пред които са изправените съвременните компании. Направен е сравнителен анализ между традиционните методи за ръководене на проекти и е предложено решение, целящо да облекчи процеса и минимизира загубите, породени от грешни решения. Особено внимание е обърнато на жизнения цикъл на проекта, отделяйки всяка негова фаза в модул. Всеки модул е независим, за да може да бъде лесно проследим и управляем.

Трите основни въпроса, които стоят в началото на всеки проект, свързан с изграждане на система за управление и оптимизация на бизнес процеси са: „Как е било преди?“, „Как ще бъде?“ и „Как може да се подобри?“. Направено е проучване на начина на управление на компания за реклама и дизайн, което отговори на първия въпрос. След анализа на получените резултати са формулирани функционалните изисквания, на които трябва да отговаря една система за управление на проекти. По този начин се отговаря и на втория въпрос. В резултат

¹⁰ Докладът е представен на студентската научна сесия на 27 април 2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: Уеб-базирана система за управление на проекти

е предложена система, която да дава свобода и ясен поглед върху процеса на работа, повишавайки максимално ефективността и ограничавайки загубите. Системата може значително да подобри качеството на работа, като по този начин отговаря и на последния въпрос.

АНАЛИЗ НА СЪЩЕСТВУВАЩИТЕ РЕШЕНИЯ

Таблица 1. Сравнителен анализ на съществуващи решения

	Insightly	Smartsheet	DotProject
Интерфейс	Наподобяващ Gmail	Наподобяващ Excell	Остарял
Поддръжка	Безплатни уроци, видеа, статии	Уроци, видеа, статии	Няма активна поддръжка от 2014г.
Отчети	Ръчно създадени, предварително дефинирани и известия	Собствена конфигурация според избрани променливи	Отчети за изработени часове
Мобилна версия / приложение	Приложение	Приложение	Няма
Диаграми	Няма Гант диаграми	Гант диаграми	Гант диаграми
Интеграция с други приложения	Над 40 приложения (Google, MailChimp, Slack, OneDrive и др.)	Интеграция с Google приложения	Няма
Цена	до 163,90 лв	до 48 лв	Безплатно

От направения сравнителен анализ е видно, че в зависимост от нуждите на проективия мениджър и екипа му, има системи, предлагащи голям набор от функционалности.

Това, което липсва и при трите системи е ролята на индивидуалния член на екипа. За да бъде успешен един проект, то неговия екип трябва да работи в синхрон, а за този синхрон трябва да се погрижи проективия мениджър, който от своя страна разчита на софтуера за управление. Много често се налага един служител да се включи в няколко проекта или задачи едновременно. Това може да стане само с добро управление и координация. При разгледаните системи това не е възможно.

И в трите системи липсват действия, които да контролират служителите, така че превишаването на бюджета или неспазването на сроковете да бъде избегнато.

УЕБ-БАЗИРАНА СИСТЕМА ЗА УПРАВЛЕНИЕ НА ПРОЕКТИ

Целта е да се проектира и реализира уеб-базирана система за управление на проекти, която да обединява функционалните възможности на съществуващите решения, но премахвайки излишната функционалност, която няма отношение към ръководенето на проекти; да наблегне на човешкия фактор по време на жизнения цикъл на проекта; и да подтиква към вземане на адекватни решения навреме. Системата трябва да централизира входния и изходния поток от данни, да следи за качеството на продукта, да повиши ефективността при работата с клиенти и да дава точни справки.

В бъдеще системата трябва да може да бъде надградена до ERP¹¹ система, която да събере и организира цялото ръководство на компанията, повишавайки силно ефективността на всички звена в нея.

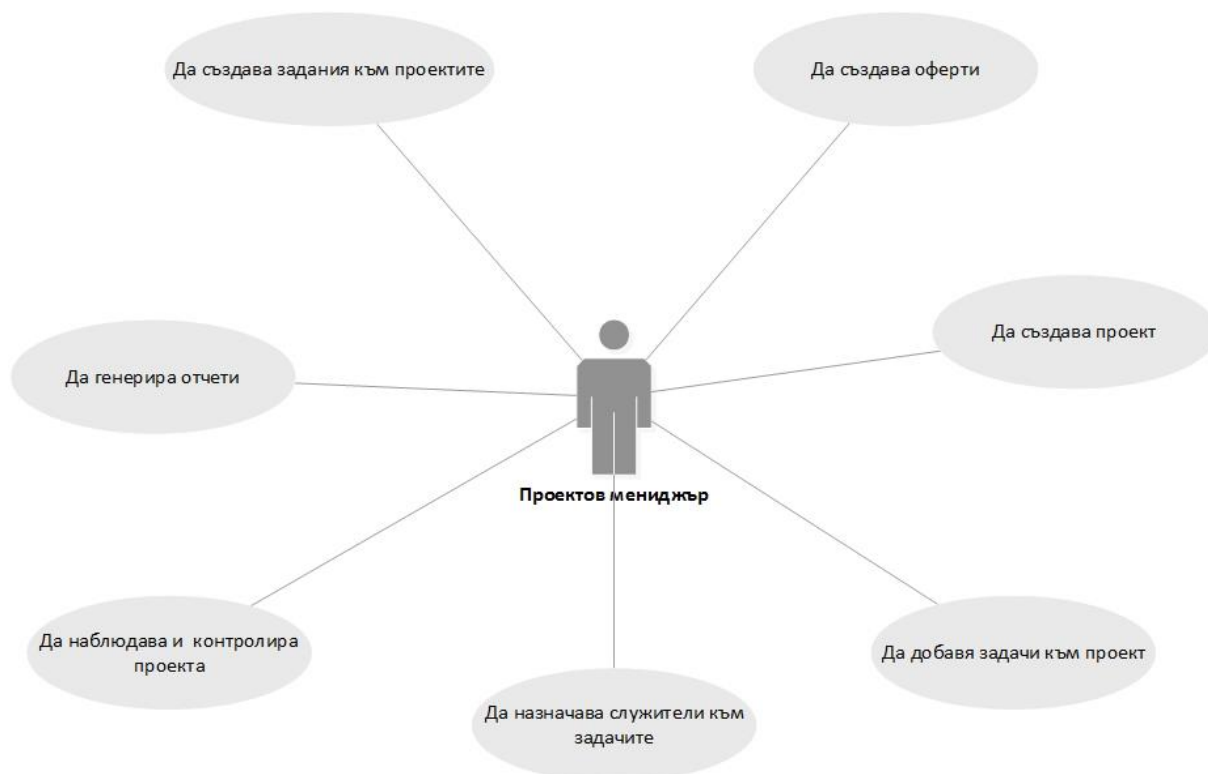
¹¹ Enterprise Resource Planning

Функционални изисквания

От направения сравнителен анализ на съществуващите решения е видно, че системата трябва да покрива основните изисквания, които да я направят годна за работа. По-конкретно:

- Да обхване всички дейности на организацията, свързани с:
 - Управление на работните процеси, отчитане и контрол на дейността на служителите;
 - Осигуряване на качество на производствения продукт посредством намаляване на неточностите, оптимизиране на работното време, приоритизиране на работата;
 - Въвеждане на единен вход и изход на данните в системата;
- Да осигури удобство за работа на служителите, както и да подобри комуникацията между тях;
- Да осигури контрол и самоконтрол на служителите;
- Да осигури обратна връзка с клиентите посредством:
 - Бързо офертиране на проект, изготвяне на точно задание, въвеждане на задачите в работния процес;
 - Изпращане и предоставяне на отчети до клиентите;
- Да може да осигури поглед върху текущите проекти, с цел планирането на нови;
- Да бъде лесно достъпна, независимо от апаратното осигуряване или местоположението на потребителя;
- Да бъде устойчива във времето и да се развива;

Взаимодействието на проектово мениджър със системата е най-голямо. Той трябва да може да управлява целия проект през всички етапи на жизнения цикъл. Функционалността на системата по отношение на него е представена на фиг. 1.



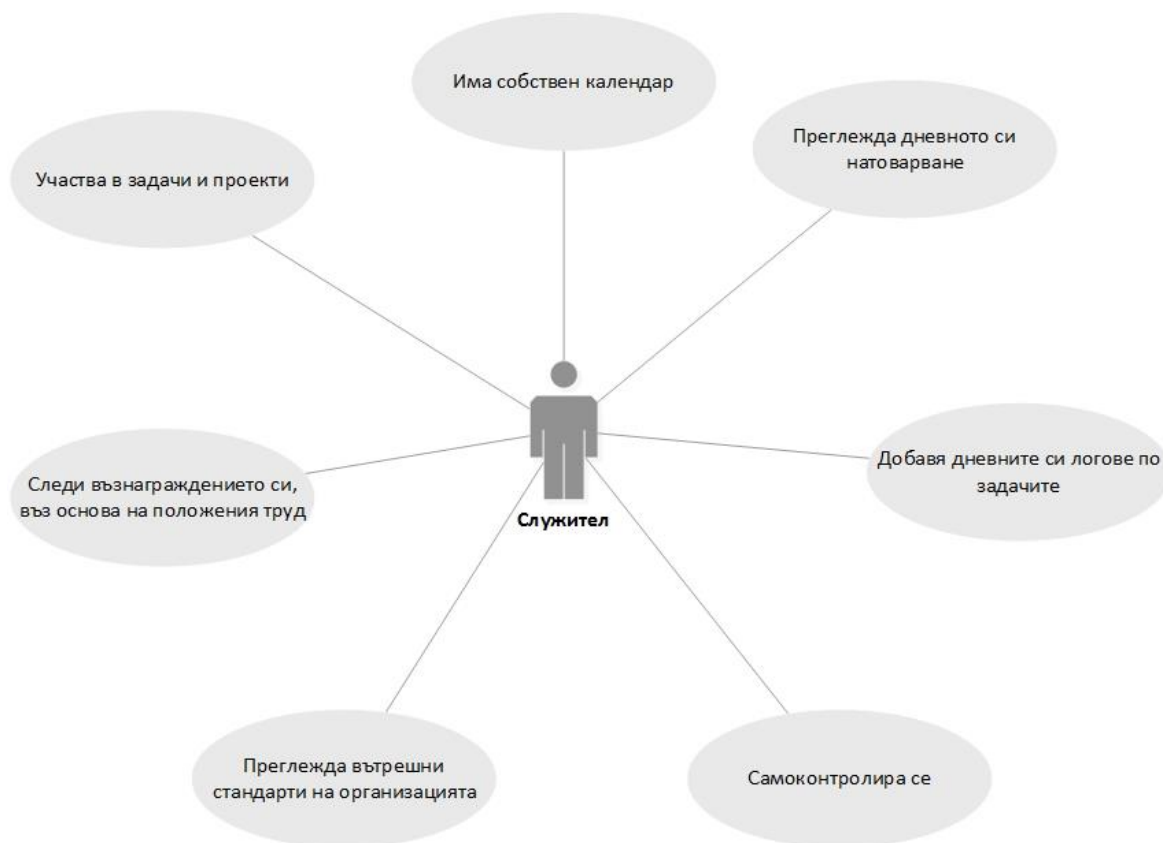
Фиг. 1. Диаграма на случаите на употреба, отразяваща функционалността на системата по отношение на мениджъра на проекти

Някои от основните функции, до които мениджърът трябва да има достъп са:

- Създаване на оферти, разбивайки проекта на задачи и оценявайки тези задачи. Потвърдената оферта от клиент е фазата, която ще даде старт на проекта;
- Писане на задания към проекта – мениджърът трябва да може да създава функционални и технически задания към проекта;
- Разбиването на проекта на по-малки задачи ще помогне за поетапното му развитие, лесния контрол и ефективната поддръжка;
- Назначаването на служители към отделните задачи, спрямо техните квалификации и умения е ключът към качествения краен продукт. Оптималното разпределение и натоварване на служителите ще повиши тяхната работоспособност;
- Наблюдението и контрола на проекта по време на неговото развитие са от съществено значение, за да се избегнат бъдещи проблеми;
- Генерирането на отчети за свършената работа ще помогнат да се проследи работата на екипа и ще даде представа на клиента как се развива неговия проект;

Служителите са вторите по важност потребители на системата. За тях системата е центърът на ежедневната им работа. Тя е тази, която ще назначи дневните им задачи, ще даде представа за изработеното им време и не на последно място – тя ще изчисли тяхното възнаграждение. Функционалността на системата по отношение на служителите е представена на фиг. 2.

От системата служителите ще могат да се самоконтролират, спазвайки вътрешните стандарти за качество на организацията. Това е доказано най-добрият метод за качествен продукт – собственото желание за усъвършенстване и развитие.



Фиг. 2. Диаграма на случаите на употреба, отразяваща функционалността на системата по отношение на служител

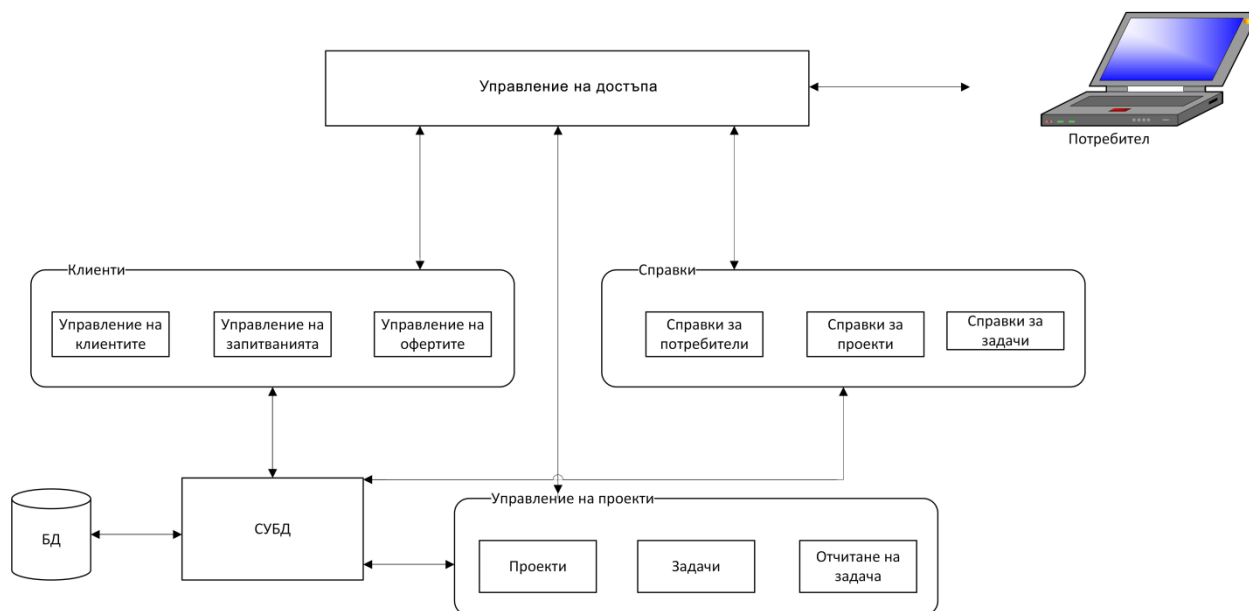
Архитектура

Модулите на приложението могат да бъдат групирани в няколко по-обща структури. На Фиг. 3 е показана архитектурата на системата. Трите основни групи от модули са „Клиенти“, „Справки“ и „Проекти“.

Структурата „Клиенти“ съдържа в себе си модулите за клиенти, запитвания и оферти. Това са трите модула, без които не може да съществува структурата „Проекти“.

Структурата за проекти събира модулите за проекти, задачи и отчитане на задачи. Тези модули са отговорни за управлението и проследяването на проектите.

Последната структура е тази за отчети. Тя не би могла да съществува, без предходните две. Тази обобщена структура е отговорна за отчитането на проектите, състоянието на задачите и дейностите на потребителите.



Фиг. 3. Архитектура на системата

Реализация

Избраният език за програмиране е PHP, а системата за управление на бази от данни - MySQL. При избора на език за програмиране са взети под внимание поддръжката и документацията на езика, бързината му, платформената зависимост и интеграцията. След обстойно проучване на няколко езика за програмиране, беше избран PHP като език, покриващ всички изисквания.

При избора на СУБД са разгледани няколко варианта – PostgreSQL, SQLite и MySQL.

Основен критерии при избора са бързината и лесния достъп до данните. При MySQL е необходимо да се укажат само адреса на сървър, името на потребителя, парола и име на базата от данни, и връзката е осъществена. Докато при SQLite базата от данни трябва да се намира на локалния компютър, за да може приложението да има достъп до нея, което би затруднило бъдещата реализация на системата като разпределена.

ЗАКЛЮЧЕНИЕ

В обобщение на написаното по-горе, е проектирана и реализирана система за управление на проекти. Тя дава възможност на проектовете мениджъри да разпределят задачи, следят за тяхното изпълнение и извършват ежедневните дейности по управление на проектите лесно.

При проектирането са взети предвид резултатите, получени при анализа на съществуващи решения, които са полезни със своите слаби и силни страни.

От голяма полза е и проведеното проучване на целевата група. То показва кои са проблемните места при организацията и начина на провеждане на работния процес. От анализите, направени във фазата на проучвания, са формирани функционалните изисквания към системата. По време на разработката тези изисквания са доразвити и обединени в модули, които по-късно са трансформирани в модулите на реализираното приложение.

Основната идея на системата за управление на проекти е тя да стане част от ERP система, която да поеме цялата организация на компанията. С модули като „Клиенти“ и „Потребители“,

които са опорни данни за една ERP система, и бъдещи модули като „Счетоводство“ и „Търговски модул“, явяващи се оперативни модули, надграждането на системата за управление на проекти до бизнес информационна система, ще бъде лесно и безпроблемно.

С прехода на текущо разработеното приложение до приложение, следящо цялата организация на компанията, ще се наложи още по-голям контрол над действията на потребителите. Злонамерените действия също не са изключени. В името на контрола и защитата от загуба на данни, ще бъде разработен модул „История“, който ще съдържа архивна информация за всяко действие на потребителите в системата. Идеята на модула е да има ясна представа кой, кога и какво е променял в системата. Така при пробив в сигурността, ще може да се възстановят много лесно повредените данни и да се проследи източника на проблема.

БЛАГОДАРНОСТИ

Този доклад се публикува със съдействието на проект 2018-ЕЕА-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Burger, R. (2016, September 26). *20 Surprising Project Management Statistics*. Retrieved from Capterra: <http://blog.capterra.com/surprising-project-management-statistics/>

WEB-BASED SYSTEM FOR MAKING DOCTOR'S APPOINTMENTS¹²

Tsvetelina Mladenova – MSc Student

Department of Computer Systems and Technologies,

University of Ruse "Angel Kanchev"

Tel.: +359 88 429 2155

E-mail: cvpmladenova@gmail.com

Birdzhan Hasan – Eng. Designer

Department of Industrial Design,

University of Ruse "Angel Kanchev"

Tel.: +359 88 347 2276

E-mail: birdzhanhasan@hotmail.com

Assist. Prof. Yordan Kalmukov, PhD

Department of Computer Systems and Technologies,

University of Ruse "Angel Kanchev"

Phone: +359 82 888827

E-mail: jkalmukov@ecs.uni-ruse.bg

Abstract: *This paper proposes a system designed to make a new way of scheduling doctor's appointments and a way to shorten the time-consuming communication between the doctors and the patients. The paper shows the basic needs of a system of that kind and in addition to that, some extra functionalities are present. A study based on the examination of existing systems and applications combined with a small study of potential users has led to the developing of a prototype of the system. This paper has been wrote in association with a marketing student and a designer.*

Keywords: Health, Appointments, Online Health Card

ВЪВЕДЕНИЕ

"21.7% от хората в големите градове са със затруднен достъп до медицински услуги. Основните причини са липса на медицинска помощ в близост."(Dimitrov, S., 2008)

Липсата на достъпно здравеопазване е основен проблем, стоящ пред обществото. Процесът по намиране на подходящ лекар, особено за младите пациенти, се оказва още по-труден.

Прегледа при лекар е неизменна част от живота на всеки човек. За запазване на преглед често са необходими няколко действия, броят на които може да бъде значително намален, а времето по процедурата – съкратено.

Единна система за записване на прегледи, водене на здравен картон и обмен на съобщения между лекари, пациенти и здравни организации, би могла значително да облекчи тромавия процес.

ОБЗОР НА СЪЩЕСТВУВАЩИ РЕШЕНИЯ

Съществуват редица сайтове и платформи, предлагащи на своите посетители по-лесно намиране на специалисти, запазване на часове за преглед и по-лесен достъп до медицински услуги. Далеч не всички са развити до нивото, на което се предполага да бъде една такава услуга, но много от тях предлагат интересни решения и подходи. Сравнителен анализ между няколко български и чуждестранни платформи е показан в табл. 1.

¹² Докладът е представен на студентската научна сесия на 27 април 2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: Уеб-базирана система за запазване на прегледи при лекар

Таблица 2. Сравнителен анализ на съществуващи решения

Платформа	DocPlanner	Planner	Superdoc	ZocDoc
Произход	Полша, Турция, Испания, Италия, Мексико, Аржентина, Чили	България	България	САЩ
Основни функционалности	Намиране на лекар, запазване на час	Запазване на час за лекар, фризьор, масажист, фитнес	Намиране на лекар, запазване на час	Намиране на лекар
Наличие на отзиви	Да	Не	Да	Да
Намиране на лекар по критерии	По държава	По град	По град или специалист	По болест или град

От проучването е видно, че много от системите имат потенциал и възможност за развитие. Разработката на една система, която да обобщава различните функционалности на съществуващите решения би била ефективна и полезна.

ИЗИСКВАНИЯ КЪМ СИСТЕМАТА

Произхождайки от направения анализ на съществуващите решения и проучване на потенциални пациенти и бъдещи потребители на системата, са съставени два списъка с изисквания към нея – основни изисквания, които ще са достатъчни за базовото ѝ функциониране и изисквания, които да ги допълват, но да не спират нейната работа при евентуалната им липса. Тези два списъка определят и основните типа профили – безплатен и платен.

Основни изисквания към системата

Набор от базови функционалности дават възможността на всеки потребител в системата да се регистрира, като по този начин му бива създаден автоматично здравен картон. Здравният картон има за цел да обедини цялата история на заболяванията на пациента, давайки му удобството да преглежда изследванията си, предписаните му лекарства и дори да поръча лекарствата директно от приложението. Основното предимство на здравния картон е централизирането на информацията и достъпа до нея по всяко време и място. Приложението е уеб базирано, което дава необходимата гъвкавост то да бъде достъпно до всички потребители, независимо от тяхното местоположение и възраст.

В приложението участват три основни типа профили, условно наречени: пациент, лекар и фармацевтична компания. Всички профили, с изключение на пациентите, ще имат версия за безплатен или платен достъп. Базирайки се на приходите от месечните абонаменти на лекарите и фармацевтичните компании, приложението ще може да генерира приходи, достатъчни за издръжката му. Едно от задължителните изисквания е пациентите да не бъдат допълнително обременявани с такси за използването на услугата. С представеното по-горе решение, е възможно изискването да бъде удовлетворено, предоставяйки пълната функционалност на пациентите безплатно.

Тъй като основната дейност, която ще се предлага, е лесното записване на часове за преглед, базово изискване е всички стъпки да бъдат опростени и минимизирани. Предложението това да бъде направено е, чрез добавянето на вече споменатия профил „лекар“.

Всеки лекар може да се регистрира, като му се предоставя възможност да изгради свой график, отбелязвайки свободните си часове. Така, когато пациент влезе в приложението и намери своя лекар, ще може да види свободните му часове и да запази най-удобния за него. Разбира се, не се изключва и възможността от злоупотреби, затова всяка заявка преминава през лекаря за преглед и потвърждение. Това прави стъпките 3, което в много случаи е общия брой от стъпки, но има няколко ключови предимства за използването на системата: а) търсенето на лекаря, контактните му данни и мнения за него са на едно място, видимо за всеки, б) добавянето на график от страна на лекаря може да става по всяко време, позволявайки му да бъде гъвкав и динамичен и в) пациентът може да запази часа си по всяко време на денонощието, премахвайки неудобството от „подходящо време за записване“.

След извършен преглед, а дори и по време на самия преглед, лекарят въвежда данни от прегледа в картоната на пациента. Предписаните лекарства от прегледа също биват въвеждани, а след като пациента влезе в системата, същите лекарства му биват предлагани за закупуване. Това е основно изискване към профила „фармацевтична компания“. Всяка компания или аптека може да се регистрира за безплатен профил, като добавя своите лекарства за продажба. По този начин търговците се появяват на нов пазар, какъвто е онлайн търговията, увеличавайки продажбите и позициите си.

От проведеното проучване се установи, че начините за намиране на лекар са прекалено общи и крайно не достатъчни. В по-голямата част от проучените решения се забелязва търсене на лекар по град или специалност. В много от случаите това е достатъчно, но когато пациент има нужда от лекарска помощ, той може да опише собственото си състояние в момента и не винаги е наясно какъв специалист му е необходим. Изхождайки от тази гледна точка, пред системата се поставя изискването търсенето на специалист да може да става и чрез описание на симптомите или въвеждане на болест. Така предварително дефинирани алгоритми биха дали предложение за подходящ специалист, като основни критерии ще бъдат близостта му до пациента и неговия рейтинг в системата.

Допълнителни изисквания

Всички изисквания, поставени като основни, имат за цел да предоставят набор от основни функционалности, напълно подходящи за функционирането на системата. Те са достатъчни за базовата работа на всички звена в системата, но не описват пълния капацитет от възможности на проекта.

Тъй като допълнителните изисквания не влияят на основната функционалност на системата, а я надграждат и обогатяват, то логично достъпът до тях е възможен с платените профили.

С допълнителните възможности профилът от тип „лекар“ ще може да:

- Генерира финансови отчети за извършените прегледи;
- Комуникира директно с други лекари и организации през системата;
- Получава данни за прегледите и изследвания на своите пациенти от други специалисти;
- Насрочва специализирани прегледи;
- Интегрира системата с външни приложения като Google Calendar, Gmail и т.н.

Всички функционалности, които профилът „лекар“ получава като пакет, имат за цел да намалят драстично времето по административна работа, за да остане повече време за обслужване на пациенти.

Основен доход на системата ще бъде и рекламата на лекарства. Тук активно биха участвали регистрираните аптеки и фармацевтични компании. Честа практика е лекарите да предлагат лекарства на избрани компании, имайки някакви споразумения с тях. На всеки пациент, решил да поръча и закупи лекарство от системата, ще му бъде предложен и заместител, с който той да сравни и избере. Факт е, че търсенето на лекарства е огромно, а признак за намаляването му няма. Инструмент като текущата система, би могъл да даде богат

набор от предложения на пациентите, а аптеките от своя страна ще предлагат продуктите си на достъпна цена, изхождайки от конкурентните цени.

Предимство за аптеките би била и опцията за преференциални цени и отстъпки към група лекарства, лекари или пациенти. Модул, грижещ се за ценообразуването, ще направи управлението на продуктите лесно и по-евтино, като ще се елиминират разходи по склад на скъпоструващи и редки лекарства.

ОЧАКВАНИ ТРУДНОСТИ

Трудностите и рисковете, стоящи пред системата не са малко – изработка на надежден софтуер, който да осигури постоянен достъп на лекари и пациенти, голяма вероятност от недоброжелатели, целящи да използват системата за неморални и неправилни действия, популяризиране на системата сред обществото, интеграцията с болнични заведения и организации, и т.н..

За да се избегнат някои от очакваните проблеми, се вземат предвид някои основни мерки за сигурност:

- Криптирана връзка на заявките;
- Разпределение на данните на различни сървъри и точки по модела на CDN¹³, за да може достъпът да бъде постоянен, а връзката по-бърза и надеждна;
- Преглед и верификация на всеки регистриран пациент;
- Преглед и верификация на всеки регистриран лекар;
- Преглед и верификация на всяка регистрирана фармацевтична организация;
- Потвърждение на всеки записан час от лекар;
- Ограничение на достъпа до личните данни на пациента – само лекарят, който обслужва пациента, ще може да вижда данните му.

ИЗВОДИ

Реализацията на веб-базирана система за управление на здравен картон и прегледи е необходимост, произлизаща от нуждата за по-достъпни здравни грижи.

Събирането на всички процеси и данни по обработката на преглед на едно място ще повиши силно ефективността на всички участващи потребители. Имайки всички данни за прегледа, симптомите, резултатите и изписаните лекарства на едно място, пациентите ще имат пълна информация за своето здравно състояние.

Целта на системата е тя да бъде основно приложение за управление на електронните картони и съпътстващите ги обработки.

Начина на запазване на час или издаване на рецепта ще бъде унифициран, като това ще създаде стандарт за работа и ще повиши качеството на предлаганите услуги.

БЛАГОДАРНОСТИ

Този доклад се публикува със съдействието на проект 2018-ЕЕА-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Dimitrov, S (2008), Dori i v gradovete vseki peti trudno stiga do lekar, Newspaper SEGA, URL: <http://www.segabg.com/article.php?id=391593>

¹³ Content Delivery Network

COMBINATIONAL CIRCUITS DESIGN USING LOGISIM 2.7.1¹⁴

Milotin Barakov – Student

Faculty of Electrical Engineering, Electronics and Automation
Department of Computer Systems and Technologies
University of Ruse “Angel Kanchev”
Phone: +358 87 79 90 817
E-mail: milotinbarakov@gmail.com

Assist. Prof. Adriana Borodzhieva, PhD

Faculty of Electrical Engineering, Electronics and Automation
Department of Telecommunications,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 734
E-mail: aborodzhieva@uni-ruse.bg

Abstract: The paper reviews designing and simulating combinational circuits using integrated circuits (IC) inside Logisim's integrated development environment (IDE). It shows the process of inputting information and creating different circuits. Accents fall on acquiring a practical realization of the scale and requirements for a given IC.

Keywords: Integrated Circuits, Education, Logisim, Simulation, Model, Design, User-Friendly.

ВЪВЕДЕНИЕ

Дисциплината „Цифрова схемотехника“ има за цел да запознае студентите от специалност „Компютърни системи и технологии“ с основните въпроси на цифровата електроника. Основният принцип на дисциплината е да се свържат функционалните възможности на цифровите елементи с тяхната микроелектронна база и с приложението им при изграждането на импулсни и цифрови устройства. Придобитите знания се използват в специалните и специализиращите дисциплини от областта на цифровата техника, включени в учебния план на специалността (<https://www.ecet.ecs.uni-ruse.bg/else>, 2018).

В процеса за придобиване на базови знания в областта на интегралните схеми (ИС) в курса по „Цифрова схемотехника“ всеки студент трябва да приложи уменията си, като създаде комбинационна схема, за което се изисква адекватна виртуална среда. Пример за такава среда е Logisim – образователен продукт с отворен код, създаден с цел използването му в учебния процес в университети и други учебни заведения. Той позволява създаването и симулацията на всевъзможни видове схеми – от прост ключ, включващ диод, до създаване на цели виртуални компютри. Средата е съвместима с всички компютри, поддържащи Java 5, т.е. поддържа се от всички разпространени десктоп операционни системи – Microsoft Windows, Mac OS X и повечето дистрибуции на Linux. Тя е лека и потребителски насочена, с ниски изисквания по отношение на хардуер (<http://www.cburch.com/logisim/>, 2018).

ИЗЛОЖЕНИЕ

Всяка цифрова схема попада в една от следните две категории: комбинационна схема или последователностна схема. В една комбинационна схема, изходните сигнали на схемата зависят само от текущите входни сигнали, докато при последователностната схема някои изходни сигнали зависят и от предходни входни сигнали.

¹⁴ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ПРОЕКТИРАНЕ НА КОМБИНАЦИОННИ СХЕМИ С ИЗПОЛЗВАНЕ НА LOGISIM 2.7.1

Анализът на комбинационните схеми е по-прост. За описание на поведението на такива вериги се използват три основни техники: 1) логически схеми, 2) Булеви изрази, позволяващи алгебрично представяне на начина на работа на схемата, 3) таблици на истинност, изброяваща възможните входни комбинации и съответните изходи.

Модулът „Комбинационен анализ“ на Logisim позволява преобразуване между тези три представяния във всички посоки. Това е особено удобен начин за създаване и разбиране на вериги с еднобитови входи и изходи (<http://www.cburch.com/logisim/>, 2018).

Реализация на непълен дешифратор на десетичните цифри

За представяне възможностите на програмата Logisim и в частност на модула „Комбинационен анализ“ е проектиран непълен дешифратор на десетичните цифри, представени в код +3 (с излишък 3). По аналогия на застъпената за изучаване в учебния процес интегрална схема SN74LS138, представляваща пълен тривходов дешифратор с *инверсни* изходи, производство на Texas Instruments, при проектирането на устройството са избрани активно ниво 1 по вход и активно ниво 0 по изход (Фиг. 1) (Tsonev, V. 2003), (Todorova, Sv., Y. Ruseva, 2008).

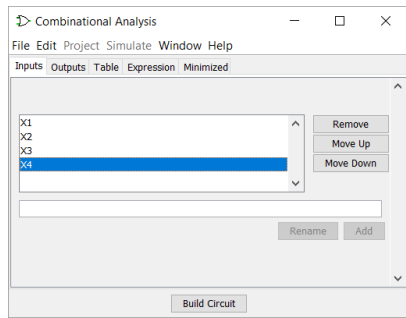
№	Д.Ц.	X1	X2	X3	X4	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
0	-	0	0	0	0	-	-	-	-	-	-	-	-	-	-
1	-	0	0	0	1	-	-	-	-	-	-	-	-	-	-
2	-	0	0	1	0	-	-	-	-	-	-	-	-	-	-
3	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1
4	1	0	1	0	0	1	0	1	1	1	1	1	1	1	1
5	2	0	1	0	1	1	1	0	1	1	1	1	1	1	1
6	3	0	1	1	0	1	1	1	0	1	1	1	1	1	1
7	4	0	1	1	1	1	1	1	1	0	1	1	1	1	1
8	5	1	0	0	0	1	1	1	1	1	0	1	1	1	1
9	6	1	0	0	1	1	1	1	1	1	1	0	1	1	1
10	7	1	0	1	0	1	1	1	1	1	1	1	0	1	1
11	8	1	0	1	1	1	1	1	1	1	1	1	1	0	1
12	9	1	1	0	0	1	1	1	1	1	1	1	1	1	0
13	-	1	1	0	1	-	-	-	-	-	-	-	-	-	-
14	-	1	1	1	0	-	-	-	-	-	-	-	-	-	-
15	-	1	1	1	1	-	-	-	-	-	-	-	-	-	-

Фиг. 1. Таблица на истинност на непълен дешифратор на десетичните цифри, представени в код с излишък 3 (активно ниво 1 по вход и активно ниво 0 по изход)

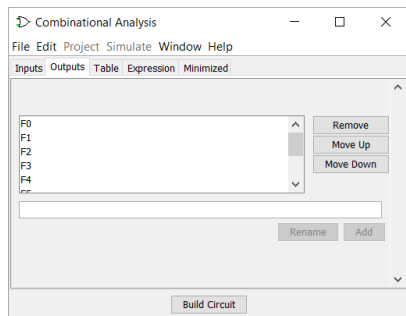
Дефиниране на входни и изходни променливи в програмата

В Logisim, в модула „Комбинационен анализ“ (Combinational Analysis) се дефинират четирите входни променливи x1, x2, x3, x4 (Фиг. 2 а) и десетте изходни функции на непълния дешифратор F0, F1, F2, F3, F4, F5, F6, F7, F8, F9 (Фиг. 2 б). Разписва се таблицата на истинност на дешифратора (Фиг. 2 в), след което програмата извежда минимизирания изходни функции, с помощта на карти на Карно (Фиг. 3). Програмата дава възможност да се генерира схемата по зададената таблица на истинност в базис 1 (И, ИЛИ, НЕ), както и с помощта на И-НЕ логически елементи без ограничение на входовете или с помощта само на двувходови И-НЕ логически елементи. Построените схеми, обаче, не са оптимизирани по отношение на инверторите, например, а е възможно да съдържат и други дублирани реализации на някои минтерми/макстерми (Фиг. 4).

Logisim няма вградени механизми, спомагащи оптимизацията на построената схема от модула „Комбинационен анализ“. Необходимо е внимателно да се проследят входовете и изходите на всеки логически елемент. В посочения случай многократно се инвертират входните сигнали. Построяват се шини, „носещи“ инверсните стойности на x1, x2, x3 и x4, след което сигналите от тези шини се използват на местата на инвертиращите логически елементи. В случая не се налага минимизация след първо стъпало (Фиг. 5).



a)

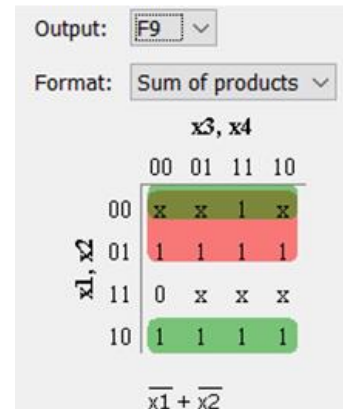
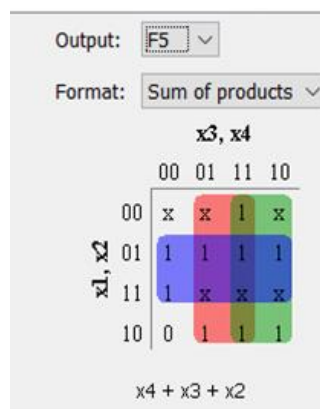
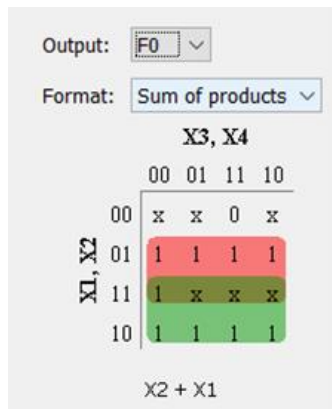


б)

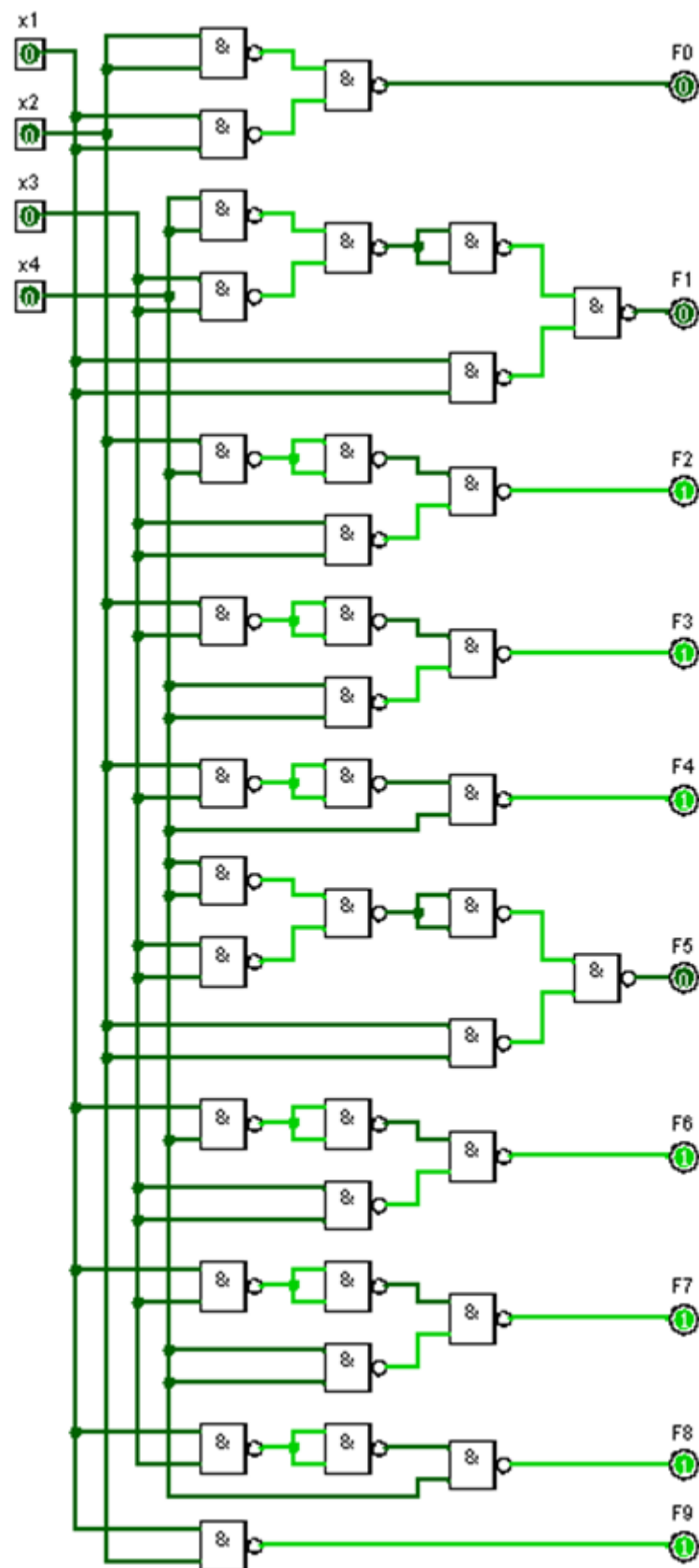
X1	X2	X3	X4	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
0	0	0	0	x	x	x	x	x	x	x	x	x	x
0	0	0	1	x	x	x	x	x	x	x	x	x	x
0	0	1	0	x	x	x	x	x	x	x	x	x	x
0	0	1	1	0	1	1	1	1	1	1	1	1	1
0	1	0	0	1	0	1	1	1	1	1	1	1	1
0	1	0	1	1	1	0	1	1	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	1	1	1	1	0	1
1	1	0	0	1	1	1	1	1	1	1	1	1	0
1	1	0	1	x	x	x	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x	x	x	x

в)

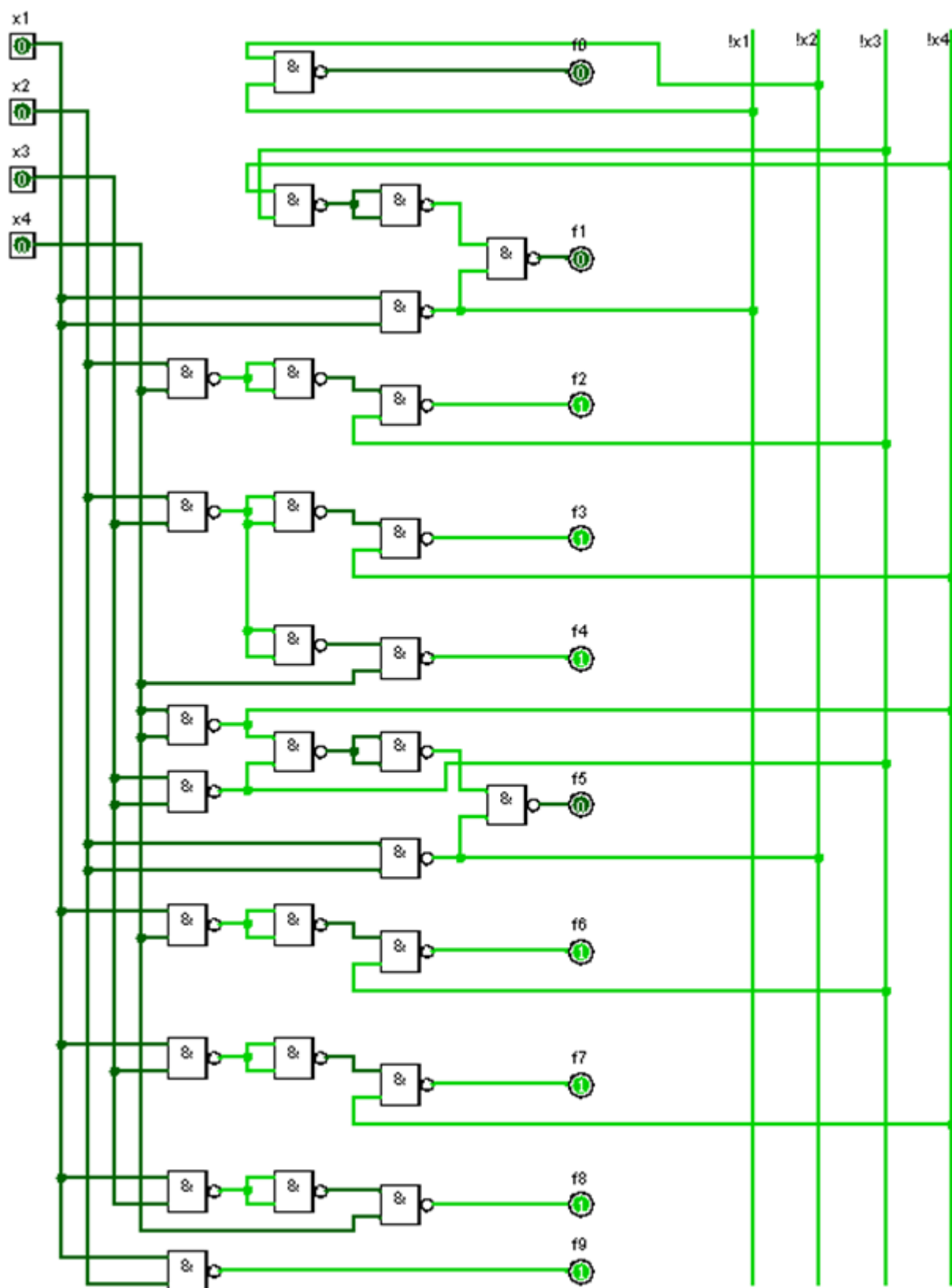
Фиг. 2. Дефиниране на входните и изходните променливи и таблицата на истинност на проектираното устройство в модула „Комбинационен анализ“



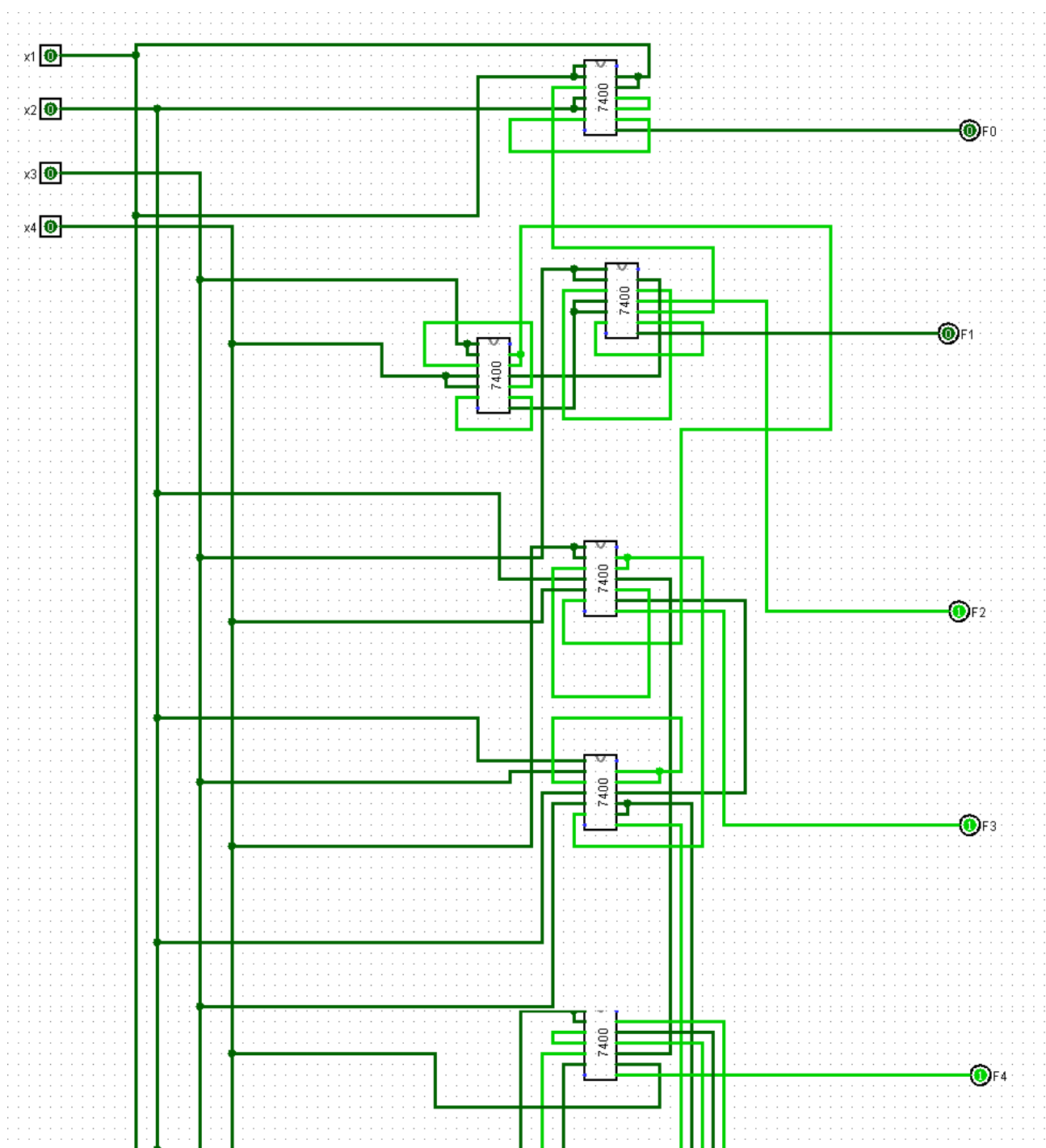
Фиг. 3. Минимизация на изходните функции (например, F0, F5, F9)



Фиг. 4. Автоматично генерирана схема на устройството с И-НЕ логически елементи



Фиг. 5. Оптимизирана по отношение на инвертори схема на устройството с И-НЕ елементи



Фиг. 6. Реализация на непълния дешифратор с използване на библиотека 74** (изображението е изрязано след F4, поради големите си размери)

Реализация на устройството с използване на библиотека 74***

С развитието на цифровата схемотехника и технологията за производство на интегрални микросхеми става икономически и технически целесъобразно разработката и производството на различни интегрални схеми, реализиращи отделни логически функции – И (AND), ИЛИ (OR), НЕ (NOT), И-НЕ (NAND), ИЛИ-НЕ (NOR), както и комбинационни и последователностни схеми, съдържащи значителен брой елементарни логически елементи – шифратори, мултиплексори, тригери, регистри, броячи, памети и др., до най-сложните микропроцесорни схеми и процесори за обработка на графична информация, при които в един корпус са обединени милиони елементарни логически елемента (<https://learn.sparkfun.com/tutorials/integrated-circuits>, 2018).

В Интернет е налична и може безплатно да бъде изтеглена библиотека от готови чипове, включително и застъпените за изучаване в учебния процес интегрални схеми по дисциплината

„Цифрова схемотехника“ (SN7400, SN7410, SN7420, SN74LS138, SM74153, SN7474) (<https://74x.weebly.com/blog/library-of-7400-logic-for-logisim>, 2018). С помощта на тази библиотека студентите трябва да реализират устройството, като изберат подходящи интегрални схеми (без да се поставя ограничението да се използват само застъпените в учебния процес интегрални схеми). По този начин студентът получава представа за реалния процес при проектирането и реализация на цифрово устройство с помощта на интегрални схеми.

При реализацията е необходимо да се намерят чипове от библиотека 74**, способни да извършват желаните от нас аритметико-логически операции. В случая е достатъчна само интегрална схема SN7400, включваща четири броя двувходови И-НЕ логически елемента (поради факта, че е зададено построяването на дешифратора само с двувходови И-НЕ логически елементи в Logisim) (Фиг. 6).

ИЗВОДИ

Дисциплината „Цифрова схемотехника“ запознава студентите от специалност „Компютърни системи и технологии“ с основните въпроси на цифровата електроника, като се свързват функционалните възможности на цифровите елементи с тяхната микроелектронна база и с приложението им при изграждането на импулсни и цифрови устройства. В процеса за придобиване на базови знания всеки студент трябва да приложи уменията си, като създаде комбинационна схема, използвайки адекватна виртуална среда, например Logisim – образователен продукт с отворен код, който позволява създаването и симулацията на различни видове схеми. Средата е съвместима с всички компютри, поддържащи Java 5, т.е. поддържа се от всички разпространени десктоп операционни системи – Microsoft Windows, Mac OS X и повечето дистрибуции на Linux. Тя е лека и потребителски насочена, с ниски изисквания по отношение на хардуер.

В доклада е показано приложението на модула „Комбинационен анализ“, който подпомага студентите при изучаването на цифрови устройства, застъпени в учебния процес по дисциплините „Синтез и анализ на логически схеми“ и „Цифрова схемотехника“. С помощта на библиотека 74** студентите трябва да реализират устройството, като изберат подходящи интегрални схеми (без да се поставя ограничението да се използват само застъпените в учебния процес интегрални схеми). По този начин студентът получава представа за реалния процес при проектирането и реализация на цифрово устройство с помощта на интегрални схеми.

REFERENCES

Tsonev, V. Impulsni i tsifrovi ustroistva. Zapiski, Rusenski universitet “Angel Kunchev”, 2003. (**Оригинално заглавие:** Цонев, В. Импулсни и цифрови устройства. Записки, Русенски университет „Ангел Кънчев“, 2003.)

Todorova, Sv., Y. Ruseva. Sintez i analiz na logicheski shemi. Rusenski universitet “Angel Kunchev”, 2003. (**Оригинално заглавие:** Тодорова, Св., Й. Русева. Синтез и анализ на логически схеми. Русенски университет „Ангел Кънчев“, 2008.)

<https://www.ecet.ecs.uni-ruse/bg/else>

<http://www.cburch.com/logisim/>

<https://learn.sparkfun.com/tutorials/integrated-circuits>

<https://74x.weebly.com/blog/library-of-7400-logic-for-logisim>

DEVELOPING WEB SERVICES IN JAVA WITH SPRING FRAMEWORK¹⁵

Teodor Todorov – MSc Student

Department of Computer Systems and Technologies

University of Ruse “Angel Kanchev”

Tel.: +359 89 537 7807

E-mail: tazzmanian@abv.bg

Tsvetelina Mladenova– MSc Student

Department of Computer Systems and Technologies

University of Ruse “Angel Kanchev”

Tel.: +359 88 429 2155

E-mail: cvpmladenova@gmail.com

Assist. Prof. Yordan Kalmukov, PhD

Department of Computer Systems and Technologies

University of Ruse “Angel Kanchev”

Phone: +359 82 888827

E-mail: jkalmukov@ecs.uni-ruse.bg

Abstract: The paper gives an insight of the most in-demand programming languages of 2018. The language in topic is also the top language chosen by several studies. The paper gives a briefly explanation over some of the most important and useful features of the language and presents an example of a framework. The framework, described here is one that's being used by many programmers. Some of its key features are also shown and an example of a working solution is being given. The paper concludes with the example of a web-based system for bank. The example itself is a complicated process and the proposed framework does a great job in simplifying it.

Keywords: Programming, JAVA, SPRING Framework, simplifying

ВЪВЕДЕНИЕ

Разнообразието от езици и варианти за разработка на приложения, били те уеб, десктоп или мобилни, е огромно. Един от най-трудните избори пред всеки млад програмист е именно избора на език, среда, библиотека за програмиране. Разбирайки, че съществуват буквално стотици езици за програмиране, програмистите са поставени преди трудната задача да разгледат всеки език и да вземат под внимание неговите предимства и недостатъци.

ИЗБОР НА ЕЗИК ЗА ПРОГРАМИРАНЕ

Най-популярните езици

Според проучване на ТЮБЕ (freeCodeCamp, 2018) 5-те най-популярни езика за програмиране за 2018г. са JAVA (14,215%), C (11,037%), C++ (5,603%), Python (4,678%) и C# (3,754%). Изхождайки от текущото проучване, както и проучвания на същата организация за предходни години, се забелязва тенденция в нарастването на нуждата от програмисти за езика JAVA.

JAVA като предпочитан език за програмиране

Не случайно много програмисти избират JAVA за тяхна специализация. Езикът е известен със своята преносимост, давайки възможност за еднократно компилиране на кода и

¹⁵ Докладът е представен на студентската научна сесия на 27 април 2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: РАЗРАБОТВАНЕ НА УЕБ УСЛУГИ С JAVA И SPRING FRAMEWORK

многократното му инсталиране и използване на всяка JAVA-поддържаща платформа, без да са необходими никакви модификации или допълнителни конфигурации. Тази му гъвкавост е сред основните предимства на езика, което го прави основен избор за много програмисти.

Голямото разнообразие от поддържани външни библиотеки е друго предимство на езика, наред с функции като Garbage Collector и високата степен на защита.

Езикът поддържа множество външни библиотеки, което го прави по-лесен за употреба. Много често програмистите се насочват към JAVA и избират библиотека, която да научат и използват.

SPRING FRAMEWORK

SPRING Framework е външна библиотека, целяща да ускори процеса на проектиране и изпълнение на проекта. Лесната конфигурация в съчетание с множеството готови решения и заготовки я правят предпочитана.

Приложенията, написани на JAVA и SPRING библиотеката могат да бъдат практически от всякакво естество. С цел демонстрация е разработена Уеб базирана система за банкови преводи.

В случая употребата на библиотеката води до значително по-лесна и бърза конфигурация на базата от данни, както и много по-опростен и изчистен интерфейс на CRUD заявките. Това от своя страна води до по-ефективна обработка на заявките и отговорите между отделните системи, към която беше свързано текущо описаното приложение.

Може би най-голямото предимство на приложението, разработено на JAVA, е възможността да успее да си комуникира с други приложения, без значение под каква среда са разработени те. Поддържа HTML, SQL, XML и други унифицирани компоненти за предаване на данни, с което правят библиотеката и езика много по-гъвкави и ефективни спрямо други езици за програмиране.

РЕАЛИЗАЦИЯ НА УЕБ УСЛУГИ НА JAVA С ПОМОЩТА НА SPRING FRAMEWORK

Създаване на проект

Създаването на Spring проект става с няколко стъпки от <https://start.spring.io/> или с плъгин в средата за разработване. По-сложно е, ако се започне като обикновен Maven проект. Maven е Java инструмент за улесняване процеса на построяване на проект, осигурява унифицирана система за построяване, осигурява качествена информация за проекта. При създаването на проекта се добавят нужните библиотеки, като не пречи да се добавят и в последствие. Такива са библиотеките за работа с бази от данни, уеб услуги, JPA, сървъри и т.н.

С добавяне на библиотеките spring-boot-starter-data-jpa, spring-boot-starter-web, mysql-connector-java, spring-webmvc и spring-data-jpa разполагаме с готов вграден сървър и напълно готово приложение, използващо MySQL.

Новото приложение има един основен клас, който е с анотация @SpringBootApplication. Анотацията е метадата, която може да бъде добавена към кода. Класове, методи, променливи, параметри и пакети могат да бъдат анотирани. Те могат да бъдат вградени и да четат от файловете на класовете генерирани от компилатора.

Свързване на базата от данни с приложението

Spring Data JPA осигурява поддръжка за Java Persistence API (JPA). Това е най-лесният начин за разработване на приложение, нуждаещо се от JPA.

В конфигурационния файл application.properties за конфигуриране на типа на базата и данните за вход, се използват следните настройки:

```
spring.datasource.url = <път до схемата на базата от данни>
(jdbc:mysql://localhost:3306/mibank?useSSL=false)
```

```
spring.datasource.username = <посочва се потребителско име за достъп до СУБД>
spring.datasource.password = <посочва се парола за достъп до СУБД>
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
```

Използване на базата от данни

Анотация `@Entity` прави връзка между класа и таблица.

Анотацията `@Repository` отбелязва, че класа е “Repository”, оригинално дефинирано като Domain-Driven Design - механизъм за капсулиращо съхранение, коригиране и търсене, симулиращо колекция от обекти. `JpaRepository` интерфейса добавя основни CRUD операции на репозиторието за специфичен тип.

Например `AccountRepository` е референция към класа `Account`, който има анотация `@Entity`, така това репозитори ясно принадлежи на Spring Data JPA.

```
@Entity @Data public class Account {}
```

```
@Repository public interface AccountRepository extends JpaRepository<Account, String> {}
```

Бизнес логика

Анотацията `@Service` отбелязва класа като услуга, дефиницията, за която е следната: „операция, предлагана като интерфейс, която стои сама в модела, без капсулирано състояние.“

```
@Service public class AccountService
```

Интерпретиране на заявките

Анотацията `@RestController`, която вътрешно е аотирана с `@Controller` и `@ResponseBody`, интерпретира заявките. Типове, които имат тази анотация са третиранни като контролери, където `@RequestMapping` методите предполагат `@ResponseBody` семантика по подразбиране.

```
@RestController public class AccountController {
    @GetMapping("/api/accounts/{iban}") public AccountExistDTO exist(@PathVariable
String iban) {...}
    @PostMapping("/api/payments") public PaymentStatusDTO transferMonery(
@RequestBody PaymentDTO body) {...}
}
```

Анотациите `@GetMapping` и `@PostMapping` са аотирани сами по себе си със `@RequestMapping`. Те осигуряват връзка между уеб заявките на методи в класовете за управление на заявките. За основен параметър приемат URL, с който се свързва метода. В пътя им могат да се слагат променливи, в които името на променливата се загражда в скоби `{}`. За да се ползва в метода, е нужно да се подаде параметър, еднакъв с името на променливата. За да работи параметъра, е нужна `@PathVariable` анотацията. При POST заявките, където имаме тяло, свързването става благодарение на `@RequestBody` анотацията със зададения тип. Връщаният резултат на тези методи се връзва с отговора на заявката благодарение на `@ResponseBody` анотацията.

Достъп от други домейни

Когато заявките ще се изпълняват от домейн различен от оригиналния се налага добавяне на `@CrossOrigin` анотацията, която като параметър „origins“ приема масив от домейни, а означението „*“ означава всички домейни. Може да се поставя на класове и методи.

```
@RestController @CrossOrigin(origins = "*") public class AccountController {...}
```

Основните класове и методи, използвани за обработката на данните

`@Entity @Data public class Account` - съдържа атрибути на сметката. Обвързва се с атрибути от базата от данни.

@RestController @CrossOrigin(origins = "*") public class AccountController – MVC

@GetMapping("/api/accounts/{iban}") public AccountExistDTO exist(@PathVariable String iban) – проверка дали съществува дадена сметка.

@GetMapping("/api/accounts") public List<AccountDTO> getAll() – извличане на списък с всички сметки

@Data public class AccountDTO – data transfer object – структура на сметката

@Data public class AccountExistDTO – проверка за съществуване на сметка

@Repository public interface AccountRepository extends JpaRepository<Account, String> - интерфейс, осигуряващ CRUD на таблицата Account

public List<Account> findByClient(Client client); - абстрактен метод за търсене по клиент в таблицата Account

@Service public class AccountService – използва се за бизнес логиката между контролера и другите компоненти

public AccountExistDTO findIban(String iban) – метод, връщащ съществуването или отсъствието на сметка

public List<AccountDTO> findAll() – метод, връщащ всички сметки

private AccountDTO convertToDTO(Account acc) – вътрешен метод за конвертиране на типовете

@Entity @Data public class Client – осигурява връзката обект клиент с таблицата

@Repository public interface ClientRepository extends JpaRepository<Client, Long> интерфейс, осигуряващ CRUD

@Service public class ClientService – използва се за бизнес логиката между контролера и другите компоненти

@Data public class CentralBankDTO – очаквания обект, върнат от централната банка

@Entity @Data public class Payment – осигурява връзката обект плащане с таблицата

@Data public class PaymentDTO – структура за връщане на извършени плащания

@Data public class PaymentStatusDTO – структура на връщания резултат при грешка

@RestController @CrossOrigin(origins = "*") public class PaymentsController – контролер за плащанията

@GetMapping("/api/payments") public List<PaymentDTO> getAllPayments() – метод, връщащ всички плащания

@GetMapping("/api/payments/{param}") public List<PaymentDTO> getFilteredPayments(@PathVariable String param) – метод, връщащ плащанията по IBAN или клиентски номер

@PostMapping("/api/payments") public PaymentStatusDTO transferMonery(@RequestBody PaymentDTO body) – метод, приемащ форма за плащане и извършване на транзакции

@Repository public interface PaymentsRepository extends JpaRepository<Payment, Long> - интерфейс, осигуряващ CRUD

List<Payment> findByIbanResOrIbanSnd(String param1, String param2) – абстрактен метод, търсещ по IBAN на получател или изпращач

List<Payment> findByIbanResInOrIbanSndIn(List<String> param1, List<String> param2) – абстрактен метод, търсещ по списък от IBAN на получател или изпращач

@Service public class PaymentsService – използва се за бизнес логиката между контролера и другите компоненти

public List<PaymentDTO> getAll() – метод, връщащ всички плащания

private PaymentDTO paymentsToDto(Payment payment) – конвертира обекта за плащане до съответния му връщане

public List<PaymentDTO> getFilteredPayments(String param) – филтриране по IBAN или клиентски номер

public PaymentStatusDTO transfer(PaymentDTO body) – обработва заявката за плащане, валидиране

private void transferMoney(Account res, Account snd, PaymentDTO body) – извършва плащането.

ИЗВОДИ

Изграждането на приложение на Spring е лесно дори за хора, които за първи път се сблъскват с REST. Има голям набор от библиотеки, които благодарение на Spring Boot автоматично се конфигурират за съответната версия, когато е възможно с цел съвместимост на отделните зависимости. При нужда от допълнително конфигуриране, нещата не винаги стоят толкова прости, но това е в много крайни случаи.

БЛАГОДАРНОСТИ

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Petkov, A. (2018), *Here are the best programming languages to learn*, URL: <https://medium.freecodecamp.org/best-programming-languages-to-learn-in-2018-ultimate-guide-bfc93e615b35>

Java Annotation. URL: <https://github.com/web3j/web3j> (Accessed on 19.05.2018).

ANDROID MOBILE APPLICATION FOR MANAGING ACCOUNTS WITH ERC20 CRYPTOCURRENCY¹⁶

Teodor Todorov - MSc Student

Department of Computer Systems and Technologies,

University of Ruse "Angel Kanchev"

Tel.: +359 895377807

E-mail: banskaliqtabg@gmail.com

Ivaylo Rusev – MSc

Senior Java Developer,

Markovski Solutions Ltd.,

Tel.: +359 899183180

E-mail: ivaylo.rusev@markovskisolutions.com

Abstract: The paper reviews how ERC20 wallet works using Web3j library on Android platform. The application needs to be connected to node in a blockchain network where the node is either remote or on the local machine. The paper consist Web3j documentation and how the components are connected in the application, and how transfers are made.

Keywords: ERC20, Wallet, Web3j, Android, Blockchain network, Node

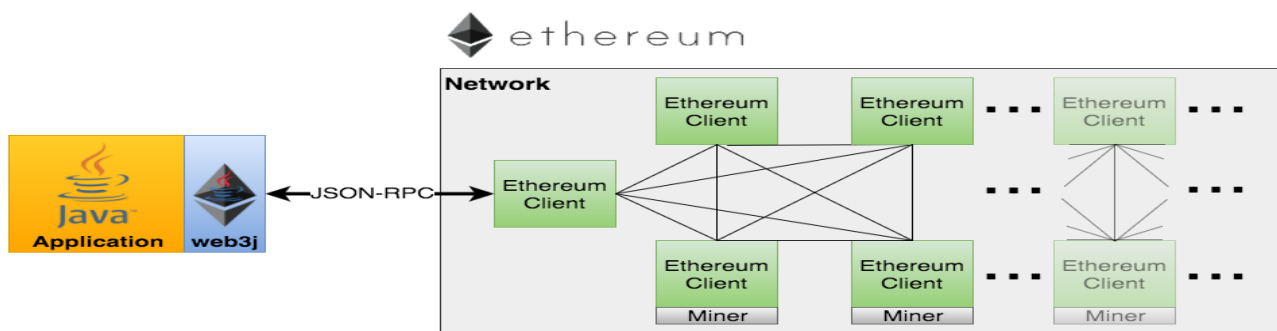
ВЪВЕДЕНИЕ

Повишаването на интереса към криптовалутите, токени и желанието да се търгува с тях във всеки един момент, налага ползване не само на настолни приложения, но и на мобилни. Това променя начина на свързване на приложението към мрежата, а именно чрез отдалечен достъп до възел.

ERC20 стандарта унифицира боравенето с токени, което означава, че едно приложение може да се използва за много токени. Приложението служи за управление на блокчейн сметки.

ИЗЛОЖЕНИЕ

Свързване на приложението



Фиг. 1. Web3j мрежа (<https://github.com/web3j/web3j>)

¹⁶ Докладът е представен на студентската научна сесия на 27 април 2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: АНДРОИД МОБИЛНО ПРИЛОЖЕНИЕ ЗА УПРАВЛЕНИЕ НА АКАУНТИ ЗА КРИПТОВАЛУТА СЪС СТАНДАРТ ERC20

Свързване на приложението към възел на мрежата, с използване на Web3j библиотеката е показано нагледно на Фиг. 1. За Android устройства, се използва метода build на класа Web3jFactory, показан по-долу. Стойността по подразбиране на HttpService е <http://localhost:8545/>.

```
Web3j web3 = Web3jFactory.build(new HttpService());
```

Създаване на нови сметки

Създаването на нови сметки става с няколко метода, от които разглеждам само generateBip39Wallet на класа WalletUtils, другите генерират текстов файл, който се пази на устройството. При него се създава мнемонична фраза, която се криптира. С вече криптираната фраза се създава частен ключ за генериране на файл, представляващ ключа. Входни параметри са паролата за частния ключ и мястото на съхранение.

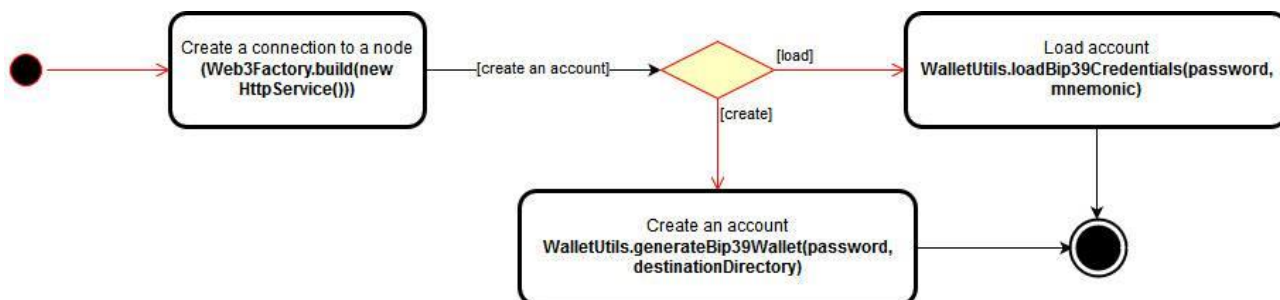
```
WalletUtils.generateBip39Wallet(password, destinationDirectory);
```

Зареждане на портфейл

Зареждането на сметка става с метода loadBip39Credentials на класа WalletUtils. За параметри метода приема паролата и мнемоничната фраза.

```
WalletUtils.loadBip39Credentials(password, mnemonic);
```

Логиката за задействане на зареждане или създаване е показана на Фиг. 2.



Фиг. 2. Диаграма на процеса на зареждане/създаване на сметка

Стандарт ERC20

Оперирането с договори става с методите, имплементирани в класа. Те са обвивка на JSON-RPC комуникация между клиента и възела. Изключват нуждата от кодиране и декодиране на обръщанията към функциите с ABI(Abstract Binary Interface).

ERC20 стандарта има следните методи и събития:

- totalSupply – общ брой предлагани токени;
- balanceOf(address _owner) const returns (uint256 balance) – взема баланса на подадения адрес;
- transfer(address _to, uint256 _value) returns (bool success) – изпраща определения брой токени към избрания адрес;
- transferFrom(address _from, address _to, uint256 _value) returns (bool success) – изпраща определения брой токени от единия адрес към другия;

- `approve(address _spender, uint256 _value)` returns (bool success) – разрешаване на някой да ползва средства от сметка до определена стойност;
- `allowance(address *_owner*, address *_spender*) constant` returns (uint256 remaining) – оставащата стойност, от която може да се тегли;
- `Transfer(address indexed _from, address indexed _to, uint256 _value)` – събитие, което се изпълнява след като токени са изпратени;
- `Approval(address indexed _owner, address indexed _spender, uint256 _value)` – събитие, което се изпълнява, когато се извика `approve`.

Операции с токени

Добавянето на токени към портфейла става с подаване на адреса на токена и имплементиране на методите му в клас, който унаследява `org.web3j.tx.Contract`. Пример за функция в класа е разгледан метода `balanceOf`.

```
Function function = new Function("balanceOf", Arrays.<Type>asList(_owner),
    Arrays.<TypeReference<?>>asList(new TypeReference<Uint256>() {}));
```

Като първи параметър е името на функцията, втори – параметрите, които приема, а третия - връщания тип.

Извикване на функцията става с метода `executeCallSingleValueReturnAsync(<име-на-функцията>)`, като този метод се използва само за четене от мрежата.

За зареждане на договора се използва метода `load()`, който прави инстанция на класа ни и подава параметрите на класа `Contract`.

```
load("0x<address>|<ensName>", web3j, credentials, GAS_PRICE, GAS_LIMIT);
```

ИЗВОДИ

Изграждането на портфейла с помощта на Web3j е сравнително лесно. Трудната част е да се гарантира сигурността на частния ключ, поради това се използва портфейл, на който личния ключ се получава от кодиране на произволни думи - seeds, с SHA256, а семената са получени от кодиране на мнемоничната фраза с паролата. Портфейлът може да се зареди без да се пази частния ключ, благодарение на мнемоничната фраза.

БЛАГОДАРНОСТИ

Този доклад се публикува със съдействието на проект 2018-ЕЕА-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Web3j documentation. URL: <https://docs.web3j.io/abi.html> (Accessed on 26.04.2018).

Web3j repository. URL: <https://github.com/web3j/web3j> (Accessed on 26.04.2018).

AN INTERACTIVE EDUCATIONAL TOOL FOR STUDYING THE HAMMING CODE ALGORITHM FOR ENCODING AND DECODING DATA USING MATRIX METHOD¹⁷

Ivan Todorov – Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 98 879 7226
E-mail: ivantodorov7226@gmail.com

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 855
E-mail: givanova@ecs.uni-ruse.bg

Abstract: *The paper discusses Hamming's data encoding algorithm. An interactive software that uses the matrix method for coding and decoding Hamming code is described. The software is designed to aid students and teachers during the learning process.*

Keywords: *Hamming code, encoding, decoding, coding theory, error detection, matrix method of Hamming code*

ВЪВЕДЕНИЕ

Съхранението и предаването на информацията е много лесно в наши дни (Jafarkhani, H., 2005). Но това е процес, който се е развивал векове на ред, с гениални разработки и открития (Richardson, T., & etc., 2008).

Историческото развитие на теорията на кодирането на данни датира от 1948 г. с изследванията на Клод Шанън (Van Lint, J. H., 2012). Той се фокусира върху проблема как най-добре да се кодира информацията. Шанън установява, че има фундаментална граница за компресиране на данни без загуба на качество и демонстрира, че е възможно да се кодират съобщения по такъв начин, че броят на допълнително предаваните битове да е възможно най-малък (Verdu, S., 1998).

Две години след Шанън, Ричард Хеминг (Hamming, R., 1980) изучава съществуващите до тогава кодове за кодиране. Той експериментира с кодовете по четност и нечетност, които могат само да откриват грешка. Хеминг създава код, който не само открива, но и посочва точното място на грешката и може да я коригира. Този код е известен в теорията на кодирането като Код на Хеминг.

Изучаване на различните методи за кодиране и декодиране на данни е включено в учебната програма на компютърните инженери. Методите за кодиране и декодиране са с различна сложност и трябва да бъдат прецизно пояснени и симулирани на студентите, за да могат да ги прилагат. Целта на настоящия доклад е да се представи един подход за проектиране и реализиране на интерактивен обучителен софтуер за решаване на задачи за код на Хеминг. Обучителният софтуер трябва да подпомогне студентите в усвояването на учебния материал по темата и да ги улесни в последователното проследяване на алгоритъма за кодиране и откриване на единични и двойни грешки по матричен метод на код на Хеминг.

¹⁷ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ИНТЕРАКТИВЕН СОФТУЕР ЗА ИЗУЧАВАНЕ НА АЛГОРИТЪМА ЗА КОДИРАНЕ НА ДАННИ ПО МАТРИЧЕН МЕТОД В КОД НА ХЕМИНГ

КОД НА ХЕМИНГ

Кодът на Хеминг е систематичен код, който открива двойни и коригира единични грешки. Контролните разряди на кода са разположени на точно определени места в закодираната кодова комбинация – степените на двойката, съответно на 1, 2, 4, 8, 16 и т.н. позиции. Контролните разряди контролират по четност определена група от информационните разряди. На Фиг. 1 са представени позициите и групите на контролните разряди в код на Хеминг (Smrikarova, S. & etc. 2013).

Номера на позициите	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Цифри на кода	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	a ₉	a ₁₀	a ₁₁	a ₁₂	a ₁₃	a ₁₄	a ₁₅
КР – C ₁	C ₁														
КР – C ₂		C ₂													
КР – C ₃				C ₃											
КР – C ₄								C ₄							

Фиг. 1. Позиции и групи на контролните разряди в код на Хеминг

Броят на контролните разряди се определя с формулата:

$$k \geq \log_2(n+1), \quad (1)$$

където „n“ е броят на информационните разряди в кодовата комбинация, а k – броят на контролните разряди.

За кодиране на кода на Хеминг се използват два метода: обикновен и матричен.

МАТРИЧЕН МЕТОД ЗА КОДИРАНЕ НА КОД НА ХЕМИНГ

При матричния метод за изчисляване на контролните разряди се строи контролна матрица H. Пример за такава матрица е представен на Фиг. 2.

$$H = \begin{pmatrix} \begin{matrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} & s_{11} \end{matrix} & \begin{matrix} C_4 & C_3 & C_2 & C_1 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix} & \begin{matrix} 1 & 0 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix} & \begin{matrix} 0 & 1 & 0 & 0 \end{matrix} \\ \begin{matrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{matrix} & \begin{matrix} 0 & 0 & 1 & 0 \end{matrix} \\ \begin{matrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{matrix} & \begin{matrix} 0 & 0 & 0 & 1 \end{matrix} \end{pmatrix}$$

Фиг. 2. Контролна матрица на Код на Хеминг

Контролните разряди се изчисляват на база на уравненията за кодиране (Фиг. 3), които се определят от контролната матрица. За декодиране се използват аналогични уравнения, но в тях участват и се проверяват и контролните разряди.

$$\begin{cases} C_4 = s_5 \oplus s_6 \oplus s_7 \oplus s_8 \oplus s_9 \oplus s_{10} \oplus s_{11} \\ C_3 = s_2 \oplus s_3 \oplus s_4 \oplus s_8 \oplus s_9 \oplus s_{10} \oplus s_{11} \\ C_2 = s_1 \oplus s_3 \oplus s_4 \oplus s_6 \oplus s_7 \oplus s_{10} \oplus s_{11} \\ C_1 = s_1 \oplus s_2 \oplus s_4 \oplus s_5 \oplus s_7 \oplus s_9 \oplus s_{11} \end{cases}$$

Фиг. 3. Уравнения за кодиране

ИЗИСКВАНИЯ КЪМ ИНТЕРАКТИВНИЯ СОФТУЕР ЗА КОДИРАНЕ И ДЕКОДИРАНЕ В КОД НА ХЕМИНГ С ИЗПОЛЗВАНЕ НА МАТРИЧЕН МЕТОД

Към проектирания обучителен софтуер са формулирани следните изисквания:

- Интуитивен съвременен интерфейс за работа;

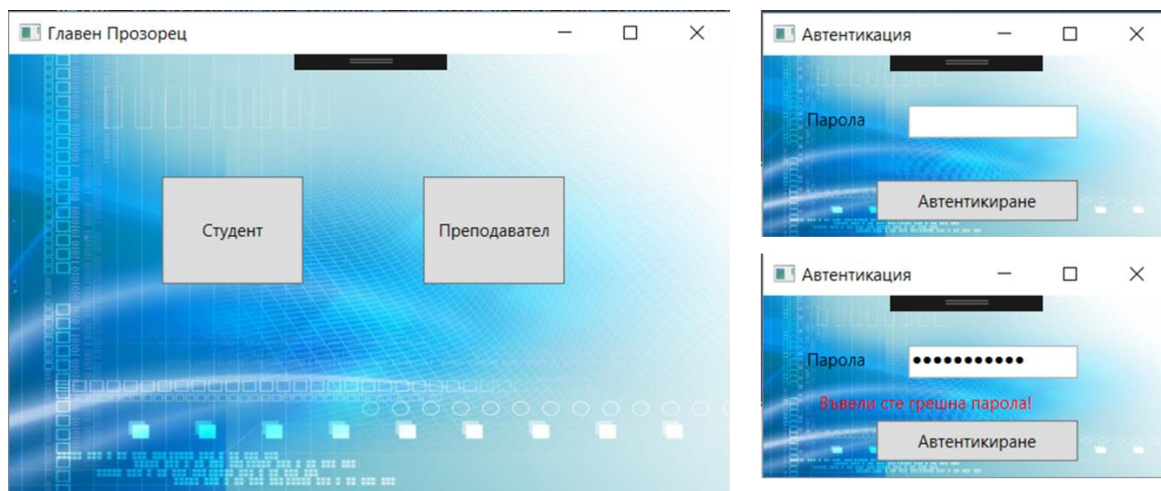
- Автоматизирано решаване на задачата в модула за преподавателя;
- Постъпково решаване на задачата за студенти, с отчитане на допуснатите грешки и време за работа;
- Предоставяне на възможност за кодиране на кодове с голяма дължина;
- Извеждане на точната позиция на грешката, при наличието на такава;
- Защита от неоторизиран достъп;
- Улесняване работата на преподавателите чрез предоставяне на средство за бърза и лесна проверка на задачи.

ПРОГРАМНИ ЕЗИЦИ И ИНСТРУМЕНТАЛНИ СРЕДСТВА

За разработката на приложението е избрана комбинацията от езикът за програмиране C# и графичната система Windows Presentation Foundation – WPF (Bourton, S., & etc, 2008). Езикът C# е широко използван и често предпочитан при изработване на приложения с графичен интерфейс, като предоставя възможност за бърза и лесна разработка с помощта на многото готови вградени функции, които позволяват на програмиста да се фокусира, не толкова върху синтаксиса и изучаването на възможностите на езика, а върху изграждането на нужната логика за решаване на дадения проблем. Системата WPF бе избрана пред други такива като Windows Forms поради значително намаления flicker при голям брой изобразени елементи на дисплея и улесненото му ползване. Графичните елементи са изградени с помощта на Extensible Application Markup Language (XAML). XAML дава възможност за използването на гъвкави елементи като DataGrid и прави използването им и тяхната визуална модификация много лесни както и от кода по време на изпълнение на приложението така и от инициализиращия код. Използвана е и средата за програмиране Microsoft Visual Studio 2017. За разработката беше възможно да се използва и текстов редактор, но използването на среда за програмиране, и в частност на Visual Studio, повишава многократно производителността на програмиста и намалява времето за разработка на приложението. Всичко това се намира под структурата .NET framework, която е предназначена за системи, използващи Microsoft Windows и не е оптимизирана за системи използващи iOS или Linux.

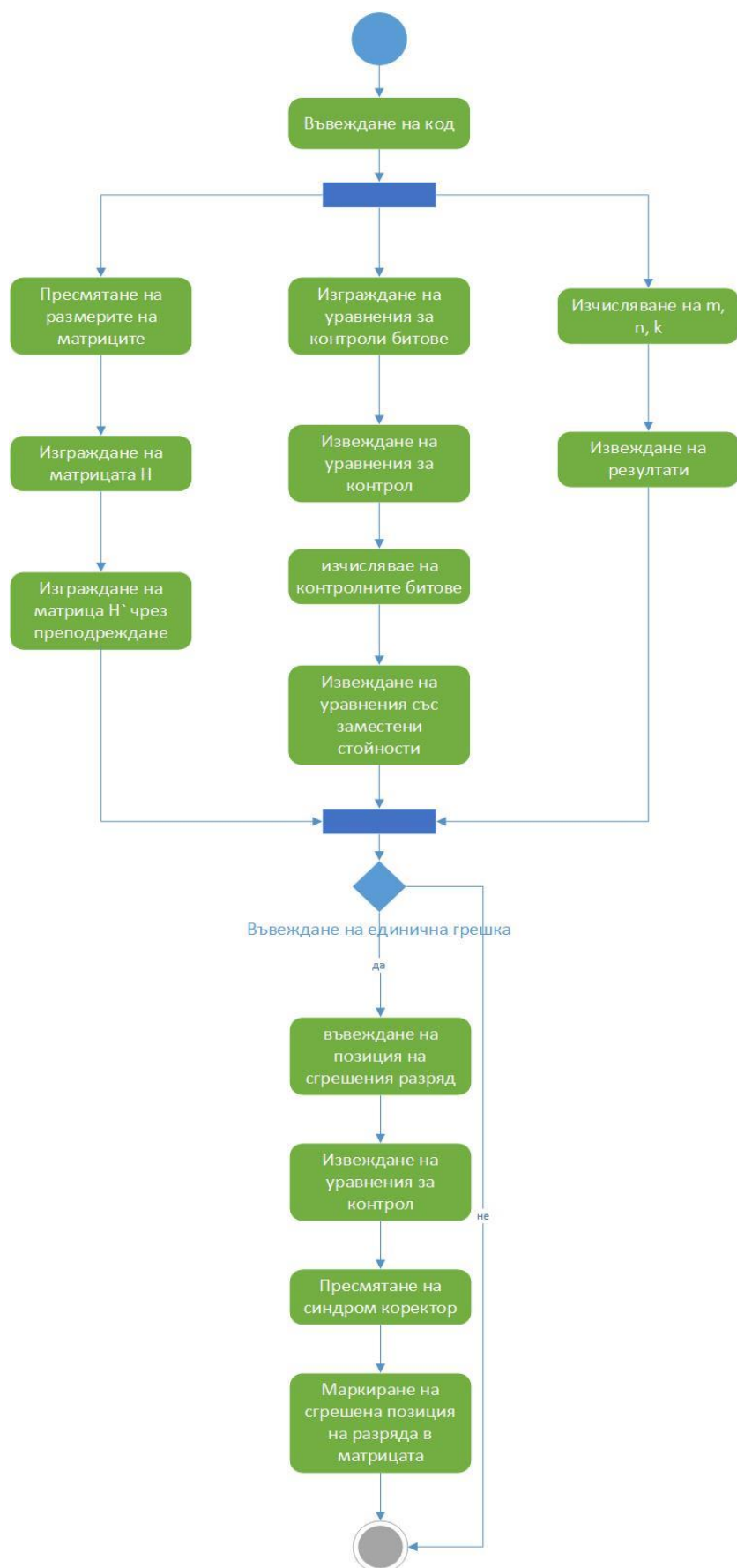
ФУНКЦИОНАЛНИ ВЪЗМОЖНОСТИ НА ПРИЛОЖЕНИЕТО

Обучителният софтуер е реализиран с два модула: модул за студенти и модул за преподаватели. На Фиг. 4 е представен началния екран на приложението, при който потребителят избира съответния модул. За работа с преподавателския модул се изисква автентикация чрез въвеждане на парола.



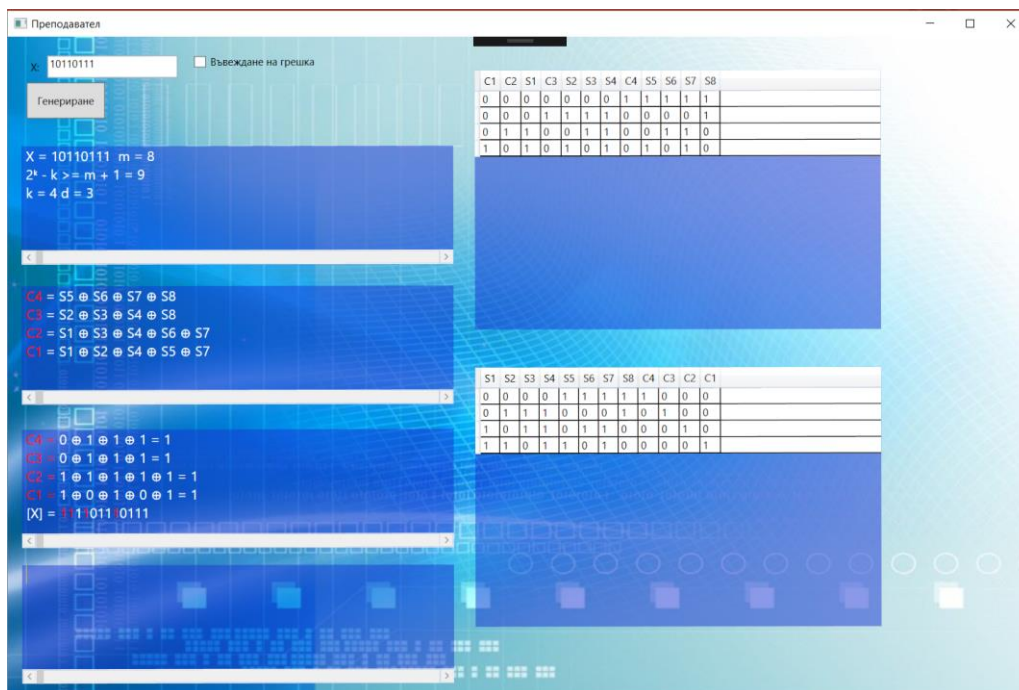
Фиг. 4. Начален екран и форми за автентикация

На Фиг. 5 са представени основните етапи в алгоритъма, който описва процеса на кодиране и откриване на единични грешки по матричен метод на код на Хеминг.



Фиг. 5. Алгоритъм за кодиране и откриване на единични грешки в код на Хеминг

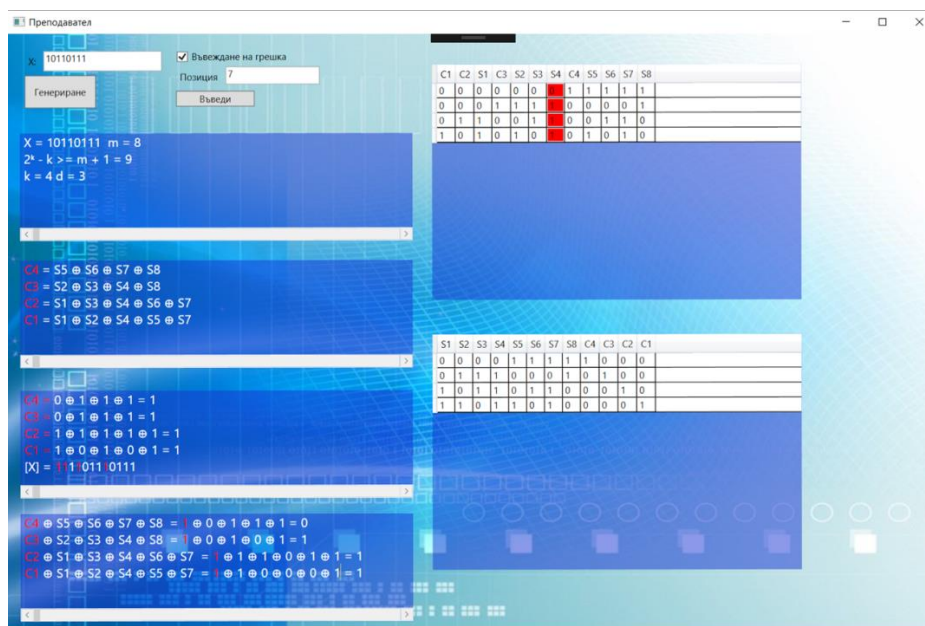
На Фиг. 6 е представено едно примерно решение на задача за кодиране в код на Хеминг с кодова комбинация от 8 бита. В преподавателския модул контролните матрици и формулите за изчисляване на контролните разряди се генерират автоматично.



Фиг. 6. Преподавателски модул

Интерактивният софтуер предоставя възможност на потребителя да въвежда позиция на грешката. На този етап софтуерът декодира единични грешки, но в бъдещо доразвиване на приложението се предвижда да се декодират и двойни грешки.

Кодът на Хеминг има способността да открива точната позиция на единичната грешка. При матричния метод откриването на позицията на грешката се реализира чрез намиране на стълб наречен Синдром, който се сравнява със стълбовете в контролната матрица. На Фиг. 7 в червен цвят е оцветен синдром, който открива единична греша в позиция S4.



Фиг. 7. Въвеждане и откриване на единична грешка

ИЗВОДИ

Представеният интерактивен образователен софтуер за кодиране и декодиране в код на Хеминг по матричен метод ще се използва в практическите упражнения по дисциплината

„Надеждност и диагностика на компютърни системи“ със студентите от специалност „Компютърни системи и технологии“.

Модулът за преподаватели ще подпомогне преподавателите по дисциплината в по-бързата и точна проверка на зададените задачи. Модулът за студентите проследява поетапно всички важни итерации за решаване на задачи с код на Хеминг – матричен метод. На всеки от етапите се проверяват знанията на студентите. При три грешно въведени отговора, студентът е насочен към верния отговор, за да успее да продължи на следващите етапи. Отчитането на грешките стимулира студентите към по-прецизна работа и стремеж към усъвършенстване. В приложението е предвидено и отчитане на времето за решаване, което въвежда и допълнителен състезателен елемент като предизвикателство за решаване на задачите.

С помощта на представения интерактивен софтуер ще могат да се решават и задачи с по-голяма разрядност на кодираната кодова информация. На този етап в упражненията по дисциплината се кодират и декодират кодови комбинации с максимална дължина от 15 бита. Благодарение на предложения в доклада програмен алгоритъм, ще може да се експериментира и да се проследяват и задачи с по-голяма разрядност, което ще допринесе до въвеждане на нови задачи в упражненията, по-близки до реалните задачи, които се използват за кодиране на данни в компютърните системи.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- Bourton, S., Jones, A., & Noble, S. (2008). WPF Recipes in C# 2008: A Problem-Solution Approach. Apress.
- Hamming, R. W. (1980). Coding and Theory. Prentice-Hall, Englewood Cliffs.
- Jafarkhani, H. (2005). Space-time coding: theory and practice. Cambridge university press.
- Richardson, T., & Urbanke, R. (2008). Modern coding theory. Cambridge university press.
- Smrikarova, S., & G. Ivanova (2013). Lecture notes for the course „Reliability and Diagnostics of Computer Systems“, URL: <https://e-learning.uni-ruse.bg> (Accessed on 05.04.2018)
- Van Lint, J. H. (2012). Introduction to coding theory (Vol. 86). Springer Science & Business Media.
- Verdu, S. (1998). Fifty years of Shannon theory. IEEE Transactions on information theory, 44(6), 2057-2078.

AN INTERACTIVE SOFTWARE FOR STUDYING THE ALGORITHM FOR DATA ENCODING USING CYCLIC CODE¹⁸

Velizar Todorov, BSc – MSc Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 82 888 855
E-mail: veltodorov@mail.bg

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 855
E-mail: givanova@ecs.uni-ruse.bg

Abstract: *Cyclic codes have been widely used in data transmission technologies, broadcast systems and computer applications. They have efficient encoding and decoding algorithms for detecting packet of errors. The aim of this paper is to present an interactive educational software designed to help computer engineers student to learn the algorithm for data encoding in cyclic code.*

Keywords: *Cyclic code, encoding, decoding, coding theory, error detection*

ВЪВЕДЕНИЕ

Надеждното кодиране, предаване и съхраняване на данни в наши дни е важен въпрос, с който се занимава теорията на кодиране (MacWilliams, F., & etc., 1977). Цикличните кодове (Cyclic Redundancy Check) намират широко приложение в практиката при предаване и съхраняване на данни (Mukhtar, H. & etc. 2016).

Цикличният код (Cyclic Redundancy Code) е често използван код при дигиталните системи поради сравнително лесната си имплементация и високата си ефективност (Коорман, Р. & etc., 2015). Той се характеризира с висока точност при откриване на грешки. Намира приложение при кодиране на данни за твърди дискове и DVD устройства; при стандартите за кодиране на видео; някои от технологиите за предаване на данни; при системите за разпространяване на данни и други (Тамо, I., & etc. 2015).

Целта на настоящия доклад е да се представи интерактивен обучителен софтуер, който подпомага решаването на задачи за Цикличен код в упражненията по дисциплината „Надеждност и диагностика на компютърни системи“ на студентите в специалността „Компютърни системи и технологии“ на Русенски университет „Ангел Кънчев“

ЦИКЛИЧЕН КОД

Цикличният код е вид систематичен код (Peterson, W. & etc., 1961), чието основно свойство се състои в това, че всяка комбинация, получена чрез циклично преместване на разрядите в една разрешена комбинация, също е разрешена (Smrikarova, S. & etc. 2013).

При построяване на цикличните кодове е възприето кодовата комбинация да се представя чрез полиноми спрямо някаква променлива, примерно x :

$$G(x) = a_{n-1} \cdot x_{n-1} + a_{n-2} \cdot x_{n-2} + \dots + a_1 \cdot x_1 + a_0, \quad (1)$$

¹⁸ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ИНТЕРАКТИВЕН СОФТУЕР ЗА ИЗУЧАВАНЕ НА АЛГОРИТЪМА ЗА КОДИРАНЕ НА ДАННИ В ЦИКЛИЧЕН КОД.

където $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ са коефициенти, приемащи стойности нула или единица (Smrikarova, S. & etc. 2013).

Например комбинацията 10101101 може да се запише под формата на полином

$$G(x) = x^7 + x^5 + x^3 + x^2 + 1, \quad (2)$$

където $G(x)$ е полином описващ кодовата комбинация, съдържаща n елемента (8 бита в конкретния пример).

Полиномът на закодираната комбинация се определя по следния начин:

$$F(x) = Q(x).P(x) = x^k.G(x) + R(x), \quad (3)$$

където:

$P(x)$ – пораждащ полином със степен k – броят на контролните разряди;

$Q(x)$ – частно;

$R(x)$ – остатък от делението на $x^k.G(x)$ на $P(x)$, (Smrikarova, S. & etc. 2013).

Пораждащият полином $P(x)$ се избира от таблица с неприводими полиноми, представена на Фиг. 1. Пораждащият полином $P(x)$ трябва да е прост, неприводим, т.е. да се дели без остатък само на единица и на себе си, (Smrikarova, S. & etc. 2013).

За откриване на двойна грешка, пораждащият полином трябва да е от степен k – броя на контролните разряди.

За откриване на тройна грешка, пораждащият полином $P(x)$ трябва да има вида:

$$P(x) = (1+x).P_1(x), \quad (4)$$

където полиномът $P_1(x)$ се избира от таблицата на Фиг. 1 така, че показателят му l да изпълнява условието $l \geq n$ (Smrikarova, S. & etc. 2013).

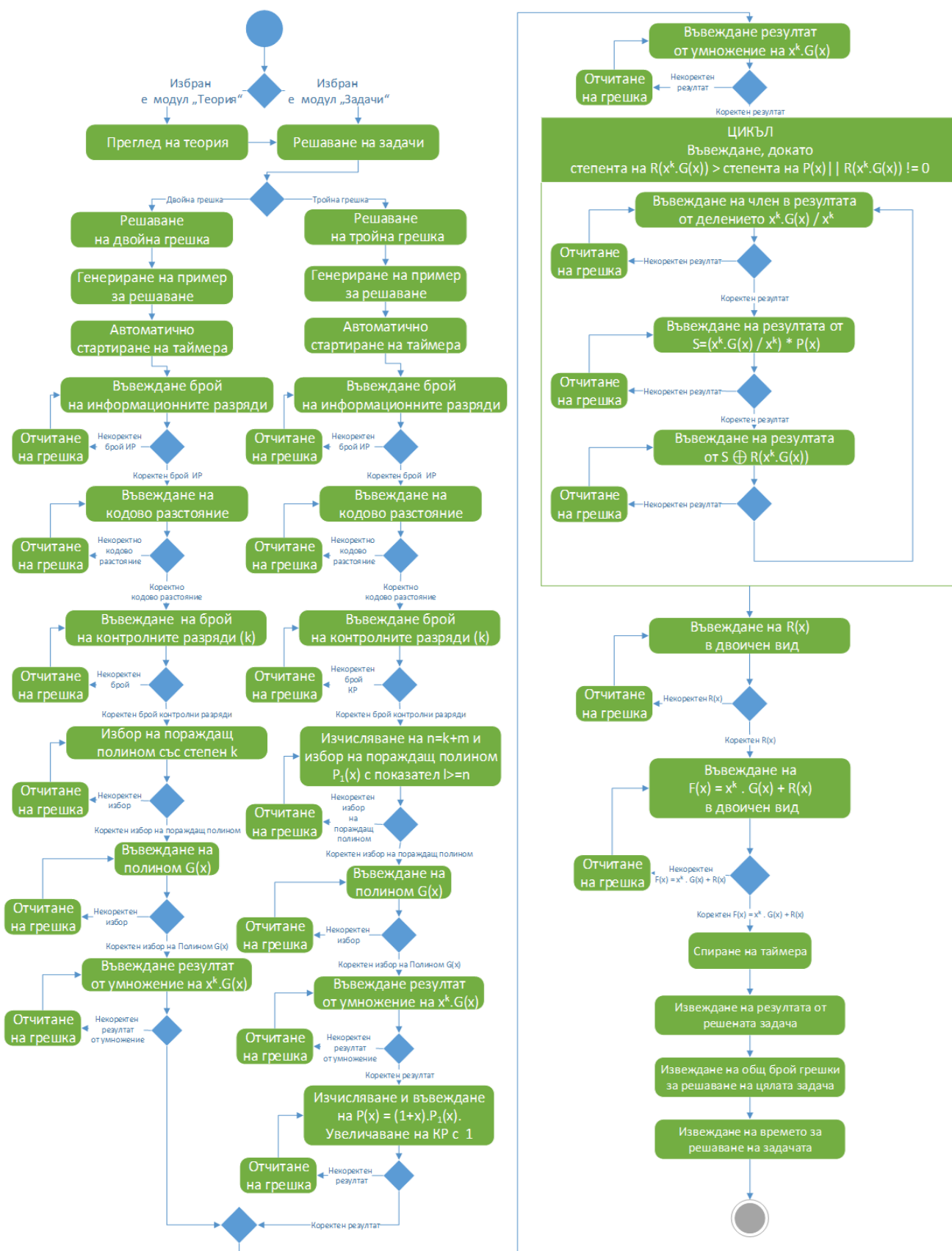
Таблица 1. Неприводими полиноми със степен от 1÷10

Степен на полинома s	Неприводим полином	Показател $l = 2^s - 1$
1	$x+1$	1
2	x^2+x+1	3
3	x^3+x+1 x^3+x^2+1	7
4	x^4+x+1 x^4+x^3+1	15
5	x^5+x^2+1 x^5+x^3+1	31
6	x^6+x+1 x^6+x^5+1	63
7	x^7+x+1 x^7+x^3+1	127
8	$x^8+x^4+x^3+x^2+1$ $x^8+x^5+x^6+x^4+1$	255
9	x^9+x^5+1 x^9+x^4+1	511
10	$x^{10}+x^7+1$ $x^{10}+x^3+1$	1023

Фиг. 1. Таблица с неприводими полиноми

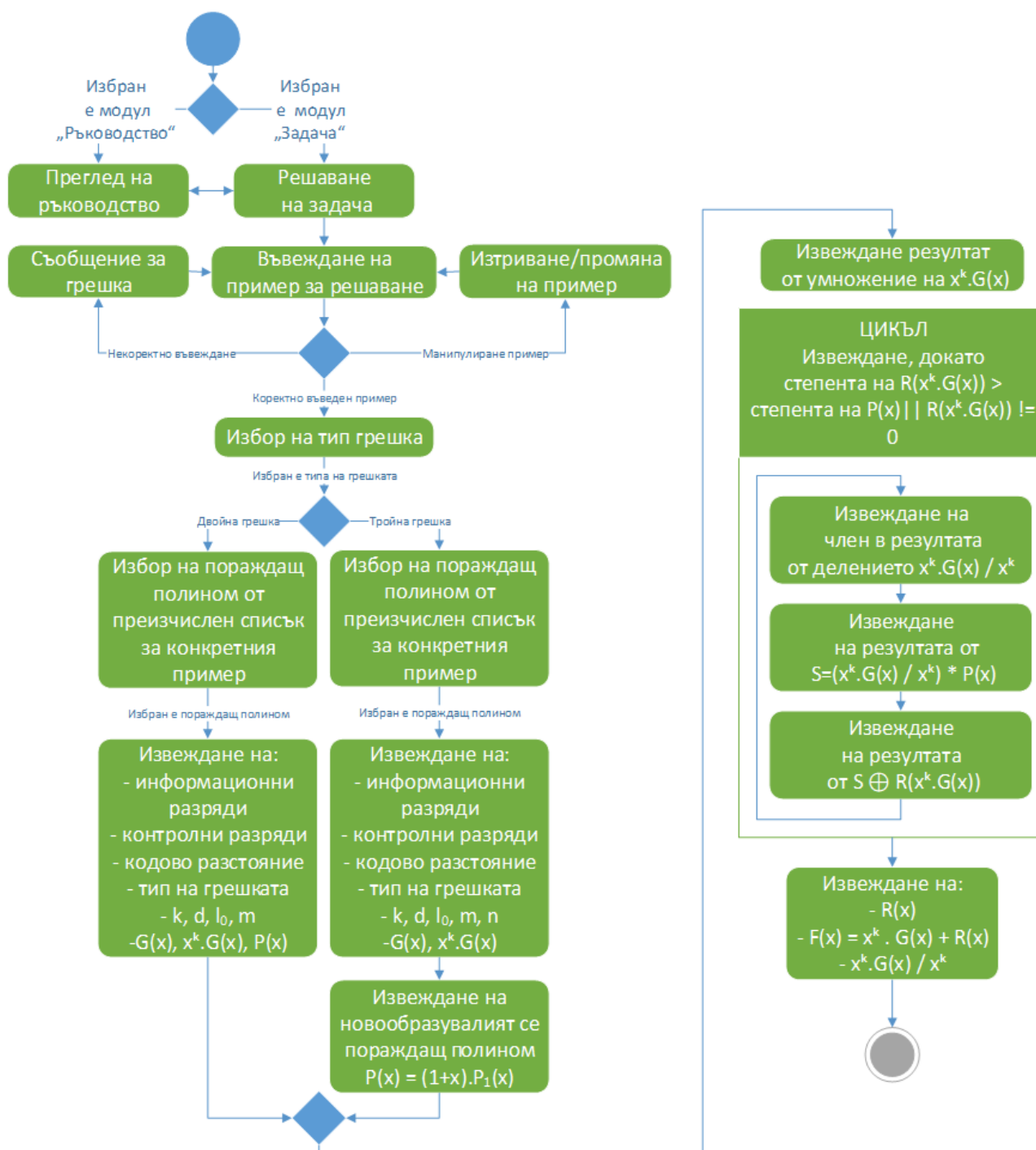
ФУНКЦИОНАЛНИ ВЪЗМОЖНОСТИ НА ИНТЕРАКТИВНИЯ СОФТУЕР

Интерактивният обучителен софтуер е проектиран с два модула: за преподаватели и студенти. На Фиг. 2 е представена диаграмата на дейности на студента.



Фиг. 2. Диаграма на дейности на студент

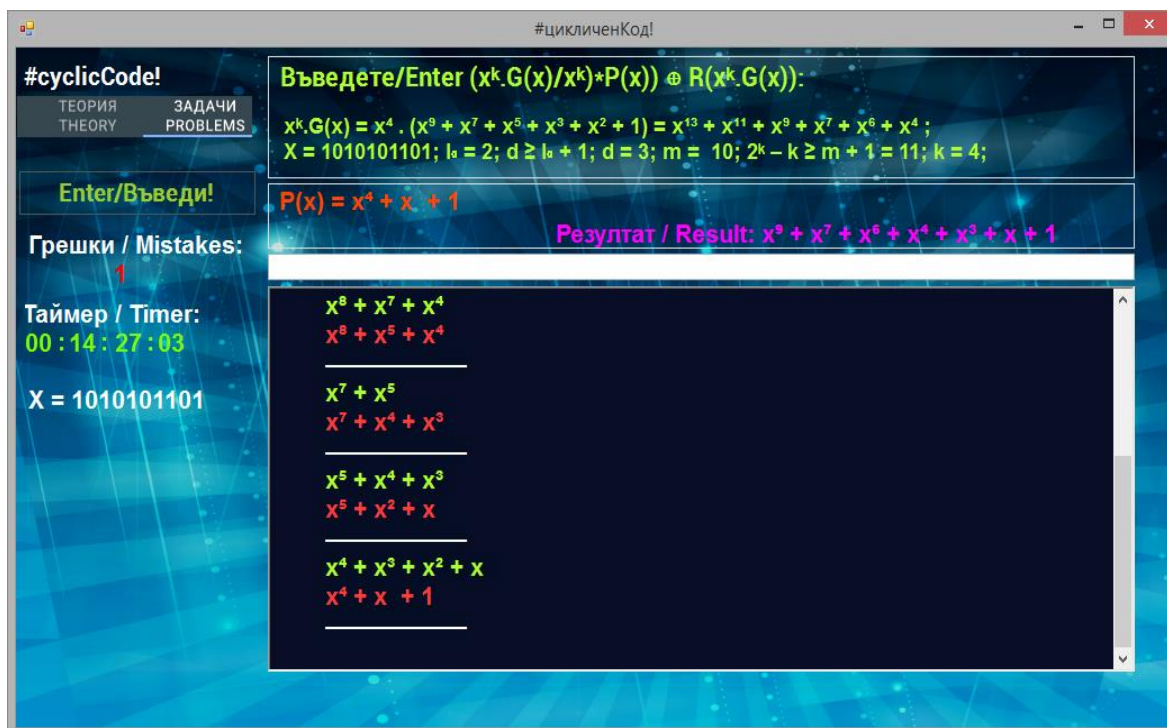
На Фиг. 3 е представена диаграмата на дейности на преподавател.



Фиг. 3. Диаграма на дейности на преподавател

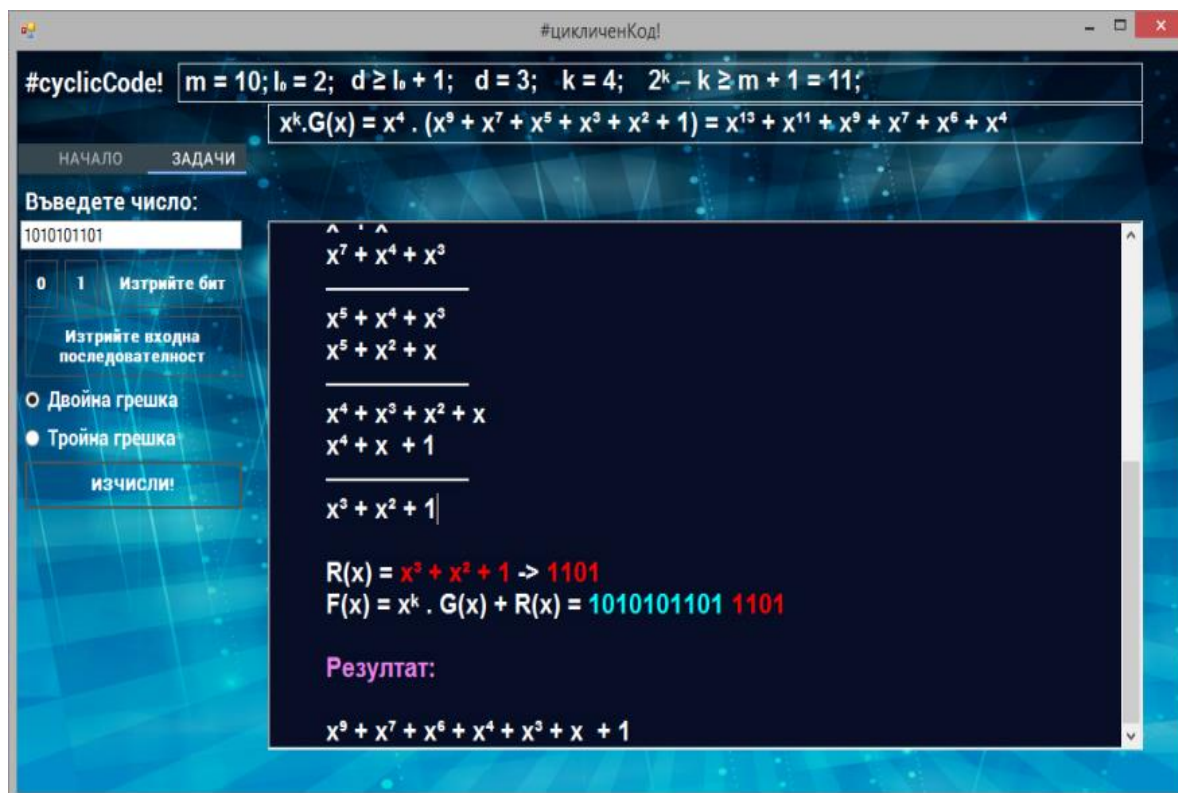
Модулът за преподаватели предоставя възможност за въвеждане на входната последователност, която може да се манипулира и изтрива както чрез клавиатурата, така и чрез бутоните в приложението. В модула за студент входната последователност се генерира на случаен принцип от предварително зададени примери в софтуера.

Модулът за студенти е проектиран с хибриден българско-английски интерфейс за обучение както на български, така и на чуждестранни студенти (Фиг. 4). На всеки етап от решаването на задачите, софтуерът очаква от студентът да въвежда коректните резултати за да се продължи към следващата итерация. Предоставена е възможност за избор на два типа грешка – двойна и тройна грешка, което е свързано и със съответни разлики в начините на избор на пораждателния полином и кодирането на входната последователност.



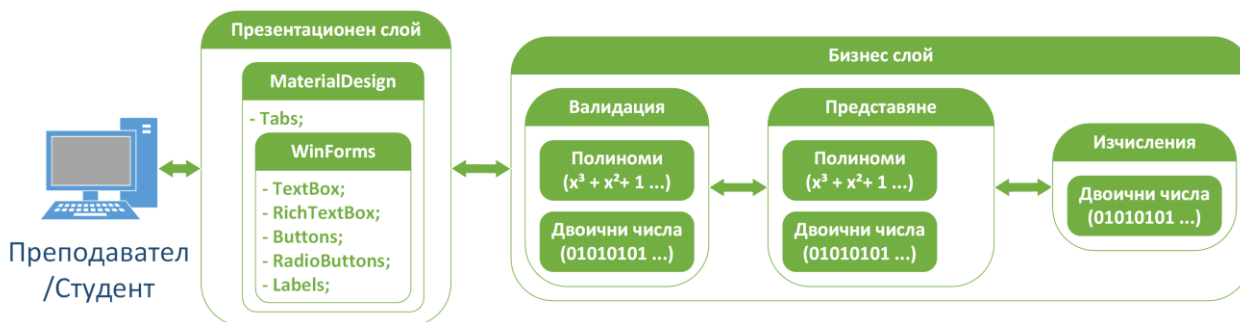
Фиг. 4. Интерфейс на модула за студенти

В модула за студенти се отчита времето на работа и допуснатите грешки за всяка от итерациите (Фиг. 5). За да се акцентира на важни моменти в алгоритъма е предвидена и подходяща по-ярка цветова гама при извеждането на важна информацията в заданието, както и в самите итерации при визуализирането на резултатите.



Фиг. 5. Пример за решена задача в модула за студенти

Архитектурата на приложението е представена на Фиг. 6.



Фиг. 6. Архитектура на приложението

ИЗВОДИ

Описаният интерактивен обучителен софтуер ще подпомогне студентите в процеса на изучаване и решаване на задачи с Цикличен код. Постъпковият подход на решаване на задачите и отчитането на грешки, с възможност за подсказване на верния отговор, ще даде възможност на студентите да проследяват и проверяват знанията си на всяка итерация от изучавания алгоритъм за кодиране на данни в Цикличен код. Описаното решение предоставя един съвременен интерактивен образователен инструмент за самоконтрол и самоподготовка на студентите с неограничен брой примери за решаване.

Модулът за преподаватели значително ще намали допълнителното технологично време за проверка и пререшаване на курсовите задачи за Цикличен код. Той ще се използва от преподавателите като прецизен и съвременен технологичен еталон за сравнение и бързо откриване на грешки и неточности в контролните работи или рефератите на студентите.

В упражненията за Цикличен код се кодират входни последователности с ограничена разрядност. За кодирането на по-дълги последователности са необходими много повече итерации и технологично време, с което не се разполага. Предложеният програмен алгоритъм за кодиране на данни в Цикличен код може да кодира входни последователности с големи разрядности. Това ще предостави възможност на преподавателите и студентите да експериментират в упражненията с по-големи по размер входни последователности.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- Koopman, P., Driscoll, K., & Hall, B. (2015). Selection of Cyclic Redundancy Code and Checksum Algorithms to Ensure Critical Data Integrity.
- MacWilliams, F. J., & Sloane, N. J. A. (1977). The theory of error-correcting codes. Elsevier.
- Mukhtar, H., Al-Dweik, A., & Shami, A. (2016). Turbo product codes: Applications, challenges, and future directions. IEEE Communications Surveys & Tutorials, 18(4), 3052-3069.
- Peterson, W. W., & Brown, D. T. (1961). Cyclic codes for error detection. Proceedings of the IRE, 49(1), 228-235.
- Smrikarova, S., & G. Ivanova (2013). Lecture notes for the course „Reliability and Diagnostics of Computer Systems“, URL: <https://e-learning.uni-ruse.bg> (Accessed on 05.04.2018)
- Tamo, I., Barg, A., Goparaju, S., & Calderbank, R. (2015). Cyclic LRC codes and their subfield subcodes. In Information Theory (ISIT), 2015 IEEE International Symposium on (pp. 1262-1266). IEEE.

AN ANDROID MOBILE APPLICATION FOR PLANNING AN EFFECTIVE CONTROL OF AUTOMOBILE SERVICE PERIODS ¹⁹

Aleksandar Dzhrekos - Student

Department of Computer Systems and Technologies,
University of Ruse "Angel Kanchev"
Tel.: +359 82 888 855
E-mail: aleksandardzhrekos@gmail.com

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,
University of Ruse "Angel Kanchev"
Phone: +359 82 888 855
E-mail: givanova@ecs.uni-ruse.bg

Abstract: *This paper discusses an opportunity for storing and tracking the car's deadlines using an android app. The system maintains effective work with deadlines fully in line with Bulgarian regulations. The software is designed for all Bulgarian automotive drivers using Android smartphones.*

Keywords: *car, android, service, control, taxes, planning periods*

ВЪВЕДЕНИЕ

XXI^в век е векът на технологиите. Свидетели сме на появата на нови открития, които бързо намират приложение в човешкото ежедневие и се внедряват в различни сфери на производството. Когато се говори за бързо развитие, не може да не се отчете темпът на развитие и разпространение на мобилните устройства.

Преди единадесет години на пазара е представен първият „модерен“ смартфон. През 2007 година се появява операционната система Андроид, която е базирана на Linux ядро и е с „отворен код“, (Smyth, N., 2017). През 2008 година се появява и първият Андроид базиран смартфон, (Aditya, S. K. & etc., 2014). Само три години по-късно Android е обявена за най-масово използваната операционна система при мобилните устройства, (SpinFold, 2016). Android операционната система се развива с много бързи темпове благодарение на големия брой софтуерни разработчици, които създават така наречените „apps“ (Applications) – игри и приложения, които разширяват функционалността на системата, (Mew, K., 2015). Приложенията могат да бъдат сваляни от различни сайтове в Интернет или от големи онлайн магазини като Android Market (впоследствие преименуван на Google Play) – магазинът на Google. По данни към януари 2011 г. за Android има над 200 000 приложения, а декември 2017 г. са над 3 500 000 приложения, (Statista, 2017).

От 2008 г. Android претърпява многобройни актуализации, които постепенно подобряват операционната система, добавят нови функции и поправят грешките в предишните версии. Всяка основна версия е наречена на десерт или сладкиш в азбучен ред - например версия „1.5 Cupcake (кекс)“ е последвана от „1.6 Donut (поничка)“. Последната „стабилна“ версия е „Android 8.1 Oreo“, (Schmidt, C., 2016).

Зависимостта, че развитието на софтуера се определя от развитието на хардуера и обратното, се наблюдава още от десктоп компютрите и важи с пълни сили и при мобилните устройства (смартфони и таблети). Само за десетина години на пазара вече се предлагат

¹⁹ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: АНДРОИД МОБИЛНО ПРИЛОЖЕНИЕ ЗА ПЛАНИРАНЕ НА ЕФЕКТИВЕН КОНТРОЛ НА СРОКОВЕТЕ ЗА ОБСЛУЖВАНЕ НА АВТОМОБИЛНАТА СИСТЕМА.

смартфони с големи хардуерни мощности и производителност (осем ядрени процесора с тактова честота 2.45GHZ+, 6-8 GB оперативна памет, вътрешна памет за съхранение 64-512 GB). Това позволява мобилните устройства да се използват за по-сложни задачи.

В условия на динамично и забързано ежедневие, ролята на технологиите е изключително важна в почти всички сфери на обществото. Голяма част от дейностите като въвеждане, обработка и съхраняване на данни се извършва от специализиран софтуер. Смартфоните постепенно изместват настолните компютри, като основните причини за това са значително по-компактните им размери, ергономичността и енергоефективността им.

В забързаното ежедневие на всеки човек, следенето на дати и спазването на срокове е проблем, който също трябва да бъде отчетен. В последните години нараства търсенето и предлагането на мобилните приложения, които могат да ни улеснят в проследяването на различни срокове и да ни подсказват ежедневни ангажименти и събития. Водачите на автомобилни превозни средства се сблъскват дори по-често с проблема за проследяване на срокове. Техническата изрядност на един автомобил зависи от редовното му обслужване, а документалната му изрядност - от редовното плащане на такси.

В доклада е описан един подход за проектиране и създаване на Андроид базирано мобилно приложение за планиране на ефективен контрол на сроковете за обслужване на автомобилната система. С реализирането и внедряването на приложението се цели да се подпомогнат ползвателите му в проследяването на сроковете за заплащане на такси и сроковете за обслужването на едно или няколко превозни средства.

ИЗИСКВАНИЯ КЪМ ПРИЛОЖЕНИЕТО

Направено е проучване на три потребителски мобилни приложения за автомобилни системи. В таблица 1 е представена сравнителна характеристика на тези приложения, като те са оценени по четири критерия с предложеното авторско мобилно приложение "Love My Car".

Таблица 1. Сравнителна таблица за оценяване и сравняване на съществуващите решения с предложеното авторско приложение

Възможности	Мобилни приложения			Авторско приложение "Love My Car"
	"aCar"	"My Car Service"	"Авто Поддръжка"	
Поддръжка на български език	НЕ	ДА	ДА	ДА
Спазване на срокове съобразени с българските наредби	НЕ	НЕ	НЕ	ДА
Сигурност на ниво потребител	НЕ	НЕ	НЕ	ДА
Backup на базата данни	НЕ	ДА	ДА	ДА

След направеното проучване и анализ на функционалните възможности на съществуващите решения са обобщени следните изисквания към разработеното мобилно приложение за проследяване на сроковете на автомобилната системата „LoveMyCar“ :

- Приложението трябва да притежава „приятелски“ и интуитивен интерфейс;
- Да се осигурява сигурност на ниво потребител;
- Да се предоставя възможност за добавяне на няколко превозни средства;
- Да се съхраняват данните на потребителя и превозните му средства;
- Да има възможност за архивиране (Backup) и експортиране/вмъкване (Export/Import) на базите от данни (данните на потребителя и автомобилите);
- Да се проследяват сроковете и да се сигнализира една седмица преди изтичането им чрез съответни нотификации в мобилния телефон;
- Да бъде осигурен лесен начин за преглед и редакция на въведените данни;
- Да се предвиди възможност за изтриване на избран автомобил.

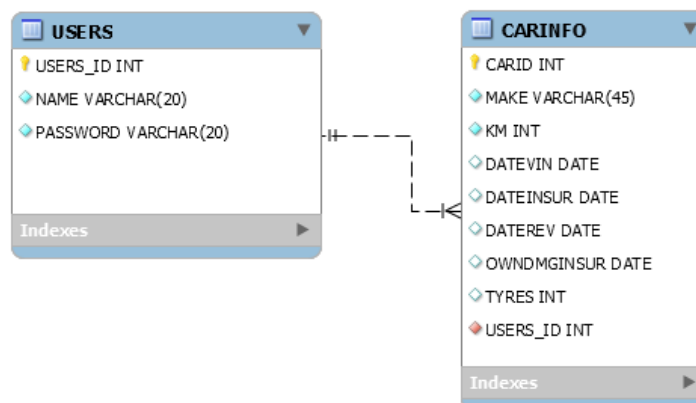
Езикът, на който е базирана средата, е Java. За изграждане на базата данни се използва SQLite, който е вграден и се използва чрез извикване на класа SQLiteDatabase, намиращ се в пакета android.database.sqlite. “SQLite е SQL база данни с отворен код, която съхранява данни в текстов файл на устройство”, (Vogel, L., 2010). За структуриране на дизайна се използва XML (eXtensible Markup Language).

На Фиг. 1 е представена диаграмата на дейностите на приложението. Поетапно е представен процесът на вписване/регистрация на потребител и процесът за добавяне/актуализиране на данни за автомобил.



- 93 -

потребителя и на превозните му средства. Таблица *“USERS”* съдържа името и паролата на потребителя, чрез които се осигурява сигурен достъп до данните на превозните му средства. Таблица *“CARINFO”* съдържа всички превозни средства и необходимата за тях информация. Двете таблици са свързани чрез уникално ID на потребителя и връзка „едно към много“.



Фиг. 2. Релационен модел на базата данни

След успешно вписване с потребителско име и парола, потребителят може да добавя нов автомобил в базата от данни, за който се съхраняват следните данни:

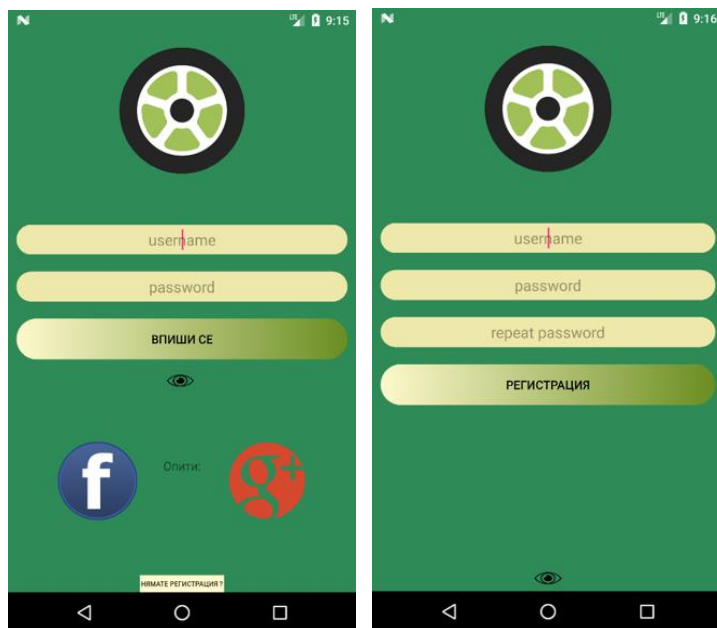
- Марка на автомобила;
- Километри на последна смяна на маслото на автомобила;
- Крайна дата на застраховката на автомобила;
- Крайна дата на винетката на автомобила;
- Крайна дата на прегледа на автомобила;
- Актуално състояние на типа гуми на автомобила (зимни/летни).

Потребителите имат възможност да добавят един или няколко на брой автомобила, за които приложението ще проследява крайните дати на сроковете, свързани със застраховката, винетката, прегледа или срока за смяна на маслото на автомобила. Една седмица преди да изтече даден срок, приложението генерира нотификация, която цели да подсети потребителя да реагира и предотврати нежелани последствия от неспазването на този срок. Приложението генерира и ежеседмична нотификация, която подсеща потребителя да въвежда периодично километрите на автомобила. На база на изчислените километри се проследява и срока за смяна на маслото на автомобила, изтичането на който също е последвано от съответна нотификация.

Допълнителните опции, с които разполага приложение, са:

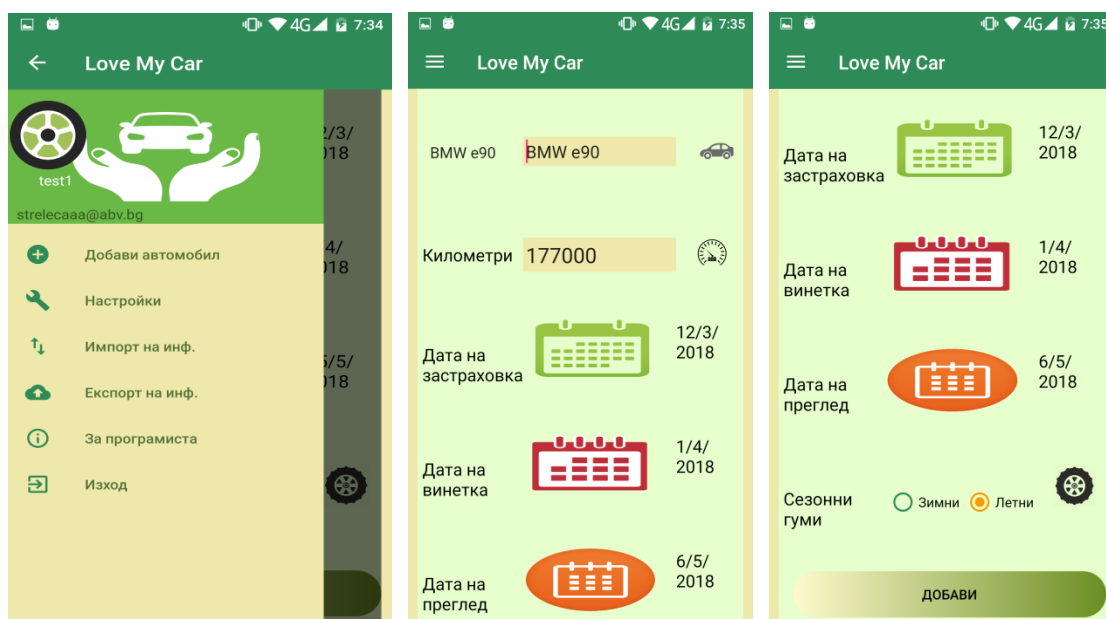
- Изтриване на избран автомобил;
- Импорт/Експорт на базата от данни;
- Меню с настройки за промяна на честота на известяване;
- Меню с настройки за промяна на темата на приложението;
- Информация за програмиста;
- Отписване и изход от приложението.

На Фиг. 3 са представени интерфейсни екрани за вписване и регистриране на потребителите.



Фиг. 3. Интерфейсни екрани за вписване и регистриране на потребители

Менюто на приложението и формата за попълване на данните за нов автомобил са представени на Фиг. 4. След успешно вписване в системата потребителите имат възможност да добавят данни за нов автомобил или да редактират съществуващи данни за вече въведени в базата данни автомобили.



Фиг. 4. Меню на приложението и форма за попълване на данни за нов автомобил

На Фиг. 5 е представена клас диаграмата на приложението, която се състои от четири основни класа. За всяка активност има отделен клас, като всички са наследници на класа *"AppCompatActivity"*. За работа с базата данни се използва клас *"Users_Database"*, който е наследник на класа *"SQLiteOpenHelper"*.



Фиг. 5. Клас диаграма за приложението

ИЗВОДИ

Еволюцията на човечеството и технологиите е неизбежен процес, който напредва с бързи темпове. В свят на дигитализация и мобилност е естествено да се доверяваме и да разчитаме на технологиите, за да ни помагат в ежедневните ни задължения.

Описаното мобилно приложение ще подпомогне автомобилните водачи в процеса за проследяване на важни срокове за превозните им средства, което ще допринесе и в частност до повишаване на изрядността на техните автомобилни системи.

Приложението е проектирано да работи на съвременните версии на Android на пазара. Направени са тестове на всички версии след „Андроид 4.4 Киткат“, вкл. работата по развитие на приложението продължава, като в бъдеще се предвижда да се разширят съществуващите и добавят нови функционалности.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- Aditya, S. K., & Karn, V. K. (2014). Android sQLite essentials. Packt Publishing Ltd.
- Mew, K. (2015). Android 5 Programming by Example. Packt Publishing Ltd.
- Schmidt, C. (2016). A Compete Guide for Beginners. What is Android?, URL: <https://www.androidpit.com/what-is-android> (Accessed on 12.04.2018).
- Smyth, N. (2017). Android Studio 3.0 Development Essentials-Android 8 Edition. Payload Media, Inc..
- SpinFold. (2016) History of first android phone. URL: www.spinfold.com/first-android-phone (Accessed on 12.04.2018).
- Statista. (2017). Android Statistics & Facts.URL: <https://www.statista.com/topics/876/android/> (Accessed on 12.04.2018).
- Vogel, L. (2010). Android sqlite database and contentprovider-tutorial. Java, Eclipse, Android and Web programming tutorials, 8.

AN EDUCATIONAL SOFTWARE FOR SOLVING TASKS TO CONTROL ARITHMETIC AND LOGICAL OPERATIONS²⁰

Tsvetomir Goranov, BSc – MSc Student

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Tel.: +359 82 888 855

E-mail: tsvento1994@abv.bg

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Phone: +359 82 888 855

E-mail: givanova@ecs.uni-ruse.bg

Abstract: *The problem with computer system reliability is extremely important today due to our dependence on technologies. The processor is the heart of the computer system. Its main unit is the arithmetic logic unit (ALU). Errors in the arithmetic and logical operations in a processor system lead to less reliability of computer system. The correctness of the results of the arithmetic and logical operations is important. The paper presents an educational software for solving tasks to control arithmetic and logical operations in ALU. The software is designed for computer engineer students to help them to learn important methods for control of these operations.*

Keywords: *arithmetic operations, logical operations, arithmetic logic unit, control, mod 3, computer reliability*

ВЪВЕДЕНИЕ

С всеки изминал ден се забелязва, че сме зависими все повече и повече от технологиите (Siewiorek, D., & etc., 2017). Приложението на компютърните технологии в почти всички области налага все по-високи изисквания, както към тяхната производителност, така и към надеждното им функциониране. Потребителят очаква от компютъра коректни резултати от обработката на информацията и гаранция за това е наличието на средства за контрол и диагностика на компютърните системи.

Процесорът е основен компонент на компютърната система, а аритметико логическото устройство е част от него, чиято основна функция е извършването на различни аритметични и логически операции с машинни думи - кодове на числа, части от инструкции и др. (Mano, M. M., & etc., 2008). Контролът върху аритметичните и логически операции (Rao T., 1970) е важен за правилното функциониране на компютърната система (Medwed, M., 2011). Проверката на тези операции се извършва програмно и апаратно (Patel, J. N., & etc., 1982).

В настоящия доклад е описан един подход за разработка на софтуерен продукт за решаване на задачи, свързани с контрола на аритметическите и логическите операции. С разработването му се цели да се създаде обучителен инструмент, който да улесни усвояването на учебния материал и да проследи постъпково методите за контрол на аритметически и логически операции. Софтуерът се използва със студентите по дисциплината „Надеждност и диагностика на компютърни системи“, които се обучават в образователно-квалификационна степен “Бакалавър” в специалност “Компютърни системи и технологии”.

КОНТРОЛ НА ЛОГИЧЕСКИ ОПЕРАЦИИ

Към логическите операции в компютърните системи се числят:

²⁰ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ОБУЧИТЕЛЕН СОФТУЕР ЗА РЕШАВАНЕ НА ЗАДАЧИ ЗА КОНТРОЛ НА АРИТМЕТИЧЕСКИ И ЛОГИЧЕСКИ ОПЕРАЦИИ.

- логическо събиране – ИЛИ ;
- логическо умножение – И ;
- поразрядно сравнение – $\oplus$;
- инвертиране ;
- изместване.

За контрол на логическите операции с използване на кодове не съществуват удобни и икономични методи, както при аритметичните операции. Причината е, че не могат да се подберат контролни операции така, че контролният признак на резултата да бъде еднозначно свързан с контролните признаци на операндите. Проверката се извършва програмно и апаратно, с използване на контрол по mod 3 и въвеждане на излишък (Smrikarova, S. & etc., 2013).

Най-голямо приложение са намерили следните два метода:

- повторно изпълнение на логическата операция;
- паралелно изпълнение на две логически операции.

При първия метод контролът се извършва в следния порядък:

- изпълнява се логическата операция;
- формират се контролните разряди на резултата по mod 3;
- изпълнява се логическата операция отново;
- формират се контролните разряди на втория резултат;
- сравняват се контролните разряди на резултатите от двете изпълнения.

При втория метод се изпълняват паралелно две логически операции, от получените резултати се формират контролните разряди по mod 3 и се сравняват по определени зависимости (Smrikarova, S. & etc., 2013).

Използват се следните логически съотношения:

$$A + B = (A \oplus B) + 2 (A \wedge B) \quad (1)$$

$$A + B = (A \vee B) + (A \wedge B) \quad (2)$$

КОНТРОЛ НА АРИТМЕТИЧЕСКИ ОПЕРАЦИИ

Основните аритметически операции в компютърните системи са:

- Събиране;
- Изваждане;
- Умножение;
- Деление.

За контрол на аритметични операции трябва да се използват кодове, пригодени за контрол и изпълнение на аритметически операции, за които:

$$[x](\text{операция})[y]=[x(\text{операция})y], \quad (3)$$

където x и y са операнди,

$[x]$ и $[y]$ – техните кодове,

$[x(\text{операция})y]$ – код на резултата

(операция) – аритметическа операция (Smrikarova, S. & etc., 2013).

Един от тези кодове е контролът по модул. Контролът на аритметични операции с използване на кода с контрол по модул се основава на следните свойства на остатъците (Smrikarova, S. & etc., 2013):

- ако $N1 \pm N2 = N3$, то $[C(N1) \pm C(N2)] \pmod{M} = C(N3)$;
- ако $N1 \cdot N2 = N3$, то $[C(N1) \cdot C(N2)] \pmod{M} = C(N3)$;
- ако $N1 = N2 \cdot N3 + N4$, то $C(N1) = \{ [C(N2) \cdot C(N3)] \pmod{M} + C(N4) \} \pmod{M}$.

Необходимо е също и да се знае, че две числа са сравними по $\text{mod } M$, ако са равни остатъците, получени при делението на числата на M . При извършване на дадена аритметична операция с информационните разряди на два операнда, същата операция, но по $\text{mod } M$ се извършва и с остатъците, записани в контролните разряди на операндите. При отсъствие на грешка двата резултата се сравняват по $\text{mod } M$ (Smrikarova, S. & etc., 2013).

При числовия контрол по модул се въвежда излишък в информацията, чрез добавяне на контролни разряди k на брой към информационните m . Съдържанието на контролните разряди е равно на остатъка R от делението на даденото число N (съдържанието на информационните разряди) на определено положително число M , наречено модул. Качеството на контрола по модул зависи от величината на модула M . Един от най-използваните контролни по модул е с величина $M=3$ – т.н. „Контрол по модул 3“ (Smrikarova, S. & etc., 2013).

ИЗИСКВАНИЯ КЪМ ИНТЕРАКТИВНИЯ ОБУЧИТЕЛЕН СОФТУЕР

Към проектирания обучителен софтуер са формулирани следните изисквания:

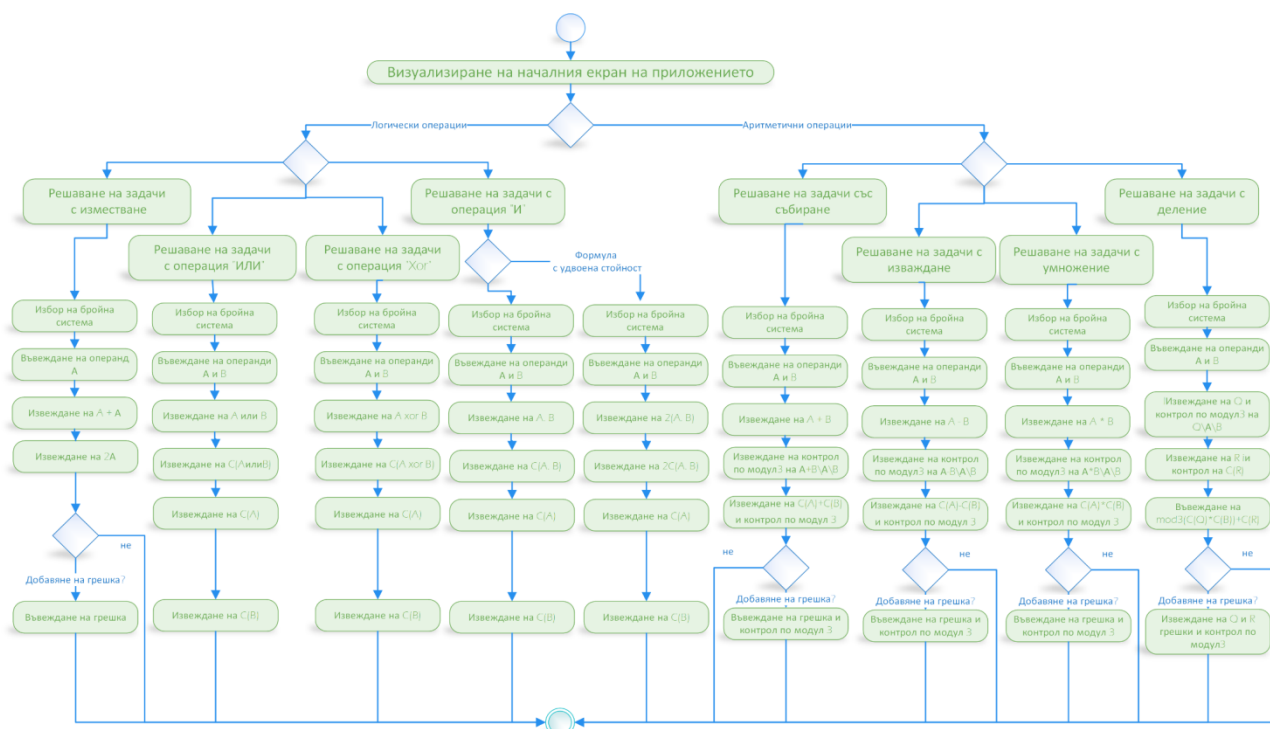
- Софтуерът да бъде с интуитивен дизайн и максимално опростен дизайн, който да насочва неопитният потребител бързо и лесно да усвоява функциите му, без допълнително обучение.
- Дизайнът на системата да бъде стилизиран, професионален и лесен за конфигуриране.
- Да се обособят два модула за представяне на теоретичния материали и за решаване на задачи;
- Да се разработят две версии на приложението: за преподаватели и за студенти;
- За решаване на задачите да се използва шестнадесетична и двоична бройна система;
- Всеки бит да се въвежда в отделен текстбокс;
- Да се извършват манипулации върху въведената входна последователност (изтриване и редактиране);
- Итеративното представяне да бъде визуализирано чрез панели, включващи текстови полета и надписи (labels);
- Да се извършват валидации за позволени бутони и входната информация;
- Да се извършва валидация за големината на входните операнди;
- Да се отчита броя на допуснатите грешки;
- При модул „Преподавател“ да се визуализира решението след въвеждането на входната последователност;
- Да се извеждат съобщения към потребителя при некоректна операция.

ПРОГРАМНИ ЕЗИЦИ И ИНСТРУМЕНТАЛНИ СРЕДСТВА

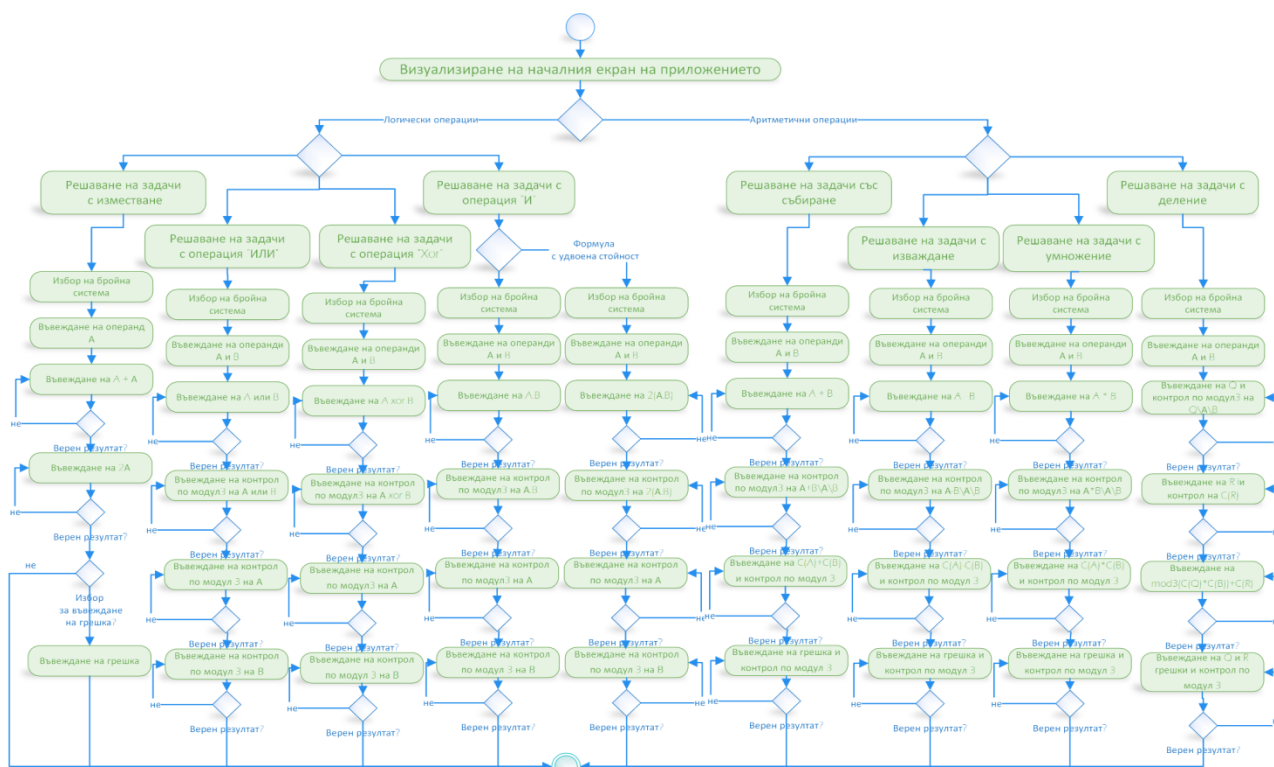
Описаният интерактивен обучителен софтуер за решаване на задачи от контрол на аритметически и логически операции представлява Microsoft .Net Application. За реализацията му е избран обектно ориентиран език C# и средата за разработка Microsoft Visual Studio 2013. Visual Studio 2013 е мощна интегрирана среда за разработка на софтуерни приложения за Windows. DOT NET платформа поддържаща различни езици за програмиране – C#, C++, Visual Basic и различни техники за разработка на софтуер Win32, Com, ASP.NET, ADO.NET, Entity Framework, Windows Form, WPF, Silverlight и други.

ПРОЕКТИРАНЕ НА ОБУЧИТЕЛНИЯ СОФТУЕР

На Фиг. 1 и Фиг. 2 са представени диаграмите на дейностите на двата типа потребители на интерактивния софтуер: преподавател (Фиг. 1) и студент (Фиг. 2). Функционалността на модула за преподаватели е аналогична с тази на модул „Студент“ с тази разлика, че задачата се решава автоматично след въвеждането на входните операнди, докато в модула за студентите се спира на всяка стъпка и се изчаква потребителят да въведе вярната стойност.



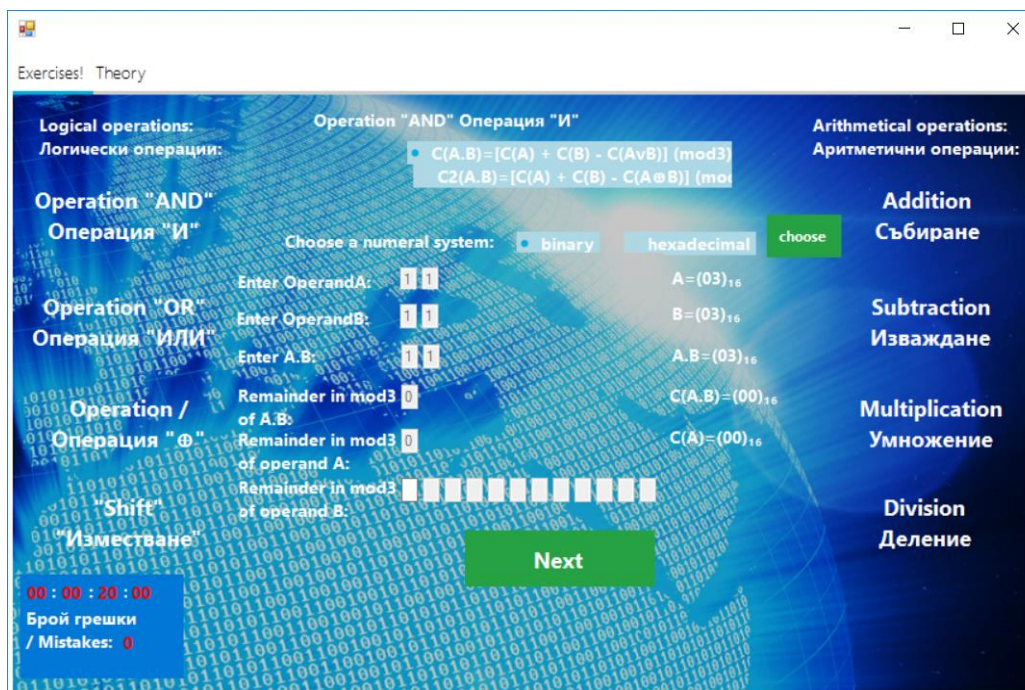
Фиг. 1. Диаграма на дейности на преподавател



Фиг. 2. Диаграма на дейности на студент

Приложението е проектирано и създадено на два езика – български и английски език. На всеки от етапите студентите трябва да изчислят и въведат верния отговор, за да продължат. Предвидена е възможност и за отчитане на допуснатите грешки и засичане на времето за решаване на конкретния тип задачи.

На Фиг. 3 е представен пример за решаване на задача за контрол на логическата операция „И“.



Фиг. 3. Задача с логическа операция „И“

На Фиг. 4. е представена задача за контрол на аритметическата операция „Деление“.



Фиг. 4. Задача с аритметическа операция „Деление“

ИЗВОДИ

През последните години сме свидетели на много изследвания свързани с методите за тестване на функционалността на аритметико-логическото устройство. За правилното функциониране на аритметико-логическото устройство трябва да се използват методи и техники, които да осигуряват надеждна и устойчива работа без грешки. Тези методи е важно да бъдат изучавани от компютърните специалисти, тъй като контролът на аритметическите и логически операции е съществен за осигуряване на надеждността в компютърната система.

Представеният интерактивен образователен софтуер за решаване на задачи от контрол на аритметически и логически операции се използва в практическите упражнения по дисциплината „Надеждност и диагностика на компютърни системи“ със студентите от специалност „Компютърни системи и технологии“. Обучителният софтуер е с интуитивен и удобен за работа интерфейс, който предоставя на студентите възможност постъпково да проследяват и изучават методите за контрол на аритметически и логически операции. Двуетичният интерфейс на приложението осигурява възможност за обучение на чуждестранни студенти, които ще изберат дисциплината по време на обучението си по програма Еразъм.

ACKNOWLEDGMENT

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- Mano, M. M., & Kime, C. R. (2008). *Logic and computer design fundamentals* (Vol. 3). Prentice Hall.
- Medwed, M., & Mangard, S. (2011). *Arithmetic logic units with high error detection rates to counteract fault attacks*. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011 (pp. 1-6). IEEE.
- Patel, J. H., & Fung, L. Y. (1982). *Concurrent error detection in ALU's by recomputing with shifted operands*. *IEEE Trans. Computers*, 31(7), 589-595.
- Rao T. (1970). *Biresidue Error-Correcting Codes for Computer Arithmetic*. *Computers, IEEE Transactions on*, vol. C-19, pp. 398–402.
- Smrikarova, S., & G. Ivanova (2013). *Lecture notes for the course „Reliability and Diagnostics of Computer Systems“*, URL: <https://e-learning.uni-ruse.bg> (Accessed on 05.04.2018)
- Siewiorek, D., & Swarz, R. (2017). *Reliable Computer Systems: Design and Evaluatuion*. Digital Press.

AN INNOVATIVE METHOD FOR STUDYING THE COMPONENTS IN COMPUTER MOTHERBOARD USING AUGMENTED REALITY²¹

Lenimir Zdravkov – Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 82 888 855
E-mail: vwlz1995@gmail.com

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 855
E-mail: givanova@ecs.uni-ruse.bg

Abstract: *An innovative approach for learning the elements in computer motherboard using augmented reality has been presented. The paper discusses an innovative and interactive way of presenting computer components using augmented reality. An interactive mobile application for visualizing objects and information about them is described. The software is designed for the needs of education using augmented reality to present various ports, slots and components of the computer system like real objects. Two modules are designed for beginners and advanced students.*

Keywords: *Computer, Components, Motherboard, Augmented Reality, Unity, 3D model, Vuforia*

ВЪВЕДЕНИЕ

Компютрите са неизменна част от ежедневието ни. Това налага да се повиши информираността и грамотността на подрастващите относно изграждащите компоненти на компютърната система и тяхното действие. Необходимо е още в началния етап на образование учениците да се запознаят с основните елементи на компютъра. Наличните на пазара учебници за изучаване на устройството на компютъра е необходимо да бъдат модернизирани в контекста на дигиталната ера, за да представят изучаваните компютърни компоненти по-ефективно и така да се повиши мотивацията на учащите.

В последните години стават все по-популярни нови учебните пособия с иновативни решения за привличане на вниманието и интереса на съвременните дигитални обучаеми. Един от тези подходи е например използването на добавена реалност (Ivanova, G. & etc., 2014). Тя позволява използването на игрови елемент в образованието, който е все по-популярен похват в последните години, за да се мотивира и ангажира интереса на съвременния обучаем (Ariso, J. M. & etc., 2017).

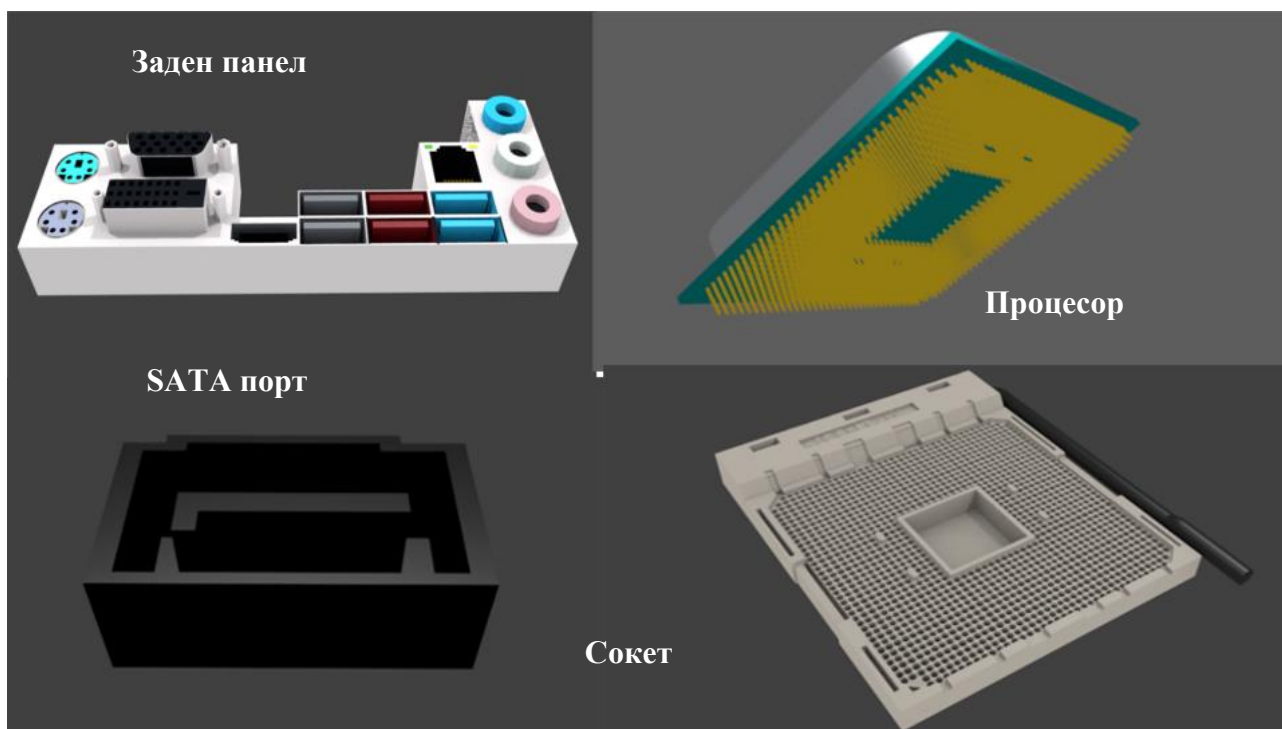
Добавената реалност (augmented reality) е сравнително нова технологията, която се различава от виртуалната реалност (virtual reality) по това, че първата използва за „основа“ съществуваща реалност и наслоява върху нея визуализирани 3D обекти с различни цели на приложение (Chen, P., & etc., 2017). В резултат се появява смесена среда, в която реалността и нереалното съществуват заедно с цел създаване на определено преживяване (Rezende, W. J & etc., 2017).

Най-разпространените начини за визуализация на моделите, използвани в добавената реалност, е чрез маркери или GPS или Wi-Fi – базирана локация (Ozdemir, M. & etc., 2018). Настоящия доклад разглежда система, която използва специално създадени маркери за визуализиране на компютърните компоненти.

²¹ Докладът е представен на студентската научна сесия на 27.IV.2018 в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ИНОВАТИВЕН МЕТОД ЗА ИЗУЧАВАНЕ НА ЕЛЕМЕНТИТЕ В КОМПЮТЪРНАТА ДЪННА ПЛАТКА ЧРЕЗ ДОБАВЕНА РЕАЛНОСТ

ЕТАПИ НА ПРОЕКТИРАНЕ И РЕАЛИЗИРАНЕ

В етапа на подготовката се създават 3D моделите (Фиг. 1), които ще се използват в приложението. Информацията за компонентите се събира, анализира и подготвя като се взема предвид сложността на компонента, планираните модули на приложението и търсеното въздействие на данните. Основният софтуер, който се използва на първия етап е Blender – софтуер за моделиране, анимация, текстуриране, рендиране и др. Чрез тази програма се създават първоначално „мрежите“ (meshes) на модела. След това чрез прилагане на различни трансформации върху обекта, като транслация, ротация и скалиране се създава първоначалният модел, върху който на по-късен етап се прилагат различни методи като екструдирание, съединяване с други обекти и изрязване на определена област от него. Така се създава един основен вариант на модела. Върху него следва да се приложат различни техники за заглаждане, скосяване на ръбове, обработка на върховете (vertices), ъглите (edges) и „лицата“ (faces) на модела, така че той максимално да наподобява на реалния обект. Крайната фаза на проектирането на моделите е да се извърши текстуриране, т.е. на моделите да се присвои материал – в случая главно метал или пластмаса и след това, ако е нужно, допълнителни текстури за засилване на реализма. Завършвайки този етап, моделите се експортират заедно със всички ресурси, заделени към тях, за да може да се използват във втория етап. Към първия етап още може да отбележим и създаването на маркерите, нужни за работата на приложението. Те се създават посредством програмата Inkscape, която представлява векторен графичен редактор. Избран е подходът с векторен редактор поради лесната скалируемост при принтиране.



Фиг. 1. 3D модели на компютърни компоненти

Вторият етап включва разработването на програмната логика на приложението, използвайки софтуерните решения Unity и Vuforia. Vuforia е „надстройката“ за добавена реалност върху Unity Game Engine. Във Vuforia API се намират всички методи, чрез които се осъществява добавената реалност: ARCamera модульт, който служи за откриване на целите, различни други методи и вградени компоненти като виртуални бутони, GPS локализация и др. Именно чрез използване на Vuforia се указва на кои цели - кои модели ще се визуализират, тяхната големина, начална позиция при визуализацията и др., (Фиг. 2).



Фиг. 2. Визуализиране на модел на процесор върху маркер

Основната работа на приложението се изгражда от Unity. Чрез Unity се създава началният екран, менюто, чрез което ще се избират двата модула на приложението – за начинаещи и напреднали, и бутонът за затваряне на апликацията. Модулът за напреднали ще бъде заключен, докато не се „изиграе“ целият модул за начинаещи. В модулът за начинаещи всеки модел ще бъде представен с различна цел, т.е. различните компоненти ще бъдат разгледани един по един, като по този начин ще се наблегне на всеки един от компонентите с нужното внимание. При модулът за напреднали компонентите ще са сглобени в дънната платка. Чрез методите на Unity ще се създаде checkpoint система в модулът за начинаещи, която да отчита кой модел е бил разгледан и кои остават. По този начин ще трябва всички цели да бъдат сканирани, моделите - разгледани и информацията прочетена, за да се премине към следващия модул. Сложността на информацията при двата модула ще е различна. В модула за начинаещи ще бъде дадена обща информацията за компонентите и тяхното действие, като няма да се навлиза в дълбочина за структурата им. В модула за напреднали, чрез виртуалните бутони, ще бъде визуализирана по-сложна информация за всеки обект, както и техническа информация. Данните за всеки обект/модел ще се показват на екрана след натискане съответния модел. В полето с информацията се прикрепя линк към cloud server, в който за съответния компонент има записани файлове с допълнително описание за разглежданите обекти. Така замислено, разделянето на приложението на два модула с нарастваща сложност, логически и програмно можем да определим като две нива на компютърна игра (Linowes, M. & etc, 2017). Unity позволява експортирането на приложението както за desktop, така и за Android и iOS, с минимални усилия от страна на разработчика. За мобилните операционни системи е нужно SDK-то, което се използва за разработка на приложения за дадената мобилна операционна система, да бъде инсталирано на компютъра.

Важно е да споменем и софтуерната добавка LeanTouch, което е част от Unity Assets, която позволява да се осъществят визуалните манипулации ротация и скалиране върху моделите (Фиг. 3).

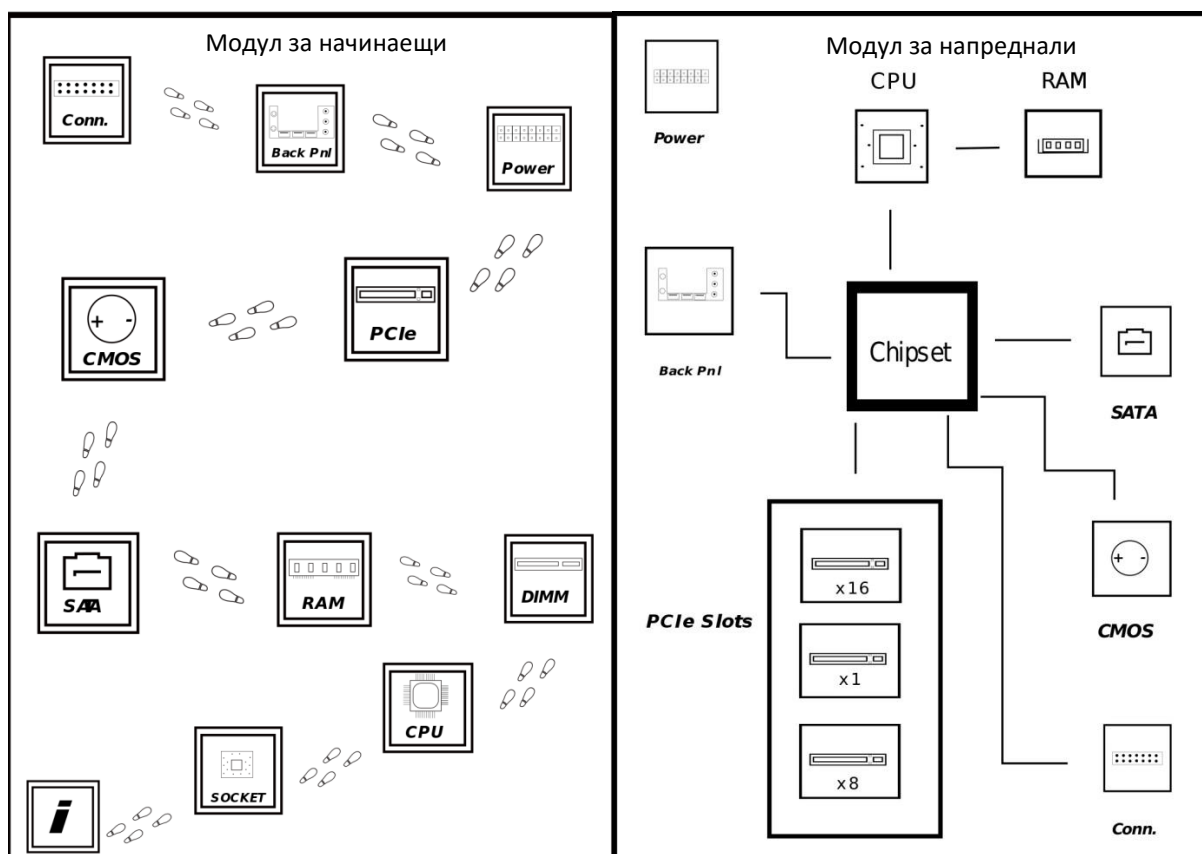
Устройството, което ще се използва, за да визуализира обектите, се насочва към маркера и когато софтуерът, чрез камерата, открие целта, визуализира съответния указан модел за тази цел (Linowes, M. & etc, 2017).



Фиг. 3. Прилагане на скалиране върху модела

На Фиг.4 е представена системата от маркери за модулите за начинаещи и напреднали.

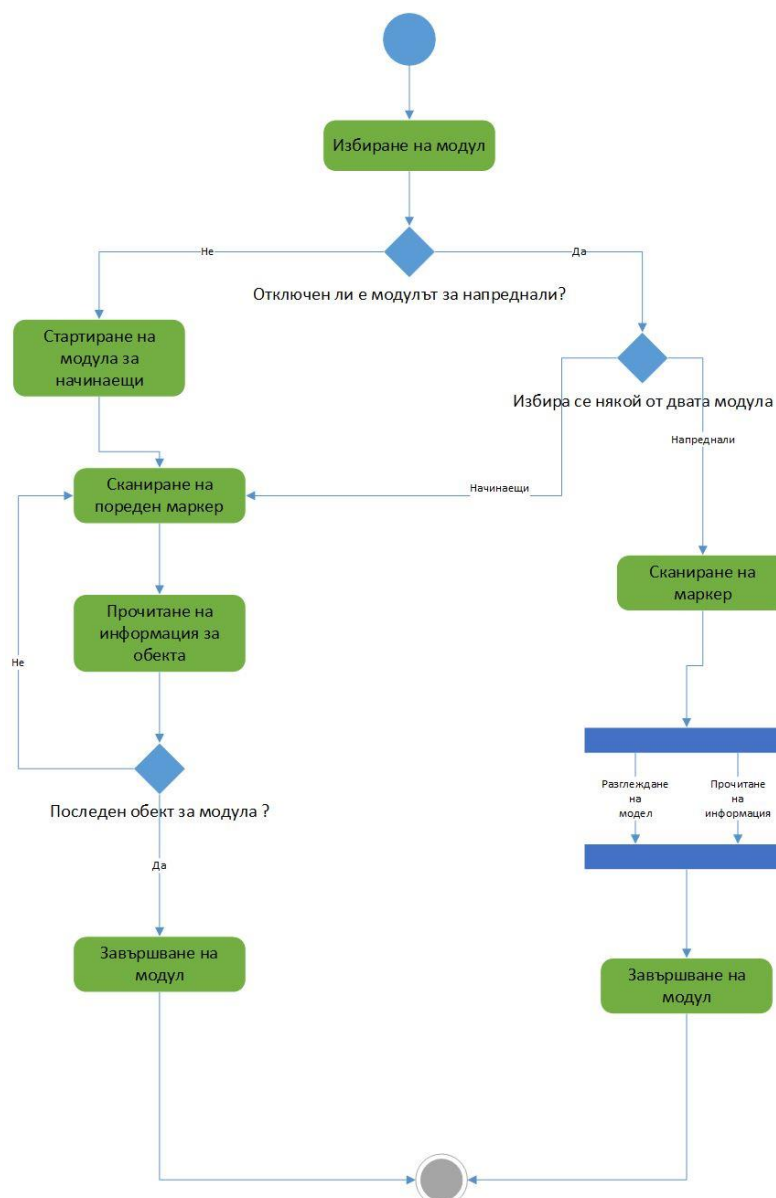
Последователността от маркери и модели в модула за напреднали е според важността на елементите в системата, (Schmalstieg, D., & etc 2016). Нужно е да бъдат сканирани всички маркери и да бъдат разучени всички компоненти, за да се премине към модул напреднал.



Фиг. 4. Система от маркери за модулите начинаещ и напреднал

За модула „напреднал“ системата от маркери е представена вдясно на Фиг.4. Повечето от моделите, които са използвани са същите като от модула за начинаещи, но са представени в друг контекст – процесорът и оперативната памет са монтирани в прилежащите им слотове, добавени са още слотове за PCIe шината и информацията е с различно ниво на сложност, (Schmalstieg, D., & etc 2016).

На фиг. 5 е представено схематично как работи приложението. За да се отключи модула за напреднали, където информацията е по-сложна и моделите са представени в контекста на истинското дъно, обучаемият трябва да завърши успешно модула за начинаещи.



Фиг. 5. Логика на действие на приложението

ИЗВОДИ

Изучаването на хардуерните компоненти на компютърната система е част от компютърната грамотност на съвременния обучаем. Описаното приложение предоставя един иновативен и интуитивен образователен подход за изучаване на компютърните компоненти, като се предоставя възможност за представяне на по-скъпи компютърни компоненти чрез техни визуално подобни 3D модели. Закупуването на най-актуалните на пазара скъпо струващи реални компютърни компоненти е непосилно за образователните институции и те трудно успяват да включват най-новите и модерни технологии в образователния процес. Представянето на модели, които наподобяват максимално точно реалните компютърни части, в среда, близка до реалната, е начин учебният процес да бъде направен по-забавен и същевременно да се задълбочи нивото на познание.

В последните години образователните институции се стремят да подобряват учебния процес, чрез въвеждане на иновативни технологии в изучаваните учебни предмети. Използването на нова технология, като добавената реалност, създава усещане в учащите, че

институцията им е в крак с новостите и заедно с това се увеличава и тяхната мотивираност за учене.

Визуализираните 3D модели в приложението представят точно реалните компютърни компоненти, а информацията за тях е достоверна, добре формулирана и лесна за възприемане. Компонентите в модула за начинаещи са подредени по важността им като приложението интуитивно води учащия да следва и изучава 3D моделите по начертания път. Възможността за прилагане на визуална манипулация на обектите, като ротация и скалиране, предоставя възможност моделите да се разглеждат от всички страни и изучаването им да става по-лесно.

Чрез разделянето на приложението на два модула (две нива) се цели да се използва игрови подход в обучението и така да се предизвика любопитството на обучаемия и същият да проследи докрай начертания път в първо ниво, за да отключи второ ниво - модула за напреднали и така да задълбочи знанията си в новото ниво.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Ariso, J. M. (2017). *Augmented Reality: Reflections on Its Contribution to Knowledge Formation*. De Gruyter.

Chen, P., Liu, X., Cheng, W., & Huang, R. (2017). A review of using Augmented Reality in Education from 2011 to 2016. In *Innovations in Smart Learning* (pp. 13-18). Springer, Singapore.

Ivanova, G., Aliev, Y., & Ivanov, A. (2014). Augmented reality textbook for future blended education. In *Proceedings of the International Conference on e-Learning* (Vol. 14, pp. 130-136).

Linowes, M., Babilinski, K., (2017). *Augmented reality for developers. Build practical augmented reality applications with Unity, ARCore, ARKit and Vuforia*.

Ozdemir, M., Sahin, C., Arcagok, S., & Demir, M. K. (2018). The Effect of Augmented Reality Applications in the Learning Process: A Meta-Analysis Study. *Eurasian Journal of Educational Research (EJER)*, (74).

Rezende, W. J., Albuquerque, E. S., & Ambrosio, A. P. (2017). *Use of Augmented Reality to Support Education*.

Schmalstieg, D., Hollerer, T. (2016). *Augmented Reality: Principles and Practice (Usability)*, First Edition.

A SURVEY OF WEB BASED EDUCATIONAL PLATFORMS²²

Plamen Dimitrov – Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 897 254 117
E-mail: plamen.dimitrov06@abv.bg

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 855
E-mail: givanova@ecs.uni-ruse.bg

Abstract: The following paper presents a survey of web based educational platforms. The advantages and disadvantages of some of the most popular e-learning platforms are discussed. The survey investigates the main characteristics of the examined platforms through the use of a detailed comparison. Conclusions about future developments and improvements are made.

Keywords: e-learning, distance learning, online education, e-learning platform

ВЪВЕДЕНИЕ

Технологиите в наши дни се развиват изключително бързо, което води и до нарастване на необходимостта от получаване на нови знания през целия живот. Уеб базираните среди за електронно обучение са неразделна част от учебния процес на днешното поколение обучаеми. Интересът към тези среди се дължи най-вече на лесния достъп до компютърна техника и интернет, и нуждата от достъпно образование по всяко време, от всяко място и с всяко крайно устройство (Sargsyan R. G. & etc., 2016). Уеб базираните обучителни системи предоставят на потребителите интерактивен и ефикасен начин за усвояване на нови знания под формата на електронни учебни материали, електронни тестове, видео клипове, задачи за самоподготовка и други. Съществуват над 300 типа среди за електронно обучение, като повече от 70 са безплатни системи за електронно обучение (Al-Ajlan, A. , 2017).

Много изследвания се занимават с различни анализи на ефективността от обучението в средите за електронно обучение. Според интервю направено с 15 онлайн инструктори, всеки от които с доказан опит в онлайн обучението, се отбелязва, че създаването на добри взаимоотношения и комуникация между обучаващи се и лекторите в средите за електронно обучение е изключително важно. Това може да се постигне само от инструктори със съпричастие към студентите, страст към процеса на обучение и желание да помогнат на учащите да успеят (Sun, A., & Chen, X., 2016).

В доклада са разгледани някои от най-популярните в последните години среди за електронно обучение (Udemy, Pluralsight, Coursera), като е направен и сравнителен анализ на някои от основните им характеристики с една от популярните среди за обучение в България – платформата за електронно обучение на Софтуерния университет.

ПРЕДИМСТВА НА СРЕДИТЕ ЗА ЕЛЕКТРОННО ОБУЧЕНИЕ

Уеб-базираните обучителни системи се характеризират с изобилие от функционалности с цел удовлетворяване на нуждите на всички потребители. Възможността за получаване на

²² Докладът е представен на студентската научна сесия на 27.04.2018 в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ПРОУЧВАНЕ НА УЕБ БАЗИРАНИ ОБУЧИТЕЛНИ СИСТЕМИ

дигитален или физически сертификат след завършването на даден курс е едно от предимствата от обучението в тези среди. На фиг. 1 е представен сертификат, получен при завършването на курс по програмиране. Възможността за получаване на сертификат за повишаване на квалификацията мотивира не само начинаещи, а и курсисти с утвърдени знания да запишат даден курс в среда за електронно обучение. Такъв документ има положително влияние при интервюта за работа и увеличава шансовете на кандидата.



Фиг. 4. Сертификат получен при завършване на курса "Programming Basics with C# - August 2017" от Software University

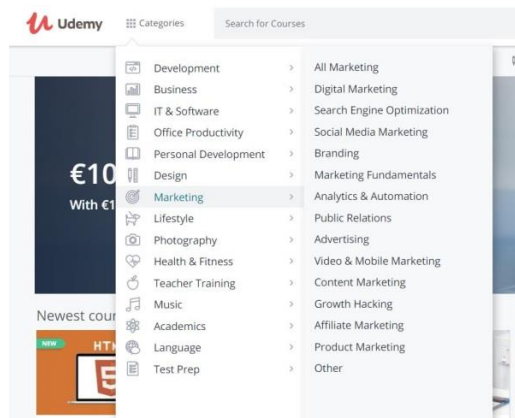
Друго предимство, което получават обучаващите се онлайн, е възможността за използването на различни мобилни устройства. Голяма част от обучителните системи предоставят на потребителите възможността да използват Android или iOS приложения, позволявайки лесен достъп до учебните ресурси по всяко време. Pluralsight, например, разполага и със софтуерен продукт за Windows и Mac устройства, който позволява гледане на видео уроци без нуждата от интернет достъп. Ресурсите на избрания курс могат да бъдат изтеглени локално на устройството и достъпвани без нуждата от интернет връзка.

Могат да бъдат обобщени и други предимства на уеб базираните обучителни системи:

- Възможност за общуване с други курсисти чрез синхронни и асинхронни инструменти за онлайн комуникация;
- Възможност за индивидуално обучение с помощта на ментор;
- Тестове за периодична проверка с резултати след решаването на задачите;
- Задачи за самоподготовка и курсови проекти.

UDEMY

Udemy е уеб базирана обучителна платформа основана през 2010 г. от Ерън Бали, Октай Каглър и Гаган Бияни (Shivraj, K. S. , 2014). В момента системата разполага с библиотека от 65 000 курса, както платени така и безплатни, в категории като разработка на софтуер, маркетинг и продажби, фотография, дизайн, езици и други (Фиг. 2).

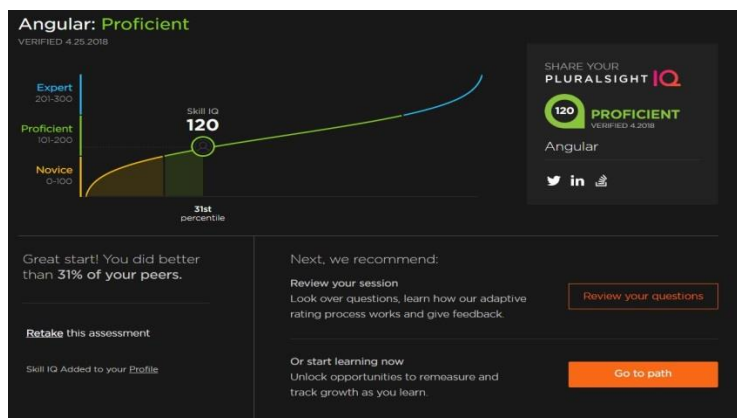


Фиг. 5. Категориите в Udemy

Голямото разнообразие от онлайн курсове позволяват на всеки курсист лесно и бързо да намери нужните материали (Jeff, J., 2014)). В условия на силно конкурентна среда, Udey се стреми да предоставя и някои функционалности, които да ѝ дадат желаното предимство. В системата е предоставена например възможност за комуникация между обучаващите се и лектора посредством система за въпроси и отговори. Това елиминира нуждата от многократно отговаряне на един и същ въпрос от страна на преподавателя, както и позволява на по-напредналите студенти да помагат на останалите. Друга полезна функционалност на платформата е модула за следене на прогреса. Той помага на курсистите да следят до коя част на курса са стигнали и ги мотивира да го завършат.

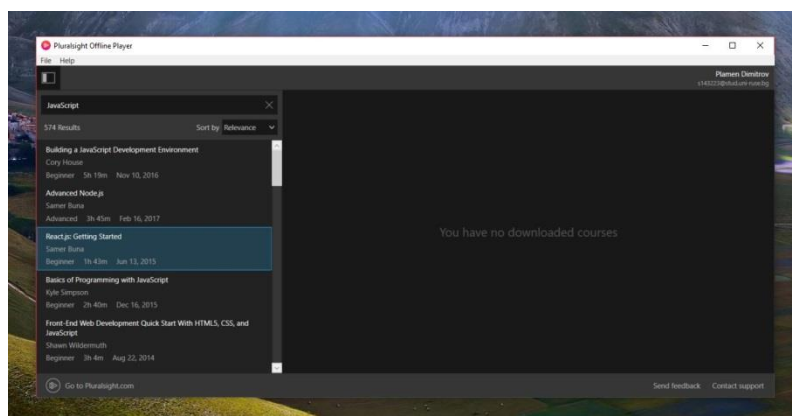
PLURALSIGHT

Сред най-популярните уеб базирани обучителни платформи се нарежда и Pluralsight (Berg, K., & etc., 2017). За разлика от Udey, тази платформа има приблизително 6000 курса на разположение на курсистите - значително по-малко курсове. Важен факт е, че тези курсове са изготвени от 1300 експерти в различни области, което прави платформата изключително привлекателна за професионалисти, нуждаещи се от актуализация на знанията си. Курсовете са организирани под формата на направления (наречени „paths”), като преди стартирането им потребители могат да решат тест от 20 въпроса, показващ знанията им в момента. На Фиг. 3 е показан резултатът от един такъв тест.



Фиг. 6. Резултати от теста за Angular направлението

За разлика от Udey, където курсистите заплащат за всеки отделен курс, Pluralsight предоставя на потребителите си достъп до всички учебни материали, след закупуване на месечен или годишен абонамент.



Фиг. 4. Offline плейърът на Pluralsight

Разбира се има и пробен период от 10 дни, както и преференциални цени за екипи и цели корпорации. Платформата разполага с изобилие от функционални удобства. Менторската

система представлява лесен начин за справяне с проблеми при изучаване на нови технологии. Срещу определена сума курсистите могат да си уговорят виртуална среща с ментор и да споделят затрудненията, които са срещнали в процеса на обучение. Друг плюс, който отличава Pluralsight от останалите си конкуренти, е възможността на курсистите да достъпват материалите без нуждата от Интернет връзка. След изтеглянето на софтуерен продукт (представен на Фиг. 4) от сайта на Pluralsight, и видео материалите на желания курс, потребителите мога да ги гледат по всяко време без нуждата от връзка към мрежата.

СОФТУЕРЕН УНИВЕРСИТЕТ

СофтУни (Nakov, S., 2014) е български представител на уеб базираните обучителни платформи с изключителен успех на територията на страната. Основана е в края на 2013 г. от Светлин Наков и Христо Тенчев в град София, като в момента се водят периодични, безплатни курсове в 38 града присъствено или онлайн. Мисията на СофтУни е предоставянето на висококачествено и достъпно образование в сферата на ИТ и дигиталния свят, създавайки истински професионалисти и бъдещи лидери, допринасяйки за една мечта, България - Силициевата долина на Европа. Освен качествено образование на ИТ кадри, потребителите имат на разположение и курсове по дигитален маркетинг (SoftUni Digital) и работа с графика и дизайн (SoftUni Creative).

Курсовете се водят от професионалисти, с опит в дадената сфера, и са практически ориентирани. Обучаващите се онлайн получават достъп до видео материалите на курса, възможността да гледат в реално време лекциите и да задават въпроси посредством виртуалната класна стая (Фиг. 5), както и достъп до Judge системата, позволяваща решаване на задачи с резултати в реално време. Тези, които изберат присъствената форма получават всичко споменато до момента, както и помощ от асистенти и ментори. При успешното преминаване на изпит в края на курс с оценка 5 или по-висока се получава сертификат, който показва изучавания по време на обучението материал, оценката и имената на притежателя. Курсистите показали изключителни резултати получават помощ с намирането на работа посредством кариерния център на Софтуерния университет.



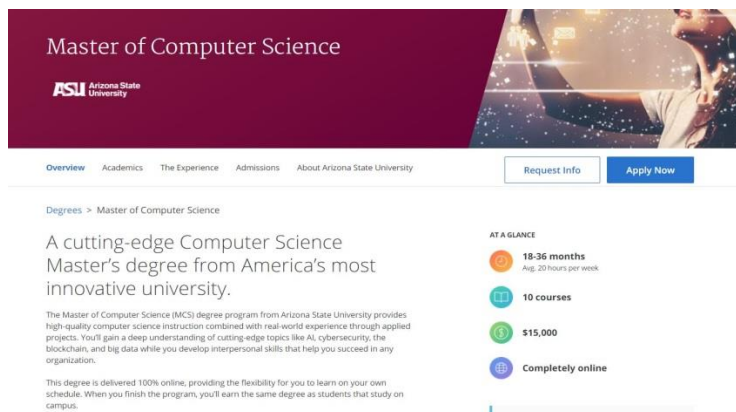
Фиг. 5. Виртуална класна стая – СофтУни

COURSERA

Coursera (Young, J. R., 2012) е основана през 2012 г. от Андрю Нг и Дафни Колър - професори по компютърни науки в университета Станфорд. В момента в системата има 2000 курса в 180 различни специализации. Регистрирани са 25 милиона потребители, които се обучават изцяло онлайн чрез платформата. Това, което отличава Coursera от разгледаните до момента системи е големия брой партньорства с компании и акредитирани университети от цял свят. Някои от компаниите са Google, IBM, MongoDB INC, Yandex, Goldman Sachs и други. Списъкът на университетите партньори включва: University of Arizona; Stanford University; University of Washington; Universitat de Barcelona; École des Ponts ParisTech и други.

Платформата позволява на потребителите си да се сдобият с диплома за бакалавърска или дори магистърска степен от някои от най-добрите университети в цял свят изцяло онлайн.

Цената на такъв курс може да стигне десетки хиляди долара, а продължителността му няколко години. На Фиг. 6 е показан пример за курсът по „Компютърни Науки“ на Arizona State University.



Фиг. 6. Курс по „Компютърни Науки“ на Arizona State University

В Таблица 1 е представен обобщен сравнителен анализ на някои от основните характеристики на разгледаните платформи в доклада.

Таблица 1. Сравнителен анализ на разгледаните платформи за електронно обучение

Характеристики	Udemy	Pluralsight	SoftUni	Coursera
Курсове	65,000+	6,000+	150+	2000+
Категории	15	11	3	11
Цена	10,99\$+/курс	29\$/месец	330лв/модул	39\$ - 20,000\$+
Субтитри	Английски	Английски	Български	Многоезични
Многоплатформеност	Android/iOS/TV	Windows/Android/iOS/TV	Responsive	Android/iOS
Модули за комуникация	Q&A модул	Менторска система	Форум	Менторска система
Тестова система	-	PluralsightIQ	SoftUni Judge	+
Offline система	-	+	-	-
Система за следене на прогреса	+	+	-	+
Брой студенти	10 милиона	5 милиона	50,000+	30 милиона
Партньори университети	-	-	4	149

За да бъдат конкурентно способни платформите за електронно обучение трябва да притежават: голямо разнообразие от курсове; приветлив, ясен и лесен за употреба потребителски интерфейс; възможност за комуникация между обучаващите, и между обучаващите и лекторите; многоплатформеност; възможност за самоподготовка под формата на задачи, курсови проекти и тестове; системи за следене на прогреса; да включват като партньори известни университети с възможност за издаване на сертификати и дипломи.

ИЗВОДИ

Уеб базираните обучителни системи са предпочитано средство за получаване на нови умения и актуализиране на знанията. През последните години броят на тези системи значително нараства. Доказателство за това са, не само разгледаните примери, но и системи на големи корпорации като Cisco – Cisco Networking Academy (Liu, M., 2014) и Lynda на LinkedIn (LinkedIn, (2015)). Все по-популярни са и уеб системите за масови онлайн отворени курсове MOOCS (Emanuel, E. J., 2013) въз основата на отворени образователни ресурси, които предлагат възможност за достъп до качествено свободно образование на хиляди обучаеми.

Уеб базираните платформи за електронно обучение са тенденция в съвременното образование. Техните функционални възможности, характеристики и структура непрекъснато ще се променят и адаптират, за да отговарят на съвременните нужди на обучаемите. Затова е важно да се правят проучвания и сравнителни анализи на най-популярните среди за електронно обучение, за да могат периодично да се оценяват техните предимства и недостатъци. Данните от тези проучвания и анализи могат да се използват за подобряване на: дизайна на системите; качеството на предлаганите курсове; методите и инструментите за доставка на образователните материали; подкрепата на обучаемите и намаляване на процента на отпадналите онлайн курсисти.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- Al-Ajlan, A. (2017). Measuring the Evolution of Open Source Software in E-Learning Systems. *Journal of Engineering Technology* (ISSN: 0747-9964), 6(1), 124-145.
- Berg, K., & Bujak, P. (2017). As the World Turns: A Glimpse into Distance Education. *Issues in Information Systems*, 18(4).
- Emanuel, E. J. (2013). Online education: MOOCs taken by educated few. *Nature*, 503(7476), 342.
- Jeff, J. (2014). Course of Udemy: The course structure of Udemy.
- LinkedIn, (2015) Lynda. <https://www.lynda.com> (Accessed on 10.04.2018).
- Liu, M. (2014). Cisco Networking Academy: Next-Generation Assessments and Their Implications for K-12 Education. Clayton Christensen Institute for Disruptive Innovation.
- Nakov, S. (2014). SoftUni. URL: <https://softuni.org> (Accessed on 10.04.2018)
- Sargsyan, R. G., & Torrijos, M. B. (2017). Massive Online Courses (Moocs) And Their Role In The Digitalized Era. *EPH-International Journal of Science and Engineering* (ISSN: 2454-2016), 3(5), 44-53.
- Shivraj, K. S. (2014). Massive Open Online Courses (MOOC) Offered by NPTEL and IIT Kanpur: An Overview. *Indian Journal of Information Sources and Services*, 4(2), 7-12.
- Sun, A., & Chen, X. (2016). Online Education and Its Effective Practice: A Research Review. *Journal of Information Technology Education*, 15.
- Young, J. R. (2012). Inside the Coursera contract: How an upstart company might profit from free courses. *The Chronicle of Higher Education*, 19(07).

MODIFIED RING-BASED MUTUAL EXCLUSION ALGORITHM WITH FAILOVER RECOVERY²³

Eng. Ivaylo Rashkov, BSc – MSc Student

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Tel.: +358 84 661663

E-mail: ivorashkov@abv.bg

Assoc. Prof. Milen Lukanchevski, PhD

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Phone: 0877 303 850

E-mail: mil@ieee.org

Abstract: The ring-based aka token-ring mutual exclusion algorithm is the simplest decentralized mutual exclusion algorithm. The only preliminary information any constituent system processes should know are its own identifier and the identifier of its immediate neighbor. The main drawback of this attractive algorithm is strong presumption of full system reliability which is impractical.

The modified version of algorithm proposed in this paper eliminates presumption of full system reliability and guarantees failover recovery from any kind of multiple faults. Two strategies exists – without or with communicational ring reconfiguration.

Main part of the modification is new algorithm for mutual exclusion token control/recovery. It introduces one new initialization message <mrk_me_clr> and extension of mutual exclusion token <mrk_me> with time marker Ti. The time marker is used to control validity of mutual exclusion token and initialization message serves as separator between mutual exclusion token before and after failover recovery.

Keywords: Distributed Systems, Failover Recovery, Distributed Mutual Exclusion, Token-Ring.

ВЪВЕДЕНИЕ

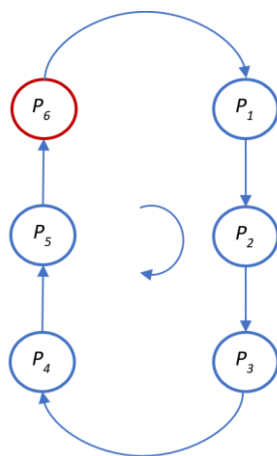
Под „взаимно изключване“ се разбира изключване на възможността повече от един активен обект (*процес, нишка*) да се обръщат в даден момент към поделен ресурс (Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G., 2011, Kshemkalyani, A. & M. Singhal. 2008, Lukanchevski, M., 2014). Докато при многозадачните операционни системи тази задача се решава чрез глобални променливи под контрола на ядрото, при разпределените системи решението се основава на идеята за критичната секция. Под критична секция се разбира участък от кода на активния обект, в който се достъпва поделения ресурс.

По отношение на алгоритмите за разпределено взаимно изключване се поставят три условия: безопасност (*safety, ME1*), осъществимост (*liveness, ME2*) и справедливост (*fairness, ME3*). Абсолютно задължителни от тях са първите две.

Алгоритмите за разпределено взаимно изключване се делят на две големи фамилии: разрешителни (*permission-based*) и маркерни (*token-based*) (Raynal, M. 1991). Един от популярните маркерни алгоритми е кръговият. При него системата е хомогенна и се състои от n идентични процеса $P_1, P_2, \dots, P_n$, свързани в логическа кръгова топология. На фиг. 1 е представена подобна система при $n = 6$. Съществува един единствен маркер *ME* (*token, служебно съобщение*), който се разпространява в кръга (*комуникационния пръстен*) по посока на часовниковата стрелка. Тъй като само процеса, притежаващ маркера *ME*, може да се намира в критичната секция, се удовлетворява условието *ME1*. Условието *ME2* се удовлетворява от

²³ Докладът е представен на студентската научна сесия на 27.04.2018г. в секция “ Комуникационна и компютърна техника“ с оригинално заглавие на български език: МОДИФИЦИРАН КРЪГОВ АЛГОРИТЪМ ЗА ВЗАИМНО ИЗКЛЮЧВАНЕ С ВЪЗСТАНОВЯВАНЕ ПРИ ОТКАЗ.

кръговата топология. Маркерът *ME* се генерира еднократно от координатора *P6* – процеса с най-голям идентификатор. Координаторът се определя при инициализацията на системата чрез алгоритъм за разпределен избор, като самата подредба на процесите в пръстена е без значение.

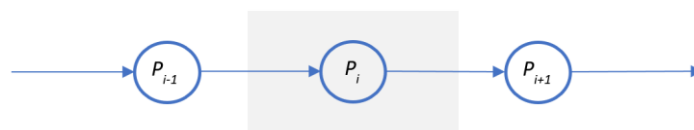


Фиг. 1. Канонично представяне

Основен недостатък на този базов вариант на кръговия алгоритъм за разпределено взаимно изключване е предположението за недопустимостта на откази от никакъв вид – нито на процеси, нито на канали.

Поставена е задачата да се намери подходяща модификация на базовия вариант на кръговия алгоритъм, която да допуска физическо отпадане/възстановяване, на който и да е от процесите по всяко време. След възстановяването на отказалия процес, всички останали процеси в кръга продължават нормалното си изпълнение от работното състояние, в което са били в момента на отказа, а маркерът *ME* се възстановява. При това трябва да се гарантира строгото съблюдаване на условията *ME1* и *ME2*. Комуникационният пръстен не се реконфигурира.

В представения доклад се разглеждат някои възлови моменти от модифицирания кръгов алгоритъм *Mx1ME* за разпределено взаимно изключване с възстановяване от отказ във варианта без реконфигуриране на комуникационния пръстен. Алгоритъмът е оригинален и позволява възстановяването от множествени откази.



Фиг. 2. Графична дефиниция на понятието „отказ на процес“

Понятието „отказ на процес“ се дефинира като отказ на самия процес *P_i*, отказ на участъка от входния канал на *P_i* или отказ на участъка от изходния канал, за който *P_i* отговаря (фиг. 2).

ИЗЛОЖЕНИЕ

Алгоритъм за управление/възстановяване на маркера

Маркерът *ME* е критичен за алгоритъма ресурс. За да се гарантира строгото съблюдаване на условията *ME1* и *ME2* този маркер не трябва да се губи или дублира. Това предполага разширяването на базовия алгоритъм за разпределено взаимно изключване с алгоритъма *MrkME* за управление/възстановяване на маркер *ME*.

Алгоритъмът *MrkME* използва две съобщения: инициализиращ маркер *<mrk_me_clr>* и модифициран маркер *ME <mrk_me, Ti>*. Модификацията на маркер *ME*, спрямо базовия

алгоритъм, се заключава в добавянето на характеристичното време T_i , което отговаря на момента на създаването на маркер ME от координатора.

Използват се следните собствени вътрешни променливи:

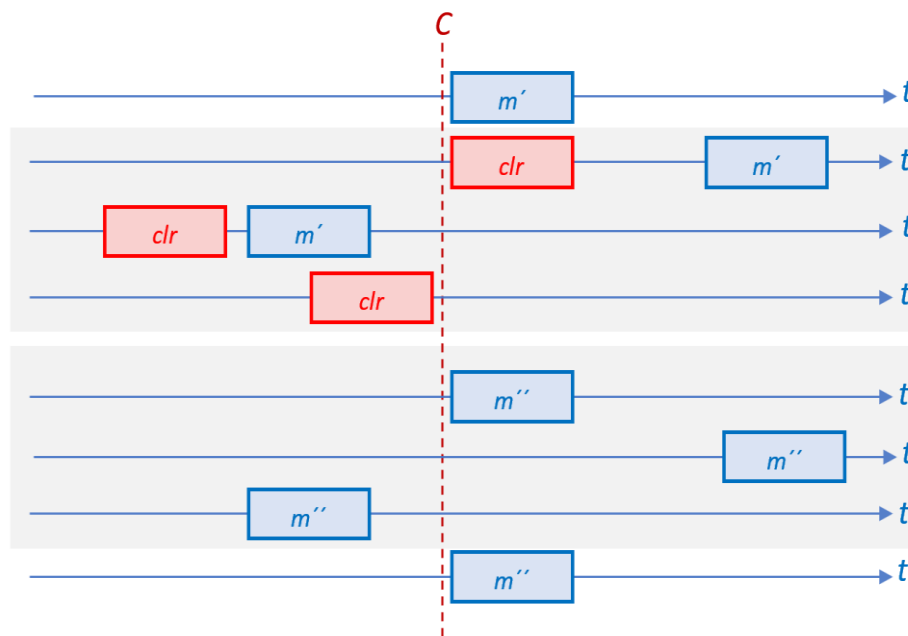
- променливата на състоянието $boolMrkMe$;
- променливата $timeMrkME$, съдържаща характеристичното време T_i на текущия за процеса маркер;

- признака $strClrPending$ за задържан инициализиращ маркер.

Освен това се използва идентификатора на координатора $idCoordinator$, дефиниран в алгоритъма за разпределен избор и променливата на състоянието $meState$ на алгоритъма $Mx1ME$.

Алгоритъмът $MrkME$ се стартира от координатора след приключване на избора и протича в два паса. По време на първия пас координаторът формира инициализиращото съобщение $\langle mrk_me_clr \rangle$. Вторият пас се стартира също от координатора като се формира новия маркер $ME \langle mrk_me, T_i \rangle$.

Принципът на работа на алгоритъма $MrkME$ във времето е представен на фиг. 3. Преди отказа в пръстена се разпространява маркера m' . След отказа и възстановяването на отказалия процес, координаторът C стартира първия пас като формира инициализиращото съобщение clr . Това съобщение се разпространява след маркера m' и анулира неговото по-нататъшно действие. Щом маркер m' постъпи отново в координатора бива погълнат от него. В някакъв следващ момент в координатора постъпва и инициализиращото съобщение clr , което е признак за анулиране действието на маркера m' в целия кръг и първият пас приключва. Координаторът C стартира втория пас като формира новия маркер $ME m''$. Премавайки последователно по кръга, този маркер се фиксира от всеки от процесите като нов маркер ME . При първото последващо постъпване на маркер $ME m''$ в координатора завършва и вторият пас. Така завършва възстановяването на маркер ME в цялата система и се възстановява работоспособността на разпределения кръгов алгоритъм за взаимно изключване.



Фиг. 3. Условно представяне на принципа на работа

Формалната спецификация на алгоритъма $MrkME$ съдържа и псевдокода на множеството събития: $OnInit$, $OnAfterElection$, $OnReceiptOf \langle mrk_me_clr \rangle$, $OnClear$, $OnReceiptOf \langle mrk_me, T_i \rangle$, например

OnAfterElection: If $i = idCoordinator$ $boolMrkME := true$ $timeMrkME := \perp$ Send $\langle mrk_me_clr \rangle$ {Начало на Пас 1} EndIf	OnReceiptOf $\langle mrk_me_clr \rangle$: If $meStatus \neq ME_HELD$ OnClear() Else $strClrPending := \langle mrk_me_clr \rangle$ {Фиксиране в протокола} EndIf
---	--

Събитието *OnInit* сработва при първоначалната инициализация на процеса, докато събитието *OnAfterElection* – при всяко приключване на разпределения избор. Разпределеният избор от своя страна се стартира не само след първоначалната инициализация на процеса, но и след възстановяването на отказалия процес, а това води до последователно включване на всички системни процеси в избора. Това гарантира, че независимо дали отказалият процес е бил координатора или не, алгоритъмът *MrkME* ще се стартира.

При получаването на инициализиращото съобщение сработва събитието *OnReceiptOf $\langle mrk_me_clr \rangle$* . Интересна негова особеност е задържането на инициализиращото съобщение $\langle mrk_me_clr \rangle$ в процеса, намиращ се в критичната секция ($meStatus = ME_HELD$). Едва след като процесът излезе от секцията ($meStatus = ME_RELEASED$) и освободи текущия маркер (m' от фиг. 3), инициализиращото съобщение се препредава по кръга. Така, това съобщение формира граница между стария и новия маркер *ME* и предотвратява дублирането му.

Модифициран кръгов алгоритъм за разпределено взаимно изключване

Базовият вариант на кръговия алгоритъм за разпределено взаимно изключване, описан в литературата (Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G., 2011, Kshemkalyani, A. & M. Singhal. 2008, Lukanchevski, M., 2014) е модифициран с оглед интеграцията му с алгоритъма *MrkME*. С това се преодолява основният недостатък на базовия вариант - предположението за недопустимостта на отказите.

Интеграцията с алгоритъма *MrkME* предполага използването на двете съобщения: инициализиращ маркер $\langle mrk_me_clr \rangle$ и модифициран маркер *ME* $\langle mrk_me, Ti \rangle$. Освен собствената вътрешна променлива на състоянието *meState* се използва идентификатора на координатора *idCoordinator*, дефиниран в алгоритъма за разпределен избор, както и вътрешните променливи на алгоритъма *MrkME* - *boolMrkME*, *timeMrkME* и *strClrPending*.

Запазено е множеството събития на базовия кръгов алгоритъм: *OnInit*, *OnAfterElection*, *OnEnter*, *OnReceiptOf $\langle mrk_me \rangle$* , *OnRelease*. При *OnEnter* и *OnReceiptOf $\langle mrk_me \rangle$* обаче се забранява смяната на работното състояние при невалиден маркер *ME*.

Добавено е събитието *OnRingUp*, което отговаря на момента на откриване изграждането на комуникационния пръстен от процеса *Pi*. Това е възлов момент, тъй като от него се стартира алгоритъма за разпределен избор.

OnRingUp:
 {Стартиране на алгоритъма за разпределен избор}
 $idCoordinator := Election()$

Преработката на събитието *OnRelease* е по-различна, заради необходимостта от задържане на инициализиращото съобщение $\langle mrk_me_clr \rangle$ в процеса, намиращ се в критичната секция ($meStatus = ME_HELD$). В случая на процеса винаги се разрешава да излезе от секцията като освободи текущия маркер (m' от фиг. 3). След това се проверява за наличието на задържано инициализиращо съобщение и ако има такова, то се препредава по кръга.

OnRelease:

```

If meStatus = ME_HELD
    meStatus := ME_RELEASED
    Send <mrk_me>
    {Занес в протокола}
    If strClrPending ≠ ⊥
        MrkME::OnClear()
        strClrPending := ⊥
    EndIf
EndIf

```

За да се обхванат възможните сценарии на отказите се извършва развиване на кръговата топология в линейна, като стремежът е координаторът да е в средата на последователността от процеси (фиг. 4).



Фиг. 4. Разгънато представяне

Това дава възможност за компактно кодиране на обобщеното състояние на всички процеси в кръга в момента на възникване на отказа, например: *RRRRRR* (всички процеси са в *ME_RELEASED*), *RRRRHR* (*P2* е в *ME_HELD*), *WRRRHR* (*P4* е в *ME_WANTED*, *P2* е в *ME_HELD*), *WRRRHW* (*P4* и *P3* са в *ME_WANTED*, *P2* е в *ME_HELD*), *WHRRRR* (*P4* е в *ME_WANTED*, *P5* е в *ME_HELD*).

	4	5	6	1	2	3
	R	R	R	R	H	R
1						
2						
3						
4						
5						
6						

	4	5	6	1	2	3
	W	R	R	R	H	R
1						
2						
3						
4						
5						
6						

	4	5	6	1	2	3
	W	H	R	R	R	R
1						
2						
3						
4						
5						
6						

Фиг. 5. Таблично представяне на сценариите *RRRRHR*, *WRRRHR* и *WHRRRR*

При предположението за единични откази се получават табличните представяния на сценариите от фиг. 5, където по колони са номерата на процесите, а по редове е маркиран отказалия процес. Предложената таблична форма може да се разшири и за случаите на множествени откази.

ИЗВОДИ

Представеният модифициран кръгов алгоритъм за разпределено взаимно изключване с възстановяване от отказ е реализиран със средствата на средата Embarcadero C++ Builder® 10.2. Реализацията е ориентирана, както към демонстрация на работата на алгоритъма, така и към евентуалното му практическо използване. Извършени са тестове на всички възможни сценарии на единични и двойни откази като са фиксирани диагностическите протоколи на отделните процеси.

REFERENCES

- Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed Systems: Concepts and Design*. 5th Ed. – Boston: Addison-Wesley, p. 1008.
- Kshemkalyani, A. & M. Singhal. (2008) *Distributed Computing: Principles, Algorithms, and Systems*. – Cambridge: Cambridge University Press, p. 736.
- Lukanchevski, M. (2014). *Razpredeleni sistemi i algoritmi: teoria i praktika*. Ruse: Izdatelski tsentar na RU „A. Kanchev”, ISBN 978-619-7071-35-1, p. 213.
- Raynal, M. (1991) *A simple taxonomy for distributed mutual exclusion algorithms*. // ACM SIGOPS Operating Systems Review Volume 25 Issue 2, April, p. 47-50.

ONLINE SYSTEM FOR ERASMUS STUDENT APPLICATION AND MONITORING INCOMING MOBILITIES²⁴

Viktor Bratov, BSc – MSc Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 82 888 855
E-mail: viktorbratov@yahoo.com

Assoc. Prof. Galina Ivanova, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Phone: +359 82 888 855
E-mail: givanova@ecs.uni-ruse.bg

Abstract: The paper reviews existing online systems for application of Erasmus mobility in some European universities. The problem about the rising number of incoming foreign student applications in University of Ruse is discussed to improve and modernize the quality of the services in the admission process. The paper presents an approach for designing an online system, which will keep a database with current information for incoming student mobilities. The system will maintain effective student feedback and coordination between the Department for International Relations, coordinators and contact persons at faculties in University of Ruse.

Keywords: Erasmus mobility, foreign students, web-base system, student application

ВЪВЕДЕНИЕ

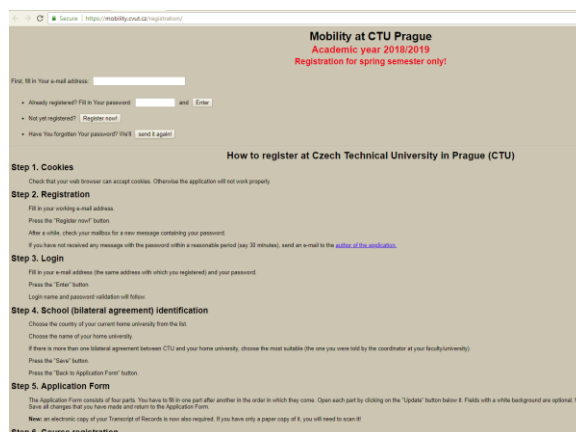
Програма Еразъм+ е програма на Европейската комисия за образование и обучение на младежи, която е продължение на програмите "Сократ" и "Учене през целия живот". В Програмата участват всички членки на ЕС, Лихтенщайн, Исландия, Норвегия, Турция и Македония (KA103 – мобилности с програмни държави), както и други държави-партньори (KA107 - мобилности с партниращи държави) (Cairns, D. & etc., 2018).

Студентските мобилности по програма Еразъм са два вида – с цел обучение и с цел практика. Мобилностите с цел обучение предоставят възможности на студентите да се обучават в чуждестранни образователни институции, партниращи на Русенски университет „Ангел Кънчев“, без да е необходимо да се заплащат академичните такси в приемащия университет. Мобилностите с цел практика предоставят възможности на студентите да проведат стаж или практика във фирма или друго работно място в чужбина като е необходимо практиката да отговаря на обучението на студентите и техните потребности за професионално развитие. Мобилностите могат да бъдат реализирани в трите степени на обучение (бакалавърска, магистърска и докторска степен). Продължителността е от 2 до 12 месеца в рамките на всяка от трите академични степени на висше образование. Студентите получават месечна финансова помощ в размер, специфичен за всяка държава (Diaz, N. S., & Crichton, H. 2017).

Процесът за кандидатстване на студентите за мобилност е свързан с попълване на редица документи и е необходимо да бъде модернизиран с използване на съвременните информационни и комуникационни технологии, които могат значително да подобрят качеството на целия процес (Walasek, T., & etc., 2007). В много университети (Cătălin, P., Marian, N., & Peana, V., 2017) се използват онлайн платформи за кандидатстване и обслужване на студентите участващи в програмата за Еразъм мобилности (Teichler, U. 2017). На Фиг. 1 е

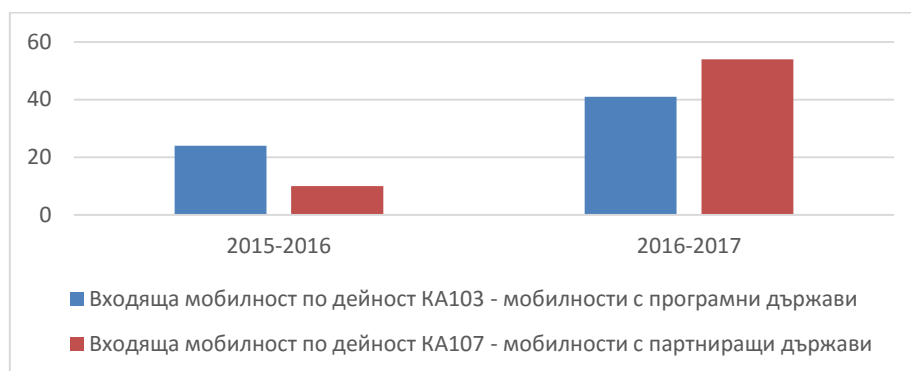
²⁴ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ОНЛАЙН СИСТЕМА ЗА КАНДИДАТСТВАНЕ И ПРОСЛЕДЯВАНЕ НА ЕРАЗЪМ ВХОДЯЩИ МОБИЛНОСТИ НА СТУДЕНТИ.

представена онлайн платформата за кандидатстване на Еразъм студенти на Техническия университет в Прага.



Фиг. 1. Онлайн платформа за кандидатстване на Еразъм студенти в Czech technical university в Прага

В последните години броят на пристигащите Еразъм чуждестранни студенти в Русенския университет нараства. На Фиг. 2 е представена статистика на входящите мобилности в Русенски университет през академичните години 2015-2016 и 2016-2017 за мобилности по дейности КА 103 и КА 107 (UoR, 2018).



Фиг. 2. Входяща студентски мобилности по програма Еразъм в Русенски университет

Стремежът към повишаване качеството на обслужване на студентите, които се включват в Еразъм мобилности, поддържането на достоверна информация за входящите мобилности, ефективната обратна връзка със студентите и координацията на взаимодействието между отдела за международна дейност, координаторите и контактните лица по факултети, наложила необходимостта от създаване на онлайн система за кандидатстване на Еразъм студенти и поддържане на база от данни с всички входящи мобилности на Русенски университет.

Целта на настоящия доклад е да се проектира и разработи онлайн система за кандидатстване и проследяване на пристигащите Еразъм студенти в Русенски университет, с която да се подпомогне процеса на кандидатстване на студентите и координирането на организационните дейности, свързани с приемането на тези студенти за обучение в съответните партниращи факултети.

НОРМАТИВНИ ДОКУМЕНТИ ЗА МОБИЛНОСТ ПО ПРОГРАМА ЕРАЗЪМ В РУСЕНСКИ УНИВЕРСИТЕТ

Всички мобилности в Русенски университет следват вътрешни нормативни правила, които описват въпросите на администрирането на Еразъм мобилността. Правилата се основават на документи на Европейската комисия (ЕК) като: Ръководство по програмата,

финансовите споразумения с Центъра за развитие на човешките ресурси и нормативните актове на Русенски университет „Ангел Кънчев”, (UoR, 2018).

За мобилностите по програма Еразъм се сключват двустранни споразумения между Русенския университет и другите висши училища от програмните държави подписали Еразъм Харта или с партниращи държави, които имат PIC номер от ЕК. Двустранните споразумения за обмен с цел практика се сключват между Русенския университет и организации от програмни държави, които предлагат работни места, (UoR, 2018).

За осъществяването на мобилностите отговарят следните лица: Университетски ECTS координатор, факултетни ECTS координатор и академични координатори (контактни лица). Университетският координатор следи мобилностите за целия Университет. Координаторите за факултетите координират мобилностите на факултетно ниво, а контактните лица участват в подготовката и провеждането на обучението или практиката за конкретните мобилности, за които имат сключено споразумение.

ЕТАПИ НА ПРОЕКТИРАНЕ НА ОНЛАЙН СИСТЕМА ЗА КАНДИДАТСТВАНЕ НА ЕРАЗЪМ СТУДЕНТИ В РУСЕНСКИ УНИВЕРСИТЕТ

На първи етап в процеса на проектиране на системата се направи изследване и анализ на онлайн платформи на други университети за кандидатстване на Еразъм студенти, за да се оценят предимствата и недостатъците на съществуващите решения и да се обобщат следните изисквания:

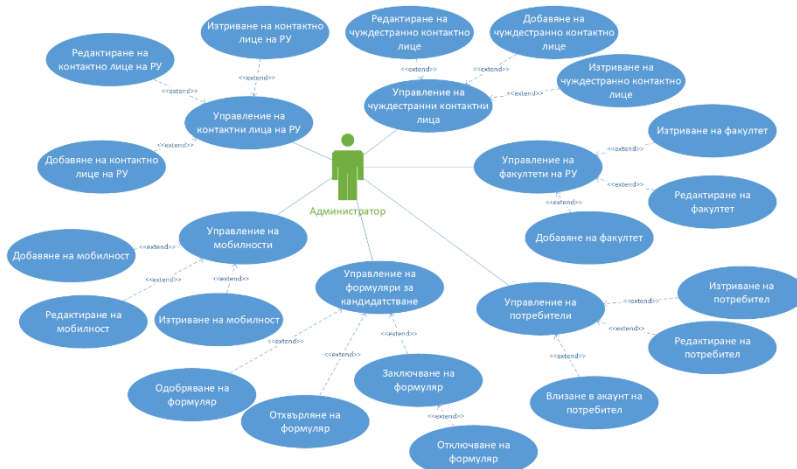
- Уеб-системата трябва да е с интуитивен интерфейс;
- При реализацията е необходимо да се използват съвременни технологии за дизайн, позволяващи поддържането и на мобилна версия на системата;
- Системата трябва да улеснява Еразъм чуждестранните студенти в процеса за кандидатстване и прием в Русенски университет;
- Системата да проследява всички входящи мобилности на студентите в обща база от данни;
- Да се поддържат актуални списъци на сключените споразумения за мобилности, със съответните кодове, факултети и контакти лица в обща база от данни;
- Системата трябва да осигурява възможност за генериране на pdf версия на формата за кандидатстване на студентите с възможност за принтиране и подписване;
- За по-лесно административно обслужване в системата е необходимо да се поддържа бърз и лесен начин за одобряване, отхвърляне и обслужване на формите на новите кандидати през административен модул.

На втори етап е направен обоснован избор на езици и среда за програмиране. Системата е написана на езика PHP с помощта на софтуерната рамка Laravel. Тя улеснява и ускорява процеса на разработка като опростява рутинни задачи като кеширане, сесии, автентикация, регистрация, рутиране и други. Приложенията направени с Laravel са качествени и сигурни и удобни за поддръжка от страна на разработчика.

Средата за разработка на системата е PHP Storm, избран заради отличното интегриране на рамката Laravel и поддръжката на разработка с нея. Програмата дава бърз и ясен достъп до всички файлове на проекта и динамично показва грешки в PHP кода и дори в потребителските изгледи написани на HTML.

Базата данни на системата е изградена с mysql, заради неговата сигурност и надеждност. MySQL е популярна система за управление на бази данни и се използва в много известни сайтове и е подходяща както за приложения с малък по размер данни, така и за системи съхраняващи огромно количество данни. Управлението на информацията става с удобен за ползване уеб интерфейс, достъпен през интернет и позволяващ на базата данни да се обновява по всяко време.

Процесът на проектиране на софтуерната система продължи със създаване на диаграми, описващи случаите на употреба на двата типа основни потребители на системата : студенти и администратори. На Фиг. 3 са представени дейностите на администраторите на системата.



Фиг. 3. Диаграма на случаи на употреба на потребител администратор

На Фиг. 4. е представена диаграмата на случаите на употреба на студент.



Фиг. 4. Диаграма на случаи на употреба на потребител студент

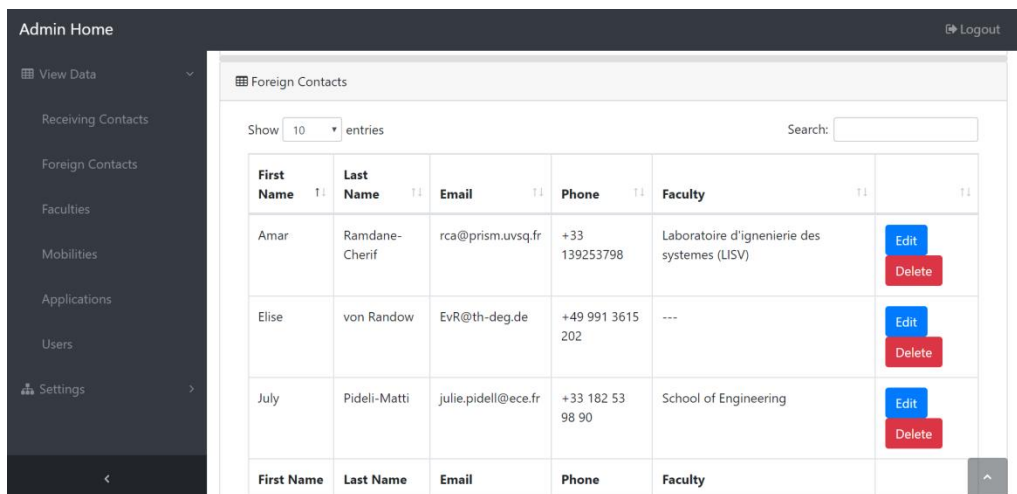
Релационният модел на базата от данни е представена на Фиг. 5.



Фиг. 5. Релационен модел на базата от данни

ФУНКЦИОНАЛНИ ВЪЗМОЖНОСТИ НА СИСТЕМАТА

Системата за кандидатстване на Еразъм студенти е изградена от два модула – административен и потребителски. Интерфейсът на двата модула е на английски език и поддържа мобилни версии за удобство на потребителите. На Фиг. 6 е представен административния модул на приложението за управление на информацията в базата от данни, потребителските акаунти и формулярите за кандидатстване.



Фиг. 6. Административен модул на приложението

Администраторът на системата поддържа и въвежда информацията в базата от данни за подписаните двустранни споразумения за мобилности между Русенски университет и партниращите университети. Етапите за въвеждане на нова мобилност в системата са представени на Фиг. 7.



Фиг. 7. Етапи за въвеждане на нова мобилност в онлайн системата

На Фиг. 8 е представена част от формата за кандидатстване на Еразъм студент.

Фиг. 8. Форма за кандидатстване на Еразъм студент

Описаната онлайн система за кандидатстване и проследяване на пристигащи Еразъм студенти ще бъде тествана от отдел Международна дейност, за да се внедри и използва в Русенски университет.

ИЗВОДИ

Създадената система ще улесни пристигащите Еразъм студенти в процеса на кандидатстване, като им се предоставя съвременен начин за регистрация и кандидатстване в Русенски университет.

Административният модул ще подпомогне работата на отдел Международна дейност в процеса на управление и проследяване на входящите мобилности на Еразъм студенти в Русенски университет, което ще допринесе и за оптимизиране на тази дейност и координация между международния отдел, координаторите и контактните лица по факултети.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Walasek, T., Piątkowski, J., & Morawska-Walasek, D. (2007). *Information Technologies supporting students' mobility*. Journal of Achievements in Materials and Manufacturing Engineering, 25(1).

Cairns, D., Krzaklewska, E., Cuzzocrea, V., & Allaste, A. A. (2018). *Mobility, Education and Employability in the European Union: Inside Erasmus*. Springer.

Diaz, N. S., & Crichton, H. (2017). *Mobility in Higher Education: New Needs to Improve it?*.

University of Ruse (UoR), Web site “University of Rousse - Erasmus Programme” (2017). URL: <http://erasmus.uni-ruse.bg/bg/?cmd=cmsPage&pid=17> (Accessed on 05.04.2018).

Teichler, U. (2017). *Internationalisation Trends in Higher Education and the Changing Role of International Student Mobility*. Journal of international Mobility, (1), 177-216.

Cătălin, P., Marian, N., & Ileana, V. (2017). *The Internationalisation of Higher Education, as Keystone for Development in Eastern Europe. A Case Study on Bucharest University of Economic Studies in Romania*. Ovidius University Annals, Series Economic Sciences, 17(1).

GOLANG – INTRODUCTION AND BASIC FEATURES²⁵

Teodora Yordanova – MSc Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 888 62 54 54
E-mail: tlyordanova@yahoo.com

George Stefanov – MSc Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 889 69 45 49
E-mail: gtstefanov@yahoo.com

Assist. Prof. Yordan Kalmukov, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 82 888827
E-mail: jkalmukov@ecs.uni-ruse.bg

Abstract: *This paper outlines the basic features of the newly released open-source general purpose programming language called Go (or Golang). It has been announced by Google in 2009 and publicly released in 2012. Its primary objective is to combine advantages of existing programming languages while avoiding their disadvantages. It is self-documenting, statically typed, scalable to large systems, productive and readable, not requiring integrated development environments, supporting networking and multiprocessing. The paper reviews its basic features, possible fields of application, syntax and compares it to popular existing procedural programming languages.*

Keywords: *Golang, Go programming language, open source, self-documenting, statically typed, scalable.*

ВЪВЕДЕНИЕ

Golang, наричан още Go, е нов програмен език. Създаден е от инженерите на Google и има за цел да обедини предимствата на съществуващите езици отстранявайки техните недостатъци. Той е език с отворен код и всичките му библиотеки са публично достъпни. Усъвършенстват се и се поддържат от Golang общността (Golang Community).

Кратка история

Езикът Go е анонсиран през ноември 2009 година, а версия 1.0 е представена през март 2012 година. Използван е в някои от продукционните системи на Google, както и от много други компании и проекти с отворен код.

Go започва като експеримент на инженерите от Google – Robert Griesemer, Rob Pike и Ken Thompson. Идеята им е била да се създаде език, който да отстрани най-често срещаните проблеми (зависимост от платформата, проблеми с управлението на паметта, генерирането на голям обем излишен код) при останалите езици като същевременно запази техните положителни характеристики. При създаването му основните цели са били:

- статично типизиран (static typing) и мащабируем до големи системи – като Java или C++;
- продуктивен и четим, без излишен код (boilerplate) – като Ruby или Python;
- да не изисква интегрирана среда за разработване, но все пак да ги поддържа;

²⁵ Докладът е представен на студентската научна сесия на 27 април 2018 в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: GOLANG – ВЪВЕДЕНИЕ И ОСНОВНИ ХАРАКТЕРИСТИКИ.

- възможност за мрежово и многопроцесорно програмиране.

В интервюта след излизането на езика, разработчиците споделят, че основната им мотивация за разработването на нов език е факта, че и тримата не харесвали C++.

Основни характеристики

Golang е самодокументиращ се език. Това означава, че всеки go файл, метод или структура трябва да съдържа коментар, описващ основната функционалност или употреба.

Този език е с отворен код. Това от своя страна означава, че общността на езика отговаря за неговата поддръжка (проверка и отстраняване на грешките в съществуващите библиотеки) и развитие (създаване на нови библиотеки и функционалности).

Инсталация

Инсталацията на Go се извършва чрез инсталатор предоставен от сайта на езика. При отварянето на сваления MSI файл се създава съответната папка на езика. В тази папка ще се съхраняват всички файлове (програми) написани на Golang, както и всички необходими библиотеки. За да може програмите, написани на Go да работят е необходимо да се добави bin папката в променливите на средата (environment variables) на операционната система. По конкретно в променливата PATH. След добавянето вече може да се пишат програми на Golang.

Езикът на Google не е зависим от средата за разработване и следователно не съдържа такава в инсталатора. Самият инсталатор предоставя възможност на операционната система да „разбира“ езика, но от своя страна не включва нищо друго – няма основни библиотеки или среда за разработване.

Go е език с отворен код (open source), което означава, че всичко свързано с езика е публично достъпно. Повечето библиотеки (включително и основните) са достъпни в GitHub. Свалянето се осъществява чрез команден ред, въвеждайки определен код за изтегляне на избраните библиотеки. Предварително трябва да се инсталира приложението Git, за да може да се осъществи достъп през Go.

За улеснено компилиране на програмите и четенето на кода може да се използва Visual Studio Code.

Посоченият начин за инсталация на Golang е конкретно за Windows операционна система и платформата GitHub. По подобен начин се осъществява и на останалите операционни системи. Повече информация има на сайта на езика (Golang Official Website, 2018).

Синтактични особености

Една програма на Go включва всички файлове, съдържащи се в нейната папка и се нарича, както се казва директорията ѝ. Програмата винаги се стартира от *main* метода на go файла, носещ същото име.

В езика няма наследяване, т.е. не съществуват директиви като *overload*, *override* или *overwrite*. Той разчита на директно напасване (*mapping*) за осъществяването на по-бърз достъп до методите.

Един go файл задължително съдържа пакет, към който принадлежи файла, и използваните библиотеки. Може да съдържа методи и/или структури, като винаги декларацията на структурите предхожда описанието на методите (фиг. 7).

```
// main.go

package main

import "fmt"

// Rectangle includes length and width of rectangle
type Rectangle struct {
    Length int
    Width  int
}

// Area of rectangle
func (r *Rectangle) Area() {
    return r.Length * r.Width
}

func main() {
    r := Rectangle{}
    fmt.Println("Default rectangle is: ", r)
    fmt.Println("Default rectangle area: ", r)
}
```

Фиг. 7. Примерна програма на Golang

Всяка нова команда започва на нов ред, т.е. не могат да се запишат две команди на един ред.

Класовете тук се наричат структури. Декларацията на структурата съдържа само описанието на основните полета. При дефинирането на даден метод се посочва дали той принадлежи на някой клас или не. Не е задължително клас методите да се намират само във файла, в който се намира структурата.

Декларирането на типа на дадена променлива става чрез ключовата дума *var* името на променливата и типа ѝ. За деклариране на тип на променлива и присвояване на стойност се използва операторът „:=“. За присвояване стойност от друг тип се използва същият оператор. Ако типът на променливата вече е деклариран, присвояването на стойност става с „=“. Ако се използва „:=“ за присвояване на стойност от същия тип, то няма да възникне грешка, но стойността на променливата няма да се промени.

В езика няма *do ... while* цикъл, а *for* цикъла се осъществява с помощта на *while* цикъл, който се бележи с ключова дума *for* (фиг. 8).

```
i := 1
for i <= 3 {
    fmt.Println(i)
    i = i + 1
}
```

Фиг. 8. Примерен *for* цикъл

Съществува *for each* цикъл, който се описва с помощта на ключовата дума *range* () (фиг. 9). При този начин на обхождане се инициализират две променливи – текущи индекс и стойност. Специален символ „_“ означава, че за програмната логика тази стойност не е от значение и не се използва, т.е. тя не се присвоява на променлива и не заема памет.

```

nums := []int{2, 3, 4}
sum := 0
for _, num := range nums {
    sum += num
}
fmt.Println("sum:", sum)

for i, num := range nums {
    if num == 3 {
        fmt.Println("index:", i)
    }
}

```

Фиг. 9. Пример за foreach цикъл

Езикът поддържа и условния оператор if ... else, който се използва по аналогичен начин на останалите езици (фиг. 10).

```

if 8%4 == 0 {
    fmt.Println("8 is divisible by 4")
}

if 7%2 == 0 {
    fmt.Println("7 is even")
} else {
    fmt.Println("7 is odd")
}

if num := 9; num < 0 {
    fmt.Println(num, "is negative")
} else if num < 10 {
    fmt.Println(num, "has 1 digit")
} else {
    fmt.Println(num, "has multiple digits")
}

```

Фиг. 10. If ... else структура

Особеност при този език е че в if условието може да съдържа освен проверка и присвояване на стойността на променливата (фиг. 11). Използва се за по кратко записване.

```

if c := rand.Intn(100); c > 50 {
    fmt.Println("Bigger than 50")
}

```

Фиг. 11. Пример за if условие с присвояване

Операторите за сравнение са: „==“ (равно), „>“, „<“, „>=“, „<=“ и „!=“ (различно). Логическите оператори са: „&&“ (и), „||“ (или) и „!“ (не). Операторът за инкрементиране с единица е „++“, а за декрементиране е „--“.

Не съществува утвърдена конвенция за именуването на променливите или форматирането на кода. Повечето програмни среди препоръчват свои собствени правила.

Основни източници на информация

Все още не съществуват учебници за този език, защото той е постоянно развиващ се и само документиращ се.

Основни източници на информация са: сайтът на езика golang.org, платформата github.com, в която се качват повечето библиотеки, и форумът stackoverflow.com.

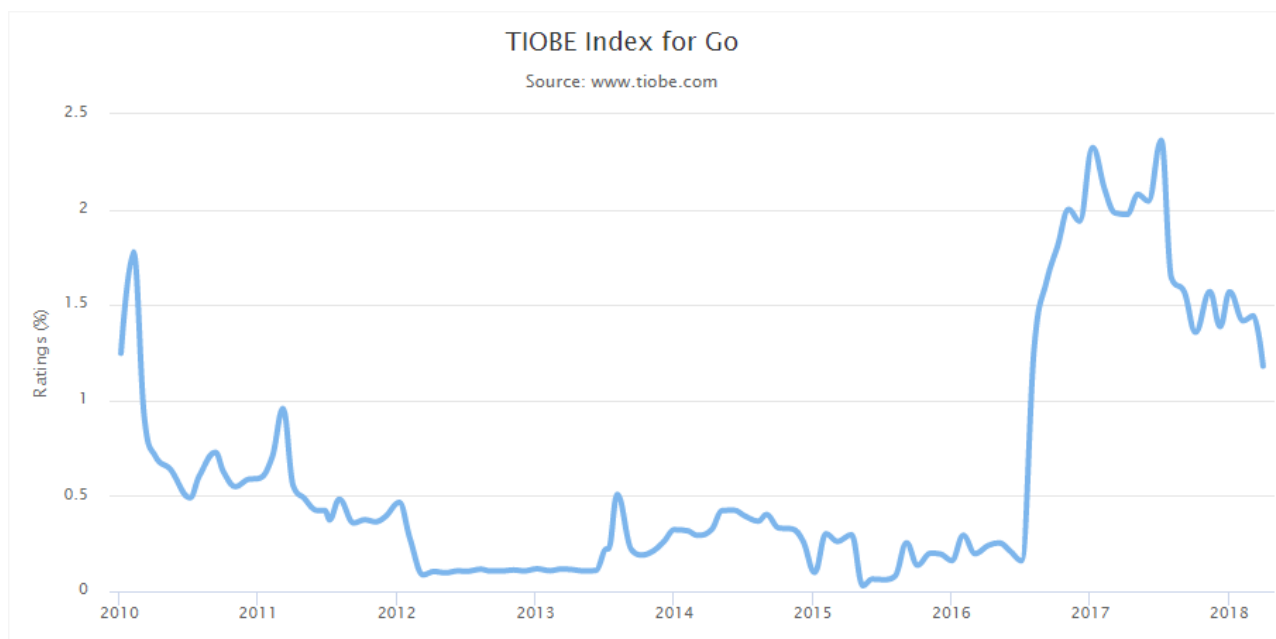
Съществуват и много други платформи, форуми и сайтове създадени специално за Golang.

Статистически данни

Според TIOBE за този месец (април 2018 год.) Go е едва на 19 място по популярност в света с рейтинг от 1.180 % (фиг. 12). Най-висока популярност езикът е имал в средата на 2017 год. (фиг. 13).

Apr 2018	Apr 2017	Change	Programming Language	Ratings	Change
1	1		Java	15.777%	+0.21%
2	2		C	13.589%	+6.62%
3	3		C++	7.218%	+2.66%
4	5	▲	Python	5.803%	+2.35%
5	4	▼	C#	5.265%	+1.69%
6	7	▲	Visual Basic .NET	4.947%	+1.70%
7	6	▼	PHP	4.218%	+0.84%
8	8		JavaScript	3.492%	+0.64%
9	-	▲▲	SQL	2.650%	+2.65%
10	11	▲	Ruby	2.018%	-0.29%
11	9	▼	Delphi/Object Pascal	1.961%	-0.86%
12	15	▲	R	1.806%	-0.33%
13	16	▲	Visual Basic	1.798%	-0.26%
14	13	▼	Assembly language	1.655%	-0.51%
15	12	▼	Swift	1.534%	-0.75%
16	10	▼▼	Perl	1.527%	-0.89%
17	17		MATLAB	1.457%	-0.59%
18	14	▼▼	Objective-C	1.250%	-0.91%
19	18	▼	Go	1.180%	-0.79%
20	20		PL/SQL	1.173%	-0.45%

Фиг. 12. Най-популярните езици в света според TIOBE



Фиг. 13. Промяната на популярността на Golang според TIOBE

ЗАКЛЮЧЕНИЕ

Go е лесен за разбиране и публично достъпен програмен език. Напълно независим от платформата. Въпреки това той не е много популярен. Поради тази причина основни източници на информация остават сайта на езика, самите библиотеки и техните описания, и форумите.

Езикът разполага с различни методи и библиотеки за динамична валидация на входни данни. Това заедно с мащабируемостта, независимостта от платформата и намаления обем заемана памет (по време на работа и като сорс файлове) прави Golang подходящ за изграждането на различни уеб услуги. От друга страна, липсата на каквото и да било наследяване не позволява Go да се използва за приложения разчитащи на фабрики, медиатори или други подобни шаблони за писане на код.

БЛАГОДАРНОСТИ

Този доклад се публикува със съдействието на проект 2018-ЕЕА-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- Git Official Website (2018), <https://git-scm.com/download>
- GitHub Official Website (2018), <https://github.com/>
- Golang Official Website (2018), <https://golang.org>
- Stack Over Flow Official Website (2018), <https://stackoverflow.com/>
- TIOBE Official Website (2018), <https://www.tiobe.com/tiobe-index/>
- Tutorials Point: Go operators (2018), https://www.tutorialspoint.com/go/go_operators.htm
- Visual Studio Code Official Website (2018), <https://code.visualstudio.com>
- Wikipedia: Go (2018), [https://en.wikipedia.org/wiki/Go_\(programming_language\)](https://en.wikipedia.org/wiki/Go_(programming_language))

REST SERVICES WRITEN IN GOLANG²⁶

Teodora Yordanova – MSc Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 888 62 54 54
E-mail: tlyordanova@yahoo.com

George Stefanov – MSc Student

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 889 69 45 49
E-mail: gtstefanov@yahoo.com

Assist. Prof. Yordan Kalmukov, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: +359 82 888827
E-mail: jkalmukov@ecs.uni-ruse.bg

Abstract: *This paper explains how to develop RESTful web services in Golang. Golang (or Go) is a newly released open-source general purpose programming language suitable for web service development. The paper represents a step-by-step guide of how to design and implement a RESTful service, capable of communicating with external database management systems and other REST APIs. To make description clearer the paper relies on real-life examples of a distributed SOA-based bank transfer management system.*

Keywords: *Golang, Go programming language, web services, RESTful services.*

ВЪВЕДЕНИЕ

Golang е нов програмен език, който все по-често се използва за създаване на REST (REpresentational State Transfer) услуги. Основното му предимство пред останалите езици в тази област е стриктността от гледна точка на получаваните данни, което от своя страна намалява възможността за хакерски атаки.

REST е архитектурен шаблон/стил, който дефинира набор от правила за комуникация между уеб услуги, базирана на HTTP протокола. Основната идея на REST е, че всеки обект или порция информация се третира като уникален ресурс, който може да бъде адресиран с уникален URI. Каква операция ще се извършва с посочения ресурс зависи от използвания HTTP метод. Уеб услугите, които съответстват на REST, наричат се още RESTful услуги, осигуряват оперативна съвместимост между различните (хетерогенни) компютърни системи в Интернет. Обикновено данните, които се обменят между клиента и сървъра при RESTful услугите се кодират във вид на JSON (JavaScript Object Notation) съобщения. JSON е текстово базиран отворен стандарт създаден за човешки четим обмен на данни. Състои се от прости структури от данни и асоциативни масиви, наречени обекти.

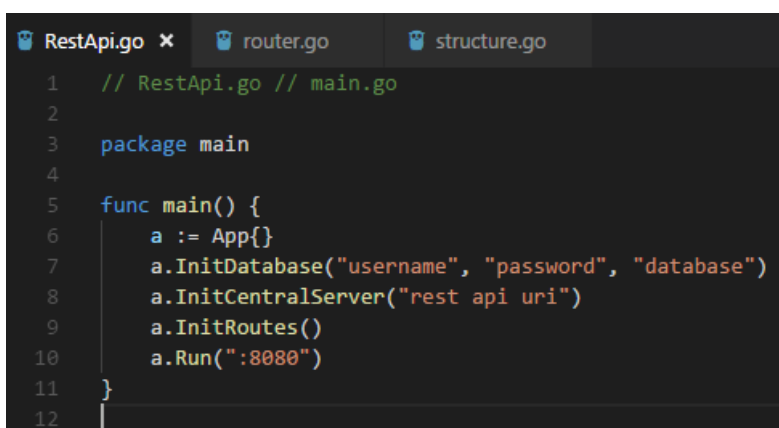
Този доклад има за цел да опише основните моменти при създаването на едно REST приложение с помощта на езика Golang. За демонстрация е избрано опростено приложение за извършване на междубанкови преводи, което дава възможност за проверка на данните за банкови сметки и осъществени транзакции, както и за извършване на вътрешнобанкови и междубанкови плащания. Валидацията на данните за сметките от други банки се осъществява от централен възел, наречен централна банка.

²⁶ Докладът е представен на студентската научна сесия на 27 април 2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: REST УСЛУГИ НА GOLANG.

Основни модули

В езика за програмиране Golang една програма може да вижда и да достъпва всички методи и структури, разписани във файлове, намиращи се в директорията на проекта. За по-ясно разграничаване на функционалността, различните методи могат да се обособят в три модула (файла): стартиращ, маршрутизиращ и структурен.

Първият модул (фиг. 14) извиква инициализиращите методи. С тяхна помощ се задава необходимата информация за комуникация с базата от данни (местоположение, име, потребителско име и парола), порт, на който слуша приложението, и адрес на другата REST услуга, към която се обръща. Golang приложенията се стартират винаги от main метода на go файловете, имащи същото име като папката, в която се намират. Поради тази причина стартиращия модул трябва има същото име като REST приложението.



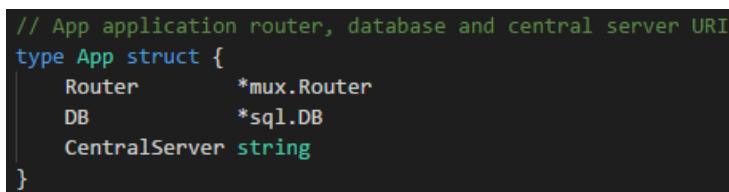
```

1 // RestApi.go // main.go
2
3 package main
4
5 func main() {
6     a := App{}
7     a.InitDatabase("username", "password", "database")
8     a.InitCentralServer("rest api uri")
9     a.InitRoutes()
10    a.Run(":8080")
11 }
12

```

Фиг. 14. Сорс код на стартиращия модул

Маршрутизиращият модул дефинира основните REST методи и осъществява прочитането и валидирането на постъпилите данни, препращането им към конкретни методи, които да вземат или актуализират определена информация в базата данни или друго REST приложение, както и връщането на подходящ отговор в зависимост от възникналите грешки при обработка на данните и от получените резултати. За целта се създава специална структура, която дефинира основните обекти в приложението (фиг. 15) – рутер на REST командите, обект за комуникация с базата от данни и URI на централната банка.



```

// App application router, database and central server URI
type App struct {
    Router      *mux.Router
    DB          *sql.DB
    CentralServer string
}

```

Фиг. 15. Дефиниране на основните обекти в приложението

Последният модул е структурният. В него се задава JSON структурата на обработваните заявки и връщаните отговори, шаблона на данните, подаван към базата от данни или вземан от нея, и структурата на подаваните и получаваните данни от други REST услуги (примерна структура е посочена на фиг. 16). Освен това в този модул са разписани и методите, осъществяващи достъп до базата от данни и REST услугата на Централната банка. При по-сложна структура или приложение с по-голяма функционалност този модул може да се раздели на три: описващ структурата на обменяните данни, дефиниращ методите за комуникация с базата от данни и съдържащ методите за комуникация с други приложения.

```
// Payment is one transaction
type Payment struct {
    IBANOrderer    string `json:"IBAN_orig"`
    IBANBeneficiary string `json:"IBAN_benef"`
    Amount         float64 `json:"amount"`
    Date           string `json:"date"`
    ReasonForPayment string `json:"reason"`
}
```

Фиг. 16. Примерна структура, използвана при комуникация с базата данни и описваща съответстващия JSON обект. В случая това е структура, описваща една транзакция.

Дефиниране на REST методите

За дефиниране на REST методите се използва обекта `mux.Router` от “github.com/gorilla/mux” библиотеката. За създаването на нов рутиращ обект се използва командата `mux.NewRouter()` (фиг. 17). С негова помощ се осъществява маршрутизацията. Задаването на URL на REST командата, заедно с изпълняващата го `go` функция, става с помощта на `HandleFunc()` метода на посочения по-горе обект. С `Methods()` се посочва HTTP метода на подаваната заявка.

```
// InitRoutes connect http command with golang command
func (a *App) InitRoutes() {
    a.Router = mux.NewRouter()
    a.Router.HandleFunc("/api/payments/", a.GetPayments).Methods("GET")
    a.Router.HandleFunc("/api/payments/{search:[a-zA-Z0-9]+}", a.SearchPayments).Methods("GET")
    a.Router.HandleFunc("/api/payments/", a.Pay).Methods("POST")
    a.Router.HandleFunc("/api/accounts/{IBAN:[a-zA-Z0-9]+}", a.IsValid).Methods("GET")
    a.Router.HandleFunc("/api/accounts/", a.GetAccounts).Methods("GET")
}
```

Фиг. 17. Създаване на маршрутизиращ обект и дефиниране на REST методите

Всеки от `go` методите, към който се препраща REST рутирация обект (например фиг. 18), включва два аргумента: `r` от тип `*http.Request` (съдържащ данните на подаваната заявка) и `w` от тип `http.ResponseWriter` (в него се записва отговора).

```
// SearchPayments get transactions by criteria
func (a *App) SearchPayments(w http.ResponseWriter, r *http.Request) {
    vars := mux.Vars(r)
    criteria := vars["search"]

    trans, err := SearchPayments(a.DB, criteria)
    if err != nil {
        RespondWithError(w, http.StatusBadRequest, err.Error())
        return
    }
    RespondWithJSON(w, http.StatusOK, trans)
}
```

Фиг. 18. Примерен метод изпълняващ REST команда. В случая се връщат всички транзакции по зададен критерий на търсене

С помощта на обекта `vars` (фиг. 19) от тип `map[string]string`, създаден чрез функцията `mux.Vars(r)`, се прочитат параметрите на заявката, подадени в URL адреса на командата. За декодиране на параметрите подадени в тялото на заявката се използва декодер.

```

var trans Payment
decoder := json.NewDecoder(r.Body)
if err := decoder.Decode(&trans); err != nil || trans.Amount <= 0 {
    RespondWithError(w, http.StatusBadRequest, "Invalid transaction data")
    return
}
defer r.Body.Close()

```

Фиг. 19. Примерна употреба на декодер на тялото на REST заявката. В случая се прочитат всички данни за една транзакция – бенефициент, наредител, сума и основание

За връщане на грешките и резултатите от обработката в подходящ HTTP формат се използват методите *RespondWithError* и *RespondWithJSON* (фиг. 20). Първият кодира грешката по подходящ начин, а вторият кодира всички резултати в JSON формат.

```

// RespondWithError translate error to json
func RespondWithError(w http.ResponseWriter, code int, message string) {
    log.Fatal(message)
    RespondWithJSON(w, code, map[string]string{"error": message})
}

// RespondWithJSON translate struct to json
func RespondWithJSON(w http.ResponseWriter, code int, payload interface{}) {
    response, _ := json.Marshal(payload)

    w.Header().Set("Content-Type", "application/json")
    w.Header().Set("Access-Control-Allow-Origin", "*")
    w.Header().Set("Access-Control-Allow-Methods", "GET, PUT, POST, DELETE, OPTIONS")
    w.Header().Set("Access-Control-Allow-Headers",
        "Content-Type, Content-Range, Content-Disposition, Content-Description")
    w.WriteHeader(code)
    w.Write(response)
}

```

Фиг. 20. Методи за обработка на резултатите и кодирането им в JSON формат

Връзка с базата от данни

За комуникация с базата от данни се използва **sql.DB* обекта от библиотеката “github.com/go-sql-driver/mysql”. Връзката с базата от данни се осъществява на два етапа.

При първия етап се инициализира (фиг. 21) с помощта на подадените от стартиращия модул данни – потребителско име, парола и име на базата от данни. Ако базата от данни не е локална, името трябва да включва и пътя до нея (IP адрес). При този етап се отваря и връзката между базата от данни и приложението.

```

// InitDatabase the database
func (a *App) InitDatabase(user, password, dbname string) {
    connectionString := fmt.Sprintf("%s:%s@/%s", user, password, dbname)

    var err error
    a.DB, err = sql.Open("mysql", connectionString)
    if err != nil {
        log.Fatal(err)
    }
}

```

Фиг. 21. Инициализация на базата от данни

Вторият етап се състои в подаването на конкретната заявка и обработването на данните ѝ (фиг. 22). Заявката се записва в обект от тип *string* и се подава в метод *Query* (при искане на

данни) или в метод *Ехес* (при добавяне, актуализиране или изтриване на данни). Отговорът на заявката се запазва в обект от тип **Rows* и се обработва с помощта на методите *Next* и *Scan*.

```
// GetPayments gets all transactions
func GetPayments(aDB *sql.DB) ([]Payment, error) {
    statement := "SELECT C_IBAN_Orderer, C_IBAN_Beneficiary, C_Value, C_Data, " +
        "C_Reason_For_Payment FROM t_transaction"
    rows, err := aDB.Query(statement)
    if err != nil {
        return nil, err
    }

    defer rows.Close()
    trans := []Payment{}

    for rows.Next() {
        var tr Payment
        if err := rows.Scan(&tr.IBANOrderer, &tr.IBANBeneficiary, &tr.Amount, &tr.Date,
            &tr.ReasonForPayment); err != nil {
            return nil, err
        }
        trans = append(trans, tr)
    }

    return trans, nil
}
```

Фиг. 22. Примерна заявка към базата от данни. В случая се взимат всички данни за осъществените транзакции.

Връзка с друго REST приложение

За връзка с друго REST приложение първо трябва да се създаде HTTP клиент, на който да се зададе максимално време за изчакване на отговор (timeout) (фиг. 23).

```
var myClient = &http.Client{
    Timeout: time.Second * 30,
}
```

Фиг. 23. Създаване на HTTP клиент

След това трябва да се сглоби URI адресът на REST командата. Той съдържа IP адреса на достъпваната REST услуга и крайната точка (endpoint) на метода (фиг. 24). Накрая командата се подава с помощта на методите *Get* или *Post* на HTTP клиента. За подаване на друг тип HTTP заявка може да се използва метода *Do*, на който се подава обект от тип **Request* като параметър. Трите метода връщат като резултат обект от тип **Response* (съдържащ отговора на заявката) и обект от тип *error* (съдържащ съобщението за грешка, ако е възникнала) (фиг. 24).

```
response, err := myClient.Get(url)
if err != nil {
    res.Result = "fail"
    res.Code = 500
    res.Message = "Problem with connection."
    return res, bank, nil
}
```

Фиг. 24. Подаване на HTTP REST заявка за проверка за валидността на банкова сметка и преглед на върнатия резултат за грешки

Прочитането на отговора се осъществява на три стъпки (фиг. 25): прочитане на съдържанието от суровия HTTP отговор в *string*, създаване на обект от тип **Reader* и създаване и прочитане на JSON обекта с помощта на **Decoder*.

```
htmlData, _ := ioutil.ReadAll(response.Body)
log.Print(string(htmlData))
readerRes := bytes.NewReader(htmlData)
decoderRes := json.NewDecoder(readerRes)
_ = decoderRes.Decode(&res)
if res.Result != "OK" {
    decoderRes.Decode(&res)
    return res, bank, nil
}
```

Фиг. 25. Прочитане на отговора, получен от друго REST приложение

ЗАКЛЮЧЕНИЕ

Основното предимство на този тип приложения е стриктното валидиране на входните данни, осъществявано още на ниво примитиви на програмния език. Това спестява време при писането на програмния код и намалява вероятността от хакерски атаки.

Друго важно предимство е това, че езикът е с отворен код. Това означава, че много бързо се отстраняват грешките в кода на методите на използваните библиотеки.

Основен недостатък е липсата на утвърдени учебници или курсове за начинаещи. Заедно с наличността на библиотеки със сходни функционалности (четене на JSON структури, комуникация с база от данни) тази липса води до затруднения при първоначално писане на програмен код. Различните особености на отделните библиотеки водят до наличието на множество примери, които често не са еквивалентни. Това от своя страна означава, че не всеки пример за търсената функционалност може да бъде реализиран с определена библиотека. Тази особеност също внася забавяне в писането на програмната логика. Поради това при първоначален допир до езика се препоръчва да се потърси помощ от програмист вече запознат с него. За тази цел могат да се използват уеб платформите stackoverflow.com, github.com и golang.org.

БЛАГОДАРНОСТИ

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- GitHub Official Website (2018), <https://github.com/>
- Golang Official Website (2018), <https://golang.org>
- Semaphoreci: Building and Testing a REST API in Go with Gorilla Mux and PostgreSQL (2018), <https://semaphoreci.com/community/tutorials/building-and-testing-a-rest-api-in-go-with-gorilla-mux-and-postgresql>
- Stack Over Flow Official Website (2018), <https://stackoverflow.com/>
- Wikipedia: JSON (2018), <https://en.wikipedia.org/wiki/JSON>
- Wikipedia: REST (2018), https://en.wikipedia.org/wiki/Representational_state_transfer

TIME PROTOCOL (RFC-868) SERVER WORKLOAD SOURCE ²⁷

Eng. Adelina Koleva, BSc – MSc Student

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Phone: 0895 790 669

E-mail: adelina.koleva94@gmail.com

Assoc. Prof. Milen Lukanchevski, PhD

Department of Computer Systems and Technologies,

University of Ruse “Angel Kanchev”

Phone: 0887 303 850

E-mail: mil@ieee.org

Abstract: Clock synchronization is one of the task distributed systems should solve. Time Protocol is standardized under RFC-868 Internet protocol for external clock synchronization. The protocol is a variant of the client-server model. There is a strong necessity of time server performance estimation.

The paper presents a workload source for time server performance tests – TSLS (Time Server Load Source). TSLS is specialized especially for Time Protocol servers but supports general techniques for TCP servers' performance evaluation. It is multithreaded and event-driven workload bench implemented under C++11 standard. Registers the key server performance metrics: Connection Setup Time, Server Response Time, Network Round-Trip Time, Number of Server Refusals, etc.

Keywords: C++11, Distributed Systems, Event-Driven, Multithread, Performance, RFC-868, Workload.

ВЪВЕДЕНИЕ

Една от задачите, която е необходимо да се решава при разпределените системи е синхронизацията на системните часовници на отделните съставлящи процеси (Lukanchevski, M., 2014, Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G., 2011). Различават се вътрешна и външна синхронизация. При вътрешната синхронизация системните процеси $P_0, P_1, \dots, P_{n-1}$ синхронизират помежду си показанията на часовниците си. Докато при външната синхронизация показанието $C_i(t)$ на часовника на процеса P_i се синхронизира с показанието $S(t)$ на външен времеви еталон. Един от стандартизираните в Интернет протоколи за външна синхронизация е *Time Protocol*, описан в *RFC-868*.

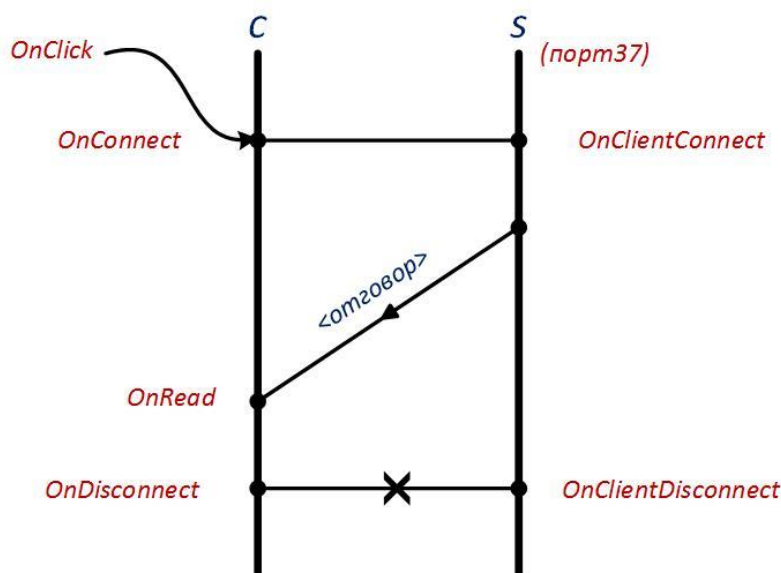
Протоколът представлява разновидност на модела клиент-сървър. Взаимодействието между клиента и сървъра се описва с комуникационната диаграма от фиг. 1. Времевият сървър S прослушва *TCP* порт 37. Синхронизацията се инициира от клиента C чрез изграждането на *TCP* връзка с порт 37 на сървъра. Това се възприема от сървъра като заявка, на която той отговаря с 4-байтово двоично съобщение. След получаването на отговора клиентът затваря връзката. На комуникационната диаграма са обозначени и събитията на сокетите, обслужвани от съответната страна.

В (Penkov, E., 2018) са предложени две реализации на *Time Protocol (RFC-868)* сървър: класическа еднозадачна (последователна) с опашка на чакащите заявки и многозадачна (паралелна). Многозадачната е изпълнена в два алтернативни варианта - с библиотечния клас *VCL::TThread* и класа *std::thread* от стандартната библиотека *C++11*. В експерименталната част на посочената работа е констатирано: „Определен интерес представлява поведението на многозадачната реализация на сървъра при голям брой множествени заявки – от няколко десетки до няколко хиляди. От една страна, така може да се получи по-точна оценка за

²⁷ Докладът е представен на студентската научна сесия на 27.04.2018 г. в секция „Комуникационна и компютърна техника“ с оригинално заглавие на български език: ИЗТОЧНИК НА РАБОТЕН ТОВАР НА TIME PROTOCOL (RFC-868) СЪРВЪР.

времеви характеристики на сървъра. От друга страна, може да се оцени максималният брой едновременни заявки, които сървърът може да поеме, както и влиянието на този брой върху реактивността на сървъра.“

Така се прави извода за необходимостта от специализиран източник на работен товар, където този товар включва пакет от заявки към *Time Protocol* сървъра. В представения доклад се разглеждат някои моменти от текущата разработка на подобен източник на множествени заявки (*TSLS, Time Server Load Source*).



Фиг. 1. Комуникационна диаграма на протокола при TCP сокет

ИЗЛОЖЕНИЕ

Етапи на работа

Сложността на работата предполага разбиването ѝ на няколко етапа (фиг. 2).

При разработката на *алфа прототипа* се тръгва от фиксиран адрес на времевия сървър, фиксиран пакет от 4 заявки и последователна (еднозадачна) обработка в събитията на сокетите (версия 0.1.0.x.TSLS). А в края на този етап се достига до възможност за преконфигуриране на параметрите и паралелна (многозадачна) обработка в събитието *OnRead* на клиентските сокет (версия 0.3.0.x.TSLS).

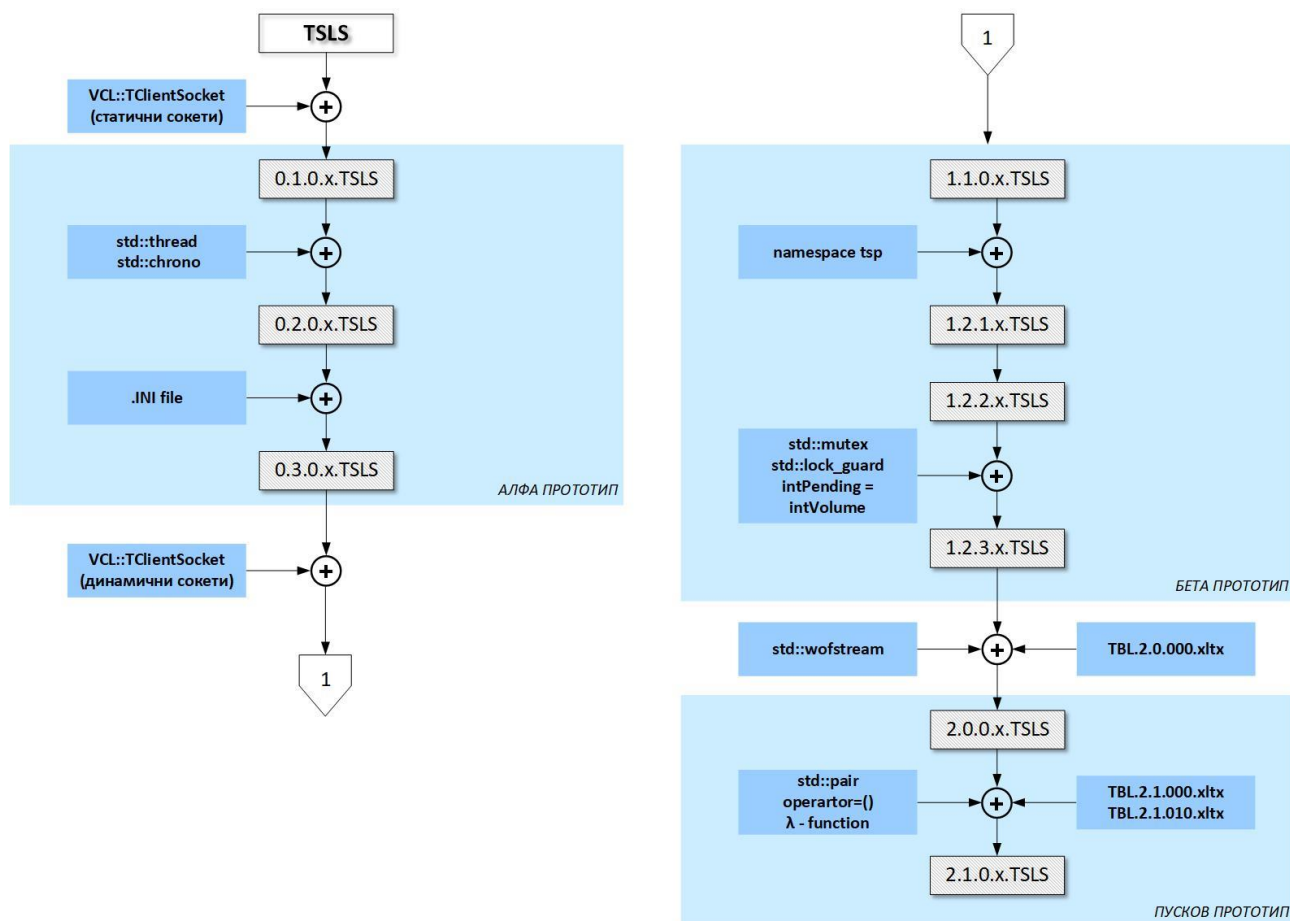
При приключването на *бета прототипа* (версия 1.2.3.x.TSLS) се отчитат не само двете крайни състояния на връзката по TCP сокетите (*Close, Open*), но и преходното състояние (*Transient, Half-Open*); въведен е параметър *Consecutive*, с който се задава последователно отваряне на сокет връзките; точността на извеждане на времето е разширена до 1 ms; допълнително е намалена инструменталната грешка чрез изнасяне на визуализацията на системния протокол след приключването на синхронизацията.

На етапа на *пусковия прототип* (версия 2.1.0.x.TSLS) се осигурява фиксиране на характеристикните времена при тестовите под товар, с оглед последващата обработка с *Excel*. Развит е *XLTX* шаблон за фиксиране и първична обработка на експерименталните резултати.

Последователното включване на необходимите базови средства (Josuttis, N., 2012) за осигуряване на посочената функционалност схематично е представена на фиг. 2.

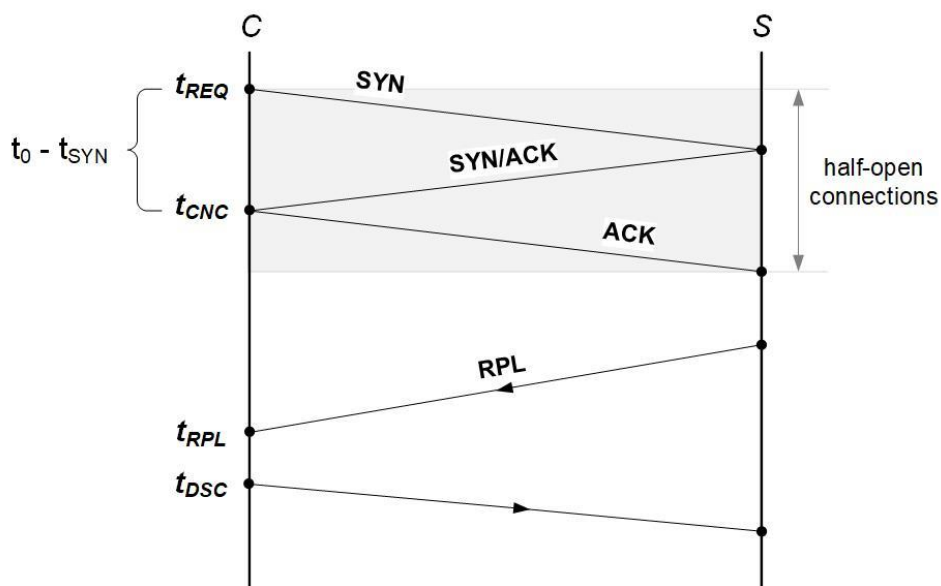
Метрики, характеристични времена, вътрешна организация

Ключовите метрики при TCP сървърите са: *времето за установяване на връзка* (Connection Setup Time), *времето за отговор на сървъра* (Server Response Time), *интервала от изпращане на заявката до получаване на отговора* (Network Round-Trip Time) и др. (Shanahan, Dan, 2015).



Фиг. 2. Етапност на реализацията

Обслужването на заявките от времевия сървър се описва с *характеристичните времена* от фиг. 3, които са директно следствие от комуникационния протокол (фиг. 1) и особеностите на *TCP* сокетите.



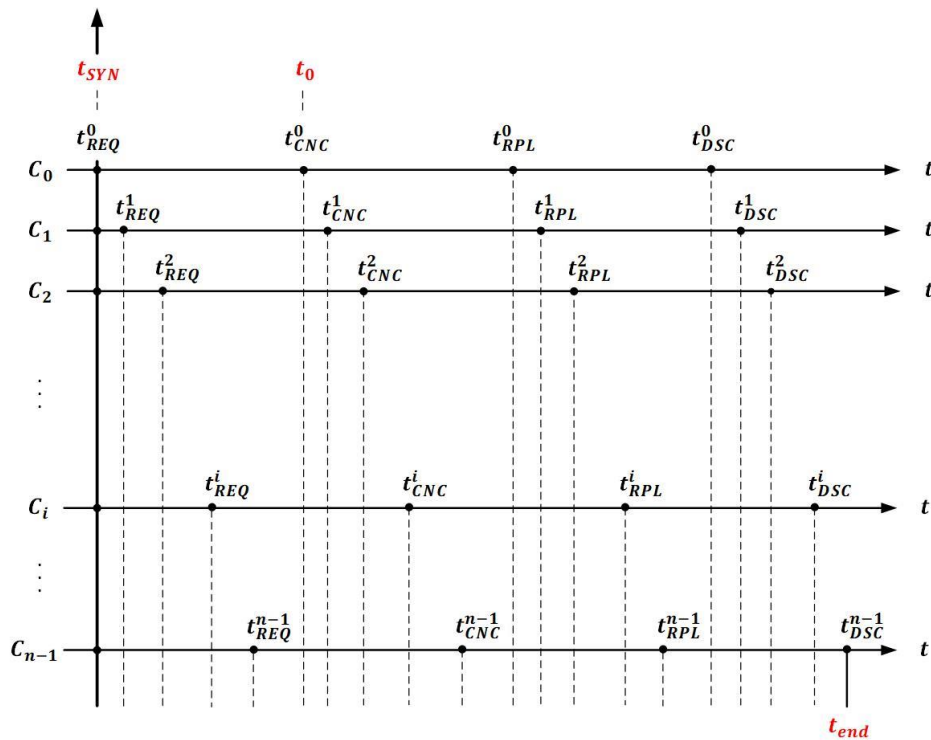
Фиг. 3. Характеристични времена

Докато на фиг. 3 са приведени характеристичните времена за отделно взет клиент, на фиг. 4 е представена цялостната времедиаграма при пакет множествовни заявки. За получаване на посочените ключови метрики за всеки един клиент C_i от пакета е необходимо да се фиксират моментите на формиране на заявка за синхронизация t_{REQ}^i , на получаване на тази

заявка от сървъра t_{CNC}^i , на получаването на отговора от сървъра в клиента t_{RPL}^i и на разпадане на връзката със сървъра t_{DSC}^i .

Началният момент на синхронизация t_{SYN} съвпада със стартирането на пакета множествени заявки, а крайният - t_{END} отговаря на момента на разпадане на последната връзка от пакета. Това не е задължително t_{DSC}^{n-1} , поради непредсказуемия асинхронен и конкурентен начин на установяване на връзките и обработката, т.е. $t_{END} = \max(t_{DSC}^i, t_{ERR}^i)$.

На фигурите не е посочен друг характерен момент – t_{ERR}^i при отказ от страна на сървъра да установи връзка с клиента C_i . Моментите на подобно отхвърляне на заявките, както и броя на тези отхвърляния - n_{ERR} също са част от метриките на сървъра и характеризират способността му да обслужва едновременно голям брой заявки.



Фиг. 4. Характеристична времедиаграма

Двойките характеристични времена формират съответните *характеристични периоди*, от които се изчисляват метриките на производителността на сървъра:

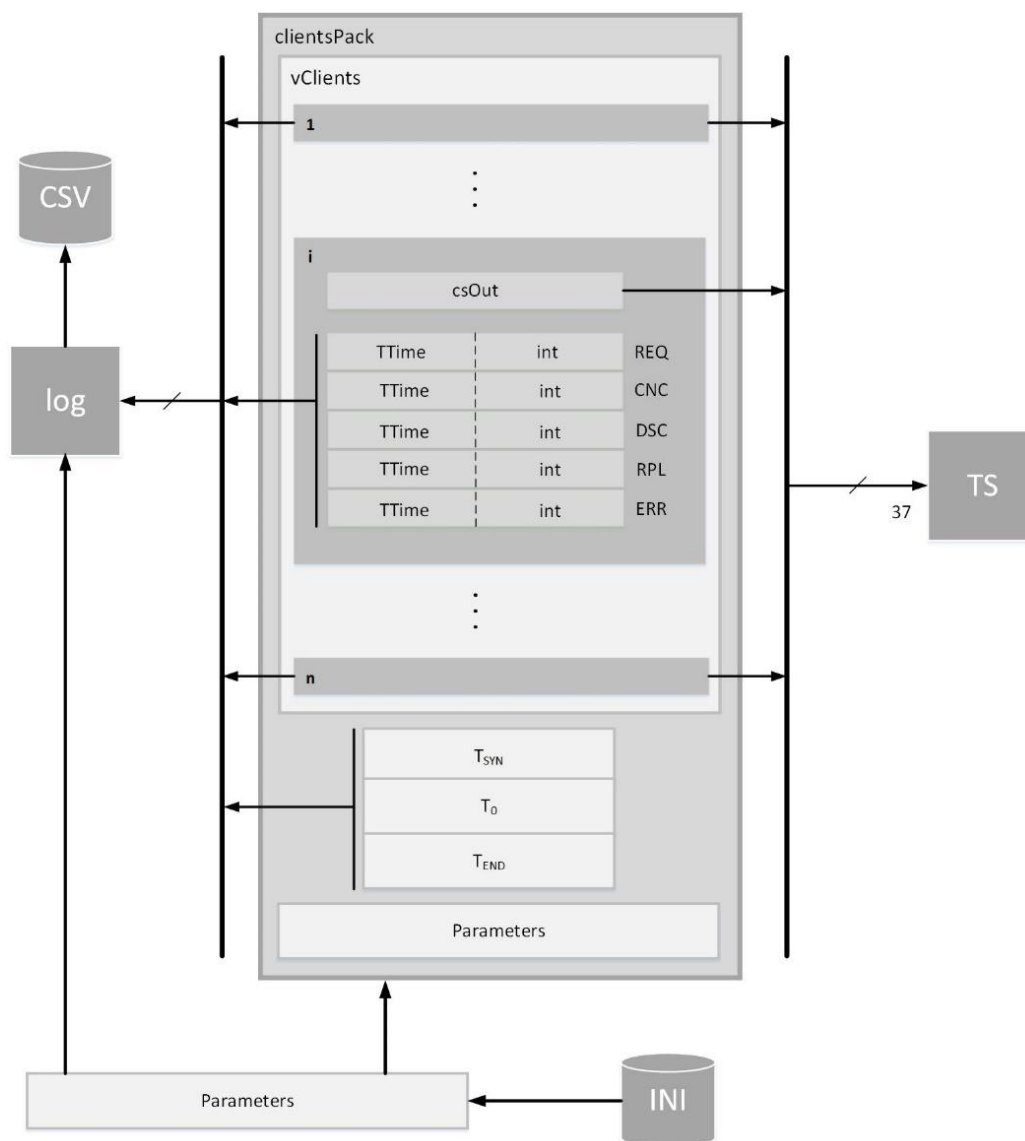
- периодът $[t_{REQ}, t_{CNC}]$ с абсолютна стойност $t_{CNC} - t_{REQ}$ отговаря на времето за установяване на връзка, а при конкретния протокол и на времето за изпращане на заявка;
- периодът $[t_{CNC}, t_{RPL}]$ с абсолютна стойност $t_{RPL} - t_{CNC}$ отговаря на времето за отговор на сървъра;
- периодът $[t_{REQ}, t_{RPL}]$ с абсолютна стойност $t_{RPL} - t_{REQ}$ отговаря на интервала от изпращане на заявката до получаване на отговора;
- периодът $[t_{SYN}, t_{END}]$ с абсолютна стойност $t_{END} - t_{SYN}$ отговаря на общото време на обслужване; производителността на сървъра $P = \frac{n - n_{ERR}}{t_{END} - t_{SYN}}$ е отношението на реалния брой обслужени заявки $n - n_{ERR}$ и общото време на обслужване.

Предложената структура на източника на множествени заявки *TSLs* е представена на фиг. 5. Въвежда се ново пространство на имената *tsp*, съдържащо декларациите на класовете *tsp::Parameters*, *tsp::Client*, *tsp::ClientsPack* и *tsp::log*.

Обект от класа *tsp::Parameters* съдържа конфигурационните параметри. Те се четат от конфигурационния *.INI* файл.

Класът *tsp::ClientsPack* включва вектора *vClients*, характеристикните времена t_{SYN} , t_0 , t_{END} и копие на конфигурационните параметри. Векторът *vClients* съдържа клиентите от пакета. Клиентите са от клас *tsp::Client*. Каналът *csOut* на всеки такъв клиент е от тип *VCL::TClientSocket* и служи за връзка с избрания времеви сървър *TS*. Клиентите фиксират индивидуалните си характеристикни времена t_{REQ}^i , t_{CNC}^i , t_{RPL}^i , t_{DSC}^i и t_{ERR}^i , заедно с общия брой на обслужваните в съответния момент клиенти.

За минимизиране на внасяната инструментална грешка не се поддържа протокол на екрана по време на измерванията, а се използва обект от специализирания клас *tsp::log* в паметта. Едва след приключването на измерванията съдържанието на този обект се извежда на екрана и фиксира в *.CSV* файл.



Фиг. 5. Структура на източника на множествени заявки

ИЗВОДИ

Представеният в доклада източник на множествени заявки може да се използва за цялостно изследване на конкретни реализации на *Time Protocol (RFC-868)* сървъри.

Интерес например представлява сравнението на обслужването при еднозадачна и многозадачна реализация на сървъра. При множествен пакет с обем $n = 50$ и зададено време на обслужване $DelayToFree = 1000\text{ ms}$ за еднозадачната реализация се получава

производителност $P' = \frac{n-n_{ERR}}{t_{END}-t_{SYN}} = \frac{50-23}{27,684} = 0,98 \approx 1$ заявка/s, а за многозадачната реализация - $P'' = \frac{n-n_{ERR}}{t_{END}-t_{SYN}} = \frac{50}{1,729} = 28,92 \approx 29$ заявки/s.

Друга възможност е снемането на зависимостта $P(n)$ между производителността на многозадачната реализация на сървъра и обема на заявките в пакета. При еднозадачната реализация може да се снесе зависимостта $n_{ERR}(n)$ между броя на отхвърлените от сървъра заявки и обема на пакета.

REFERENCES

- Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed Systems: Concepts and Design*. 5th Ed. – Boston: Addison-Wesley, p. 1008.
- Josuttis, N. (2012). *The C++ Standard Library: A Tutorial and Reference*. 2nd Ed. – Boston: Pearson Education, Inc.
- Lukanchevski, M. (2014). *Razpredeleni sistemi i algoritmi: teoria i praktika*. Ruse: Izdatelski tsentar na RU „A. Kanchev”, ISBN 978-619-7071-35-1, p. 213.
- Penkov, E. (2018). *Realizatsia na protokola za vanshna sinhronizatsia Time Protocol (RFC-868)*. // Diplomen proekt, RU „A. Kanchev”, p. 102.
- Shanahan, Dan (2015). *5 Key TCP Metrics for Performance Monitoring*. URL: <http://www.thevisiblenetwork.com/2015/04/12/5-key-tcp-metrics-for-performance-monitoring/> (Accessed on 13.03.2018).

PHOTO SCANNED TEXTURES²⁸

Nikolay Marinov, Student,

Department of Computer Systems and Technologies,
University of Ruse "Angel Kanchev"
Tel.: +359 88 534 6814
E-mail: nmarinov35@gmail.com

Assoc. Prof. Irena Valova, PhD

Department of Computer Systems and Technologies,
University of Ruse "Angel Kanchev"
Phone: 00359082888685
E-mail: ivalova@ecs.uni-ruse.bg

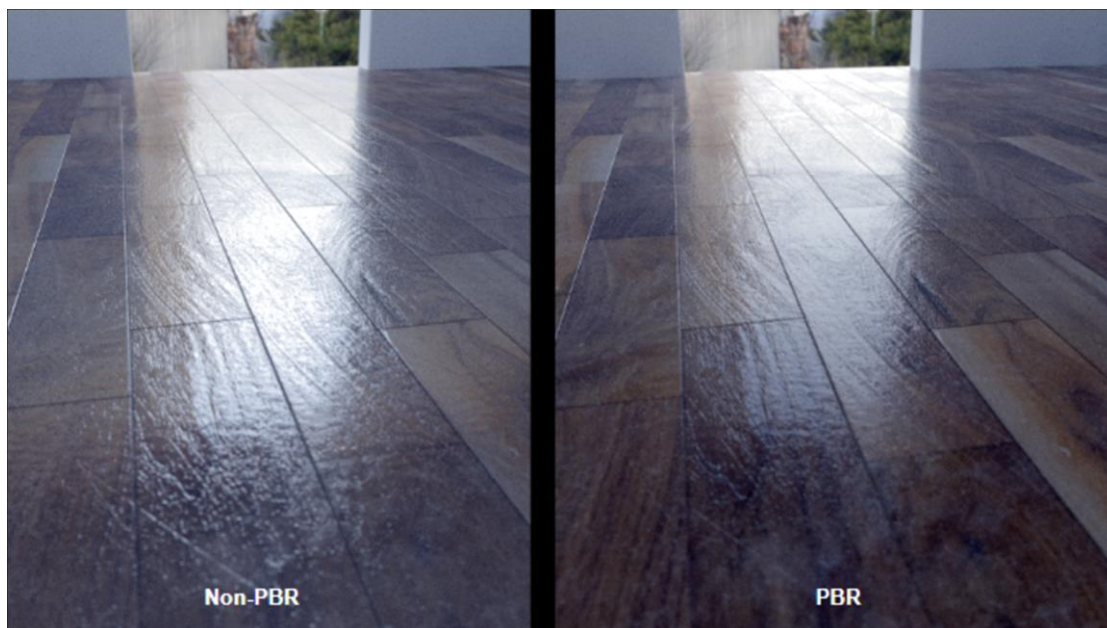
Abstract: *The paper reviews existing methods of creating photoscanned textures for pbr and real time rendering. There are some examples of extraordinary work and the different methods that you can achieve them. Different physical properties of different dielectrics and metals are shown as an example of what needs to be followed to create good renders. The ways photoscanned textures are created and the different texture maps used in the process.*

Keywords: *CGI, Phtoscan, Textures, 3D modeling*

ВЪВЕДЕНИЕ

С всеки ден CGI индустрията става все по-голяма. Причината за това е, че тази индустрия е навсякъде около нас започвайки от рекламите по телевизията до филмите в киното и видео игрите. Все по-големи и амбициозни стават тези проекти, което води до търсенето на повече специалисти в тази област и разнообразни решения за ускоряване на процеса на работа. Със всяка година стандарта за PBR (Photo-Realistic Based Rendering) (фигура 1) става все по-висок за това и има много фирми и стартапи, които се съревновават за да представят най-новите трендове в CGI сферата така, че да са полезни на обществото и да имат доста високи приходи от това. Едно от най-важните неща в CGI са текстурите. До скоро старомодния метод беше да се снимат текстурни и от тях чрез обработка да се изкарат нужните карти, но никой няма време да се занимава с това за това се направиха онлайн магазини за текстурни. Те са едно голямо решение, но в крайна сметка голяма част от тях просто представляват манипулирани изображения. Моето решение на проблема е онлайн магазин за фотосканирани текстурни (които са 99.9 % близки до реалния облик на обекта) до 16k (15360 x 8640 пиксела) резолюция, като следва те да са мащабируеми сетове т.е.- 512x512, 1024x1024, 2048x2048, 4096x4096, 7680x7680, 15360x15360. Като моето решение е съпоставимо със всички софтуери на пазара и към момента е от изключително важно за повечето хора работещи в тази сфера.

²⁸ Докладът е представен на студентската научна сесия на 27.04.2018 в секция Комуникационна и компютърна техника с оригинално заглавие на български език: ФОТОСКАНИРАНИ ТЕКСТУРИ.



Фиг. 1. Photo-Realistic Based Rendering текстури

ДРУГИ ПОДОБНИ РЕШЕНИЯ

Не съм първият измислил това решение, но мога да го представя така, че да бъде различно от другите подобни на него. Най-голямата компания занимаваща се с това е Quixel, като има два софтуера, които дават достъп до библиотека от фотосканирани обекти(асети). Разделя се на Quixel Suite и Quixel Megascans.

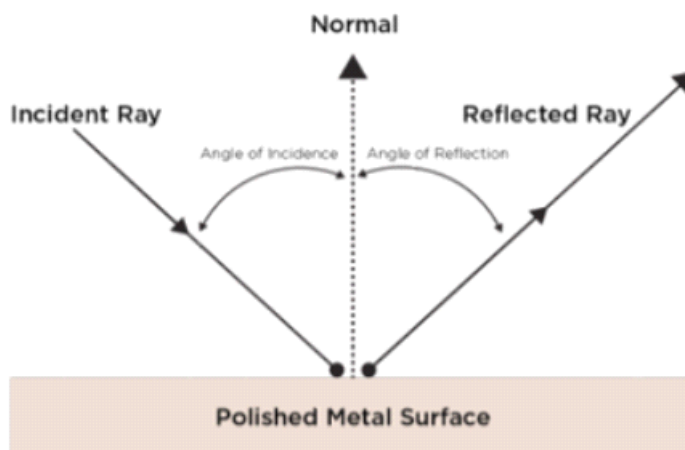
Най-добрата компания според мен е Friendly Shade. Създадена от Себастиан Запата от Колумбия. Той създава свой стартап със колекция от може би най-детайлните текстури в света, които достигат до мащабируемост от 64к пиксела. След доста опити от негова страна той намира най-правилния начин за направата на най-големите и детайлни PBR текстури и създава свой сайт за тяхната продажба. Когато от Quixel разбират за малката му компания те решават да му предложат 100 000\$ за компанията му, но той отказва и се очертва скоро да бъде една от най-големите компании занимаващи се с това тъй като вече има договор с фирми, като Audi, Infinity и други.

За да разберем как работят текстурите трябва първо да разберем как се държат различните метали и диелектрици в истинския свят подложени на различни условия.

- Physically-Based Rendering:

Светлината е сложно явление, което може да се разгледа в състояние на елементарна частица или като вълна. В резултат на това различни модели за работа са създадени. Светлината е важна, защото на теория във виртуалния свят преди лъчът светлина да стигне до обекта той не съществува. Нашето възприемане на един обект е отразената от него светлина достигаща нашето око. За това колкото по добре разбираме как работи светлината в истинския свят и как тя взаимодейства с дадена повърхност толкова по-добре можем да изградим нашите текстури.

- Light Rays:



Фиг. 1. Модел на светлината

Моделът за светлина (фигура 2) се описва, като лъч с формата траектория на права отсечка в хомогенна прозрачна среда подобна на въздуха. В този модел също се описва, че преминавайки в непрозрачна или по плътна (като преминаването то въздух във вода) среда траекторията ще се изкривява, както би се случило в истинския свят базирано на математически уравнения. Лъчът, който пада към дадена повърхност се нарича лъч на падане, а ъгъла между него и отразения се нарича съответно ъгъл на падане. При падане на лъча могат да се наблюдават две събития:

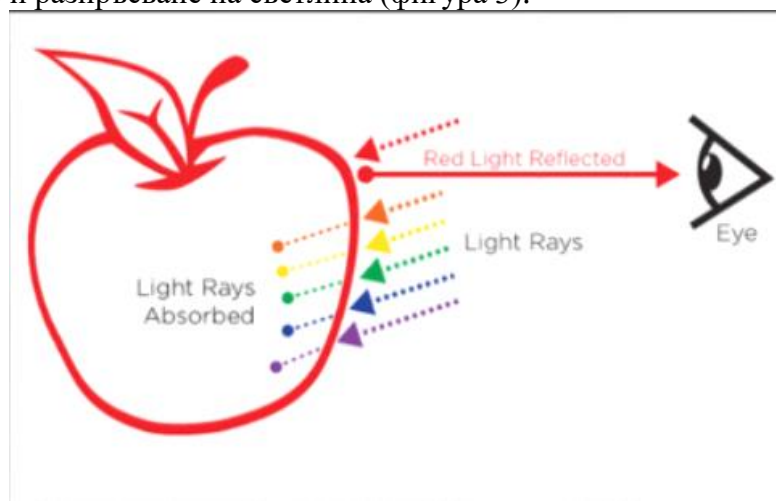
1. Падащия лъч се отразява от повърхността и отива в друга посока. Следвайки закона за отражението: ъгъла на отражение е равен на ъгъла на падащия лъч. Това събитие се нарича пълно отражение на светлината.

2. Падащият лъч преминава в друга среда без да променя своята траектория. Това събитие се нарича рефракция на светлината.

При рефракцията наблюдаваме промяна на скоростта и посока (в някои случаи) на светлината зависещо пряко от индекс на рефракцията IOR.

IOR представлява оптическа мярка която описва промяната на посоката на светлината преминаваща през дадена среда. Като IOR при водата има стойност 1.33, на стъклото 1,500, на леда 1.301, на кристал 2.000, на човешкото око (корнията) 1.380 и на ириса 1.410 и т.н.

- Абсорбиране и разпръскване на светлина (фигура 3):



Фиг. 3. Абсорбиране и разпръскване на светлина

Когато светлината преминава през нехомогенна или прозрачна среда светлината може да бъде абсорбирана(погълната) или разпръсната.

Когато светлината бъде погълната, силата на светлината намалява и се превръща в друга състояние най-често топлина. Цвета на светлината също се променя, като това зависи пряко дължината на вълната, въпреки това траекторията на падащия лъч никога не се променя.

Абсорбирането на светлина може да стане само под повърхността.

Когато светлината бъде разпръсната, посоката на лъчите става произволна отклонявайки се от вече установените закони за отражение и рефракция в зависимост от материала върху, който лъча пада. При разпръсването на светлина лъчът променя посоката си произволно, но силата на светлината не се променя.

Добър пример за това е човешката кожа. Там където е по-тънка като при пръстите или ухото при достатъчно силна светлина наблюдаваме как става червена и виждаме светлината от другата страна.

- Запазване на енергия в дадено състояние:

Законът за запазване на енергията гласи, че отразената светлина не може да бъде по-ярка от тази на падащия лъч върху повърхността на обекта.

- Френел ефект (фигура 4):

Открит е от френският физик Френел. Това явление представлява, че един обект погледнат от различно място отразява светлина по различен начин. Тоест падащия лъч достига до нашето полезрение под различен ъгъл.



Фиг. 4. Френел ефект

- Различните Текстурни карти:

1. Albedo:
2. Diffuse:
3. Roughness:
4. Glossy:
5. Normal:
6. Ambient Occlusion:
7. Specular:
8. Height:
9. Displacement:
10. Metallic:
11. Bump:
12. Curvature

ОПИСАНИЕ НА РЕАЛИЗАЦИЯТА

- Шейдъри и тяхното ютилизирание:

Шейдърът представлява процедурно генерирана текстура върху обект без да се ползват координати за разгъване. Заема повече място и ресурс, но ускорява начина на работа. За предпочитане е за хората правейки обикновенни рендъри.

- Процес на фотосканиране:

Използват се много на брой снимки за генериране на полигонен обект. Този обект след това се генерира като твърдо тяло върху, което се генерират текстурите. След което се генерира всяка отделна текстурна карта върху нов обект.

ИЗВОДИ

Намира приложение във филмовата индустрия и във видео игрите, като нов и алтернативен метод за PBR методи на работа.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

Polycount.com

Blender Guru

Chamferzone.com

3D GAME “INSIDE THE DREAM”

Denislav Kolev, BSc

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Tel.: 00359878 52 16 11
E-mail: denislav.vv@mail.bg, denislav.vv@icloud.com

Assoc. Prof. Irena Valova, PhD

Department of Computer Systems and Technologies,
University of Ruse “Angel Kanchev”
Phone: 00359082888685
E-mail: ivalova@ecs.uni-ruse.bg

Abstract: The article presents no intricate fun game that helps to unload the player or develop skills for quick reactions, tracking multiple objects at once, and acting wisely. The game players are very nice and fun.

Keywords: Unity3D, games

ВЪВЕДЕНИЕ

Unity3D е междуплатформен игрови двигател, разработен от Юнити Технолоджис и използван за разработка на видеоигри за компютри, конзоли, мобилни устройства и сайтове. В началото е обявен само за OS X, на световната конференция за програмисти на Епъл, през 2005 г. Оттогава е разширен за повече от петнадесет платформи, основният инструмент за разработка на софтуер (SDK) за Wii U. Юнити се отличава с възможността да разработва игри за различни платформи. В един и същ проект, разработчиците имат контрол върху доставката до мобилни платформи, мрежови четци, компютри и конзоли. Платформите които поддържа включват BlackBerry 10, Windows Phone 8, OS X, Android, iOS, Unity Web Player (включително Фейсбук), PlayStation 3, PlayStation 4, PlayStation Vita, Xbox 360, Xbox One, Wii U, Nintendo 3DS line и Wii. Включва сървър с подобрения и физика на NvidiaPhysX.

*Повече информация за самия engine може да се намери на адрес: <https://unity3d.com>.
А за различните продукти, които предлагат на <https://store.unity.com/>.*

Основните типове/жанрове игри са:

- *Стратегическа симулация или така наречената стратегия* – игра в, която потребителят управлява дадени ресурси. Често срещани стратегии са исторически симулации, митологии и фантастични светове.
- *Екшън* – прозилиза от английското “**action**”, което се превежда “**действие**“. Тук главното е противопоставянето на играча срещу опоненти (един или повече). Екшън жанровете се делят и на „екшън от първо лице“ и „екшън от трето лице“. При „екшън от първо лице“ - гледната точка на играчът съвпада с тази на компютърният герой, като от последния се вижда най-много ръката/цете държаща/щи оръжието, докато при „екшън от трето лице“ – с камерата се наблюдава героят в цял план. - *Ролева игра* - игра, в която *играчът поема управлението на герой или група герои*, като всеки от тях притежава индивидуални статистики и умения, които се развиват и обновяват

- **Приключенска игра** - този тип игри предполага наблягане на историята. Играта протича в диалози с героите от света на играта и събиране на различни предмети и пособия. Тук се изпълняват и различни типове „задачи“.
- **Симулативна игра** - игра, която симулира процес или *дейност от реалния свят*. Като на моменти се доближава до стратегическите игри
- **Спортна игра** - това е игра, *в която се играе някакъв спорт*. Целта е да се постигне победа с уменията на един играч. Тук също има игри, в които се игра от първо лице и от трето лице.

Наред с изброените жанрове игри, *InsideTheDream* се вписва към „Екишън“ игрите и по-конкретно към „*екишън от трето лице*“.

Както при този жанр игри и тук потребителят управлява герои, който трябва да се защитава максимално добре от нападащите го противници и да оцелее. Разликата идва в това, че *InsideTheDream* е и “survival game” или така наречената „оцелей“ тип игра. На фона на други *екишън игри от трето лице*, които достигаш някакъв брой точки или изпълниш определен брой задачи и преминеш към следващо ниво, тук шанса ти е веднъж. Какво ще рече това?

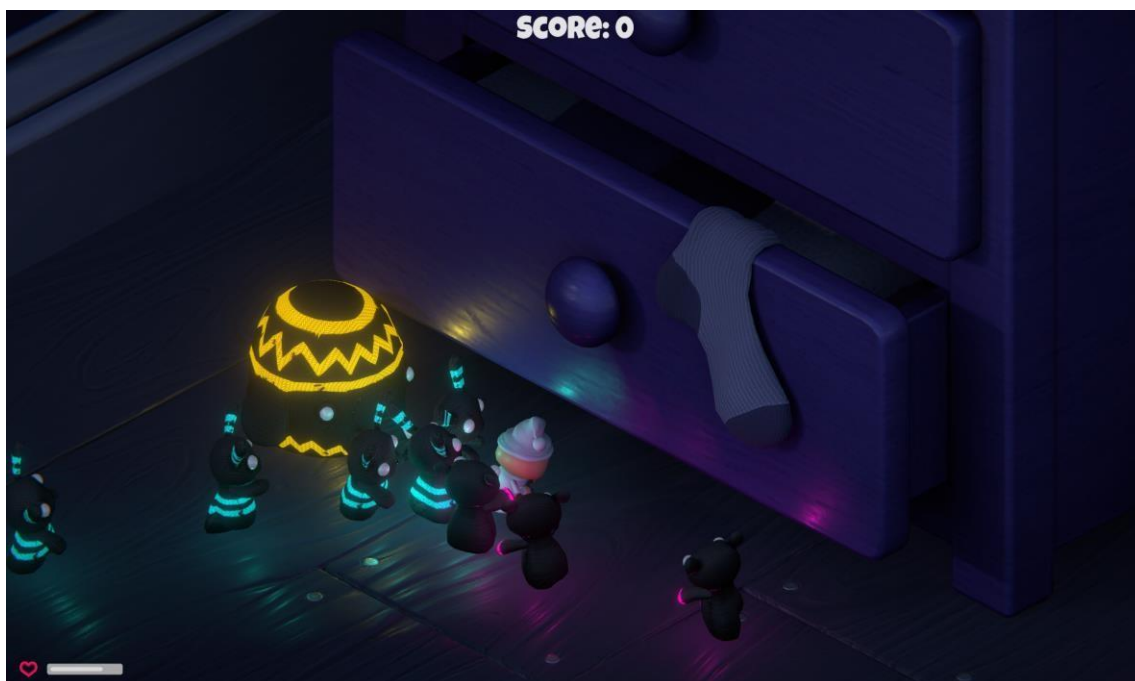
Разполагаш с определено количество живот, разбира се това количество е 100%. Докато не изгубиш целият си живот (100%) ти трупаш точки, които са показател за това колко време си се задържал на бойното поле (респективно повече точки – повече време).

След като изгубиш живота си играта се прекратява и рестартира (0 точки, 100%).

В играта противниците са три вида зомбита – зайци, мечки и слонове.

Всяко зомби ти носи различен брой точки. Респективно зайци-зомбита и мечки-зомбита – 10. Слон – 50.

За да се добие представа за визията на играта - фигура 1.



Фиг.1 Как изглежда InsideTheDream

ТЕХНИЧЕСКО ОПИСАНИЕ:

Играта е главно базирана за:

- Windows (7,8,10)

С течение на времето ще бъде cross-platform:

- iOS Android
- Windows Phone

Както по-горе е споменато, играта е направена изцяло на Unity3D, като по-основни assets са:

Audio – Аудиозвуците за играта (shooting, death, damage dealt, background music, gameovermusic) с различните за Unity 5 + ефекти от Mixer, Effects, Music

Camera – Съдържа скрипт за следенето на играча от камерата

Fonts – Използваме точно определен шрифт, отличаващ се от съществуващите – наречен LuckiestGuy Models – Героите, които участват (Player, Zombear, Zombunny, Hellephant) и Environment

Scenes – Сцената, на която работим, в случая е една единствена - Level01

Scripts – скриптове за отделните Enemy, Player и за Manager

Environment - компонентите (Arches, Bat, Blox, Clock, DollArm, Dollhouse, Drawers, Firetruck, Floor, Hearse, Robot, SpinningTop, Stool, Train, Wall).

СЪЩЕСТВУВАЩИ РЕШЕНИЯ:

Като цяло, няма голям брой до такава степен известни игри, които вече съществуват, но една която се доближава, до **InsideTheDream** и дори я доразвива е **“Surviveland”**.

Подобно на InsideTheDream е играта **„Surviveland“**, която може да се преведе “оцеляване“. Тук отново жанра е **„екшън от трето лице“**, отново управляваш герои, и да оцеляваш, но до тук с еднаквостта. Различното в **„Surviveland“** е, че разполагаш с много по-голям набор от опции.

Какво ще рече това? Много просто – имаш възможност да събираш ресурси (дърво, метал и т.н.), с които можеш да строиш, имаш различни видове оръжия и капани. Убиваш животни, които те нападат, защото реално ти се намиращ на някакъв остров, а те се явяват противниците ти. Събираш месо, с което се храниш. Точковата система е на база, колко дни си оцелял, които се броят в долният десен ъгъл. Всичко това е много добре имплементирано в геймплея и в самата история и прави играта една идея малко по-атрактивна и увлекателна, тъй като действията не се повтарят (стреляне по противници), а има нотка разнообразие. Също е важно да се отбележи, че в **“Surviveland”** площта на развиване на самото действие е по-голяма, така наречената „map“ (на бълг. „карта“).

Понеже играта е в Alpha version, така изглежда част от играта. Като ще прикача й линк към preview в youtube.com, за по-добра представа как работи играта.



Фигура 2 “Surviveland”

Линка към „Survivalend“ е: <https://www.youtube.com/watch?v=PC2mjd8su3I>. Като то е в поредица от няколко клипа с различните опции на играта.

В *InsideTheDream* потребителя може да извършва различни типове действия. Понеже това е „*shooter*“ игра, той може да се оглежда с помощта на мишката в обхват от 360 градуса, за да има възможност да се защитава от всеки един ъгъл. Наравно с това, за да се защитава той стреля също с помощта на мишката, чрез натискане на левият бутон. За движението се изисква натискане на клавишите на клавиатурата a, s, d, w (ляво, надолу, дясно, нагоре). Извън геймплея, потребителя може да паузира играта, чрез натискане на клавиша esc, и да се върне в игра, когато пожелае от същата позиция. Чрез натискането на esc клавиша, се влиза и в самото меню на играта, което дава редица възможности, като активиране/ деактивиране на звуците в играта и тяхното регулиране и Изход от играта.

Част от скриптовете за движение на героят:

```

.. using UnityEngine;
using UnitySampleAssets.CrossPlatformInput;

namespace CompleteProject
{
    public class PlayerMovement : MonoBehaviour
    {
        public float speed = 6f; // The speed that the player will
        move at.

        Vector3 movement; // The vector to store the direction of the player's movement.
        Animator anim; // Reference to the animator component.
        Rigidbody playerRigidbody; // Reference to the player's rigidbody.
        #if !MOBILE_INPUT
        int floorMask; // A Layer mask so that a ray can be cast just at gameobjects
        on the floor Layer. float camRayLength = 100f; // The length of the ray from the camera
        into the scene. #endif
        void Awake ()
        {
            #if !MOBILE_INPUT
            // Create a Layer mask for the floor Layer.
            floorMask = LayerMask.GetMask ("Floor");
            #endif
        }
    }
}

```

```
// Set up references.
anim = GetComponent <Animator> ();
playerRigidbody = GetComponent <Rigidbody> (); ..”
```

Част от скриптовите за джеиение на опонентите:

```
„ . . . using UnityEngine;
using System.Collections;

public class EnemyMovement : MonoBehaviour
{
    Transform player;
    PlayerHealth playerHealth;
    EnemyHealth enemyHealth;
    UnityEngine.AI.NavMeshAgent nav;
    void Awake
()
{
    player = GameObject.FindGameObjectWithTag ("Player").transform;
    playerHealth = player.GetComponent <PlayerHealth> ();
    enemyHealth =
    GetComponent <EnemyHealth> ();
    nav = GetComponent
<UnityEngine.AI.NavMeshAgent> ();
}

    void Update ()
    {
        . . . „
```

ВЪЗМОЖНОСТИ ЗА РАЗВИТИЕ И ИЗВОД:

Играта, макар и не толкова интензивна, богата на различни опции, нива, като подобни на нея игри, е разтоварваща и забавна.

Бъдещото ѝ развитие би могло да бъде в посока на:

- Различни левели.
- Постижения с награди (облекло на героя, различен тип оръжия).
- Класация.
- Възможност за мултиплеър.

ACKNOWLEDGMENTS

Този доклад се публикува със съдействието на проект 2018-EEA-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

<https://forum.unity.com/forums/game-design.71/>
https://www.udemy.com/devslopes-unity3d/?utm_source=adwords-learn&utm_medium=udemyads&utm_campaign=NEW-AW-PROS-TECH-ROW-DSA1-EN-EURO.ci.sl.ENG.vi.TECH.sd.All.la.EN.&utm_content=deal4584&utm_term=.aq.57696320151.ad.267562974153.de.c.dm.pl.ti.aud-325382095161:dsa-304639796103.li.9040136.pd.&qclid=CjwKCAjw8nXBRAiEiwAXWe2yVdd93FMLObtZTGV4tCYOCJWTtZW6Fd7gqEFToVd8KurhCotTmqUJBoC2SIQAvD.BwE

THE KNIGHT 3D GAME²⁹

Nikola Rashkov, BSc

Department of Computer Systems and Technologies,
University of Ruse "Angel Kanchev"
Tel.: 00359884 600 585
E-mail: nrashkov@icloud.com

Assoc. Prof. Irena Valova, PhD

Department of Computer Systems and Technologies,
University of Ruse "Angel Kanchev"
Phone: 00359082888685
E-mail: ivalova@ecs.uni-ruse.bg

Abstract: The idea of the "Knight" project is to create a 3D game for fun. RPG and MMORPG games are used to create realistic movements of the hero, its impact on the environment and the various objects within it.

Keywords: MMORPG, games

ВЪВЕДЕНИЕ

Идеята на проекта „The Knight“ е създаването на 3D игра с цел забавление.

Използват се похвати, известни при RPG и MMORPG игрите, създаващи реалистичните движения на героят, въздействието му спрямо средата и различните обекти в нея.

Контроли за движение/атака:

Движението на героят в самата среда се извършва с помощта на мишката, чрез кликове определяме координатите където героят ще бъде изместен. С помощта на бутоните от 1 до 6 изпълняваме магиите с които нанасяме щета на противника. (фигура 1)

Интерфейсът на героя съдържа:

- Бар с неговите жизнени точки, които той губи или възстановява.
- Няколко различни магии, спомагащи битката му с враговете и така нареченият Бос.



Фиг. 1. Интерфейсът на героят.

²⁹ Докладът е представен на студентската научна сесия на 27.04.2018 в секция Комуникационна и компютърна техника с оригинално заглавие на български език: THE KNIGHT 3D игра.

СЪЩЕСТВУВАЩИ РЕШЕНИЯ

World of Warcraft (фигура 2):

Това е продукт известен на голяма част от гейминг обществото, развит във всички аспекти и истински пример за MMORPG игра. Предлага множество от герои с различни характеристики, уникални сами по себе си. Играта предоставя гъвкав интерфейс с възможност за подреждане на магиите и добавяне на допълнителни барове. Разнообразие от анимации, огромна среда на действие и абсолютен контрол над героят. Играта предоставя и различни трудности, мисии, дори и професии, с които “героите” могат да се занимават.

В основата на всичко седи leveling системата, която отключва постепенно всички възможности на играта.

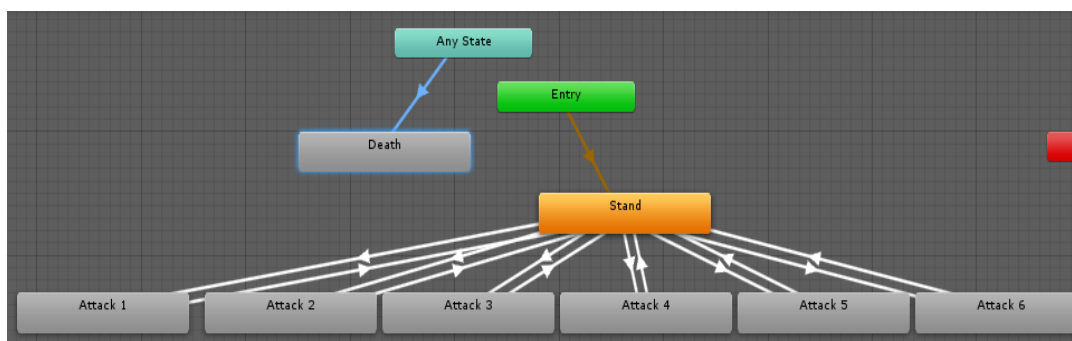


Фиг. 2. World of Warcraft gameplay.

ГРАФ НА СЪСТОЯНИЯТА

Съществува едно основно състояние, съответстващо с определена анимация, всяко действие се определя от предното, като възможността за активирането на едно от следните 6 зависи от това, че героят трябва да бъде в основното Stand състояние. Героят има възможност за развитие на движенията от едно към друго, без задължителното преминаване в Stand. На снимката на фигура 3 виждаме граф на действие от Stand в Attack(1 – 6). Както и възможността за Death състояние независимо от сегашното активно.

Част от скриптът за изпълнение на Attack: Е показан на (фиг.4)



Фиг. 3. Граф на състоянията

```
if (Input.GetKeyDown (KeyCode.Alpha1)) {
    playerMove.TargetPosition = transform.position;

    // fadeImages[0] meaning image thats at index 0 e.g. the first image
    if (playerMove.FinishedMovement && fadeImages [0] != 1 && canAttack) {
        fadeImages [0] = 1;
        anim.SetInteger ("Atk", 1);
    }
}
```

Фиг. 4. Скрипт за атака.

Движението на героя спрямо кординатите, подадени с мишката се извършват с помощта на показания на фигура 5 скрипт.

```
void MoveThePlayer() {
    if (Input.GetMouseButtonDown (0)) {

        //colider hit point in scrpx bla bla

        print ("Point in Screen pixels" + Input.mousePosition);

        Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);
        RaycastHit hit;

        if (Physics.Raycast (ray, out hit)) {
            if (hit.collider is TerrainCollider) {

                player_ToPointDistance = Vector3.Distance (transform.position, hit.point);

                //collier hit bla bla
                print ("Point in world" + hit.point);

                if (player_ToPointDistance >= 1.0f) {
                    canMove = true;
                    target_Pos = hit.point;
                }
            }
        }
    }
}
```

Фиг. 5.Скрипт за движение.

Игра(gameplay): Самото действие се развива на гробището, като главният герой е рицар, който бива възпрепятстван от редица трудности, биейки се с полумъртвите си врагове излезли от някой гроб. (фиг.6)



Фиг.6. Сцена на действието от процес на игра.

Интерфейс: На (фиг.6) виждаме също и:

1. Зелен бар за кръвта на героя (доло до лявото лъвче);
2. Бар с магии от 1 до 6, като всяка магия спомага за различни неща(1.Огнена атака, 2. Ритник, 3. Огнен лъч, 4. Огнен щит, 5. Лекуване и 6. Буря), като Бурята е най-силната магия с най-дълъг период на изчакване(CD) за следващо активиране.

Сцена:Сцената показана на горната фигура (фиг.6) представлява поле с гробове, оград, различни предмети, надгробни плочи, паметници и различни статуи.

ВЪЗМОЖНОСТИ ЗА РАЗВИТИЕ:

The Knight има възможност за добавяне на допълнителни сцени, герои, предмети, системи за: мисии, leveling , инвентар, чат/гласов чат и др. С развитието на историята, могат да бъдат издадени различни експанжии предоставящи нови мисии, врагове, трудности и похубави предмети.

ACKNOWLEDGMENTS

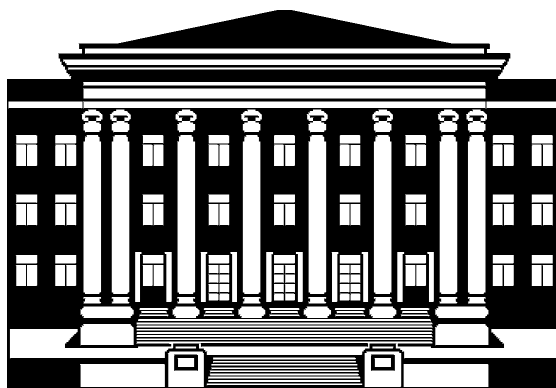
Този доклад се публикува със съдействието на проект 2018-ЕЕА-01 „Методи за събиране, организация, достъп, обработка и анализ на големи обеми от структурирани и неструктурирани данни“, финансиран от фонд „Научни изследвания“ на Русенски университет „Ангел Кънчев“.

REFERENCES

- <https://docs.unity3d.com/ScriptReference/CharacterController.Move.html>
- <https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial/moving-player>
- <https://answers.unity.com/questions/185226/c-movement-script-basic.html>
- <https://unity3d.com/learn/tutorials/topics/multiplayer-networking/creating-player-movement-single-player>
- <https://www.udemy.com>
- <https://stackoverflow.com/questions/44845580/attack-enemy-in-front-of-character>
- <https://answers.unity.com/questions/728518/making-my-player-attack-and-do-damage-when-button.html>
- <https://answers.unity.com/questions/619577/melee-attack-c-script-he.html>
- <https://forum.unity.com/threads/rpg-class-job-structure-c.396721/>
- <https://codereview.stackexchange.com/questions/100886/rpg-character-with-fluent-interface>

UNIVERSITY OF RUSE „ANGEL KANCHEV“

UNION OF SCIENTISTS - RUSE



**58-TH ANNUAL SCIENTIFIC CONFERENCE
OF UNIVERSITY OF RUSE „ANGEL KANCHEV“
AND UNION OF SCIENTISTS - RUSE**

24-25 OCTOBER 2019

I N V I T A T I O N

**Ruse, 8 Studentska str.
University of Ruse
Bulgaria**

PROCEEDINGS
Volume 57, Series 3.4.

**Electrical Engineering, Electronics and Automation
&
Communication and Computer Technologies**

**Under the general editing of:
Assoc. Prof. Milko Marinov, PhD**

**Editor of Volume 57:
Prof. Diana Antonova, PhD**

**Bulgarian Nationality
First Edition**

**Printing format: A5
Number of copies: on-line**

**ISSN 1311-3321 (print)
ISSN 2535-1028 (CD-ROM)
ISSN 2603-4123 (on-line)**

The issue was included in the international ISSN database, available at <https://portal.issn.org/>.
The online edition is registered in the portal ROAD scientific resources online open access



**PUBLISHING HOUSE
University of Ruse "Angel Kanchev"**