ИЗВЕСТИЯ

НА СЪЮЗА НА УЧЕНИТЕ - РУСЕ / ТОМ 15
ISSN 1311-106X
2018

СЕРИЯ 1
ТЕХНИЧЕСКИ НАУКИ

СЪЮЗ НА УЧЕНИТЕ - РУСЕ

ИЗВЕСТИЯ

НА СЪЮЗА НА УЧЕНИТЕ - РУСЕ

Серия 1

ТЕХНИЧЕСКИ НАУКИ

Том 15, 2018

PROCEEDINGS

of the Union of Scientists Ruse

Book 1

TECHNICAL SCIENCES

Volume 15, 2018

ИЗВЕСТИЯ
НА СЪЮЗА НА УЧЕНИТЕ - РУСЕ

Известията на Съюза на учените - Русе е научно списание, което се издава от 1998 г. Годишно списанието излиза във вид на няколко серии. Основни езици са български и английски, с тенденция постепенно да се премине към цялостното му отпечатване на английски.

Всяка серия се подготвя от съответния редакционен съвет, одобрен от Управителния съвет на СУ – Русе. **Редакционният съвет** урежда всички организационни и технически въпроси, свързани с издаването, одобряването на рецензентите и научната редакция на постъпващите материали.

За публикуване се приемат научни и научноприложни статии, съобщения, обзори с конкретни приноси в областта на физико-математическите, хуманитарните, технологичните, техническите и социалните науки.

Авторите носят отговорност за своите материали. Текстовете се представят в редакцията на електронен носител. Оформлението на материалите трябва да бъде съобразено с приложения на последната страница модел.

Всички материали се рецензират от хабилитирани специалисти в съответната научна област. Тези материали, неполучили положителна рецензия, няма да бъдат отпечатани. Ръкописи не се връщат. Хонорари на авторите не се изплащат.

След отпечатването на серията авторският колектив може да си закупи цялата книжка.

Известията на Съюза на учените - Русе се изпращат във всички големи библиотеки в страната и в редица български и чуждестранни университети.

ГЛАВЕН РЕДАКТОР: проф. д-р Златоживка Здравкова
ОТГОВОРНИ РЕДАКТОРИ: доц. д-р Теодор Илиев и проф. Владимир Хвърчилков
ПРЕДПЕЧАТ И ТЕХНИЧЕСКА ПОДГОТОВКА: доц. д-р Ивайло Стоянов
ДИЗАЙН НА КОРИЦАТА: д-р инж.-диз. Теодор Кючуков

АДРЕС НА РЕДАКЦИЯТА: Русе 7000, ул. "Константин Иречек" 16,
Е-поща: suruse@uni-ruse.bg, **URL:** http://suruse.uni-ruse.bg/

Печат: Университетски издателски център при Русенски университет „Ангел Кънчев“

ISSN 1311-106X

СЪДЪРЖАНИЕ / CONTENTS

Страхил Соколов, Михаил Чалашканов, Красимира Иванова,Димитър Радев
Избор на критерии за оценка на бързодействието на алгоритми и програмни платформи ..... 7 - 9

Strahil Sokolov, Mihail Chalashkanov, Krassimira B. Ivanova, Dimitar Radev
Criteria selection for performance estimation of algorithms and programming platforms

Георги Кръстев, Ваня Динева
WEB базирано приложение за обучение по ролевата игра „The Quest Adventurer" ..... 10 - 18

Georgi Krastev, Vanya Dineva
WEB Based Application for Role-Playing Training „The Quest Adventurer"

Страхил Соколов, Стефан Влаев, Асен Захариев, Михаил Вукадинов
Дизайн на инфраструктура обслужваща среда за елект-ронно обучение по информационни технологии във ВУТП ..... 19 - 22

Strahil Sokolov, Stefan Vlaev, Asen Zahariev, Mihail Vukadinoff
Infrastructure Design for IT e-Learning Environment at the University of Telecommunications and Post

Лазар Хекимов
Създаване на компютърна игра ..... 23 - 31

Lazar Hekimov
Creating a computer game

Иван Иванов
Киберсигурност - заплахи за сигурността и уязвимости ..... 32 - 40

Ivan Ivanov
Cyber Security - Threats to Security and Vulnerabilities

Филип Цветанов, Мартин Пандурски
Симулационно изследване на противопожарна система за работа в облачна структура ..... 41 - 46

Filip Tsvetanov, Martin Pandurski
Simulation research of the fire system working in cloud structure

Васил Иванов
Изследване ефективността на постояннотоков двигател с независимо възбуждане ..... 47 - 52

Vasil Ivanov
Study of the Efficiency of a Separately Excited DC Motor

Васил Иванов, Христо Иванов
Изследване капацитета на автономна фотоволтаична система, захранваща IP видеокамера в непрекъсваем работен режим ..... 53 - 61

Vasil Ivanov, Hristo Ivanov
Investigation of the capacity of an autonomous photovoltaic system supplying an IP video camera in continuous operation

Росен Даскалов, Иван Белоев
Обзорен преглед на архитектурата на системите за задвижване на електрически превозни средства ..... 62 - 67

Rosen Daskalov, Ivan Beloev
An overview of the powertrain topologies for electric vehicles

ИЗБОР НА КРИТЕРИИ ЗА ОЦЕНКА НА БЪРЗОДЕЙСТВИЕТО НА АЛГОРИТМИ И ПРОГРАМНИ ПЛАТФОРМИ

Страхил Соколов[1], Михаил Чалашканов[1], Красимира Иванова[1],Димитър Радев[1]
Висше училище по телекомуникации и пощи

Criteria Selection for Performance Estimation of Algorithms and Programming Platforms

Strahil Sokolov, Mihail Chalashkanov, Krassimira B. Ivanova, Dimitar Radev
University of Telecommunications and Post

Abstract: This paper presents an overview of the proposed criteria for performance estimation of algorithms and programming platforms. The featured criteria selection is based on the elapsed time for allocating a certain set of variables and the allocated amount of memory. A recommendation is provided for using the approach on one particular platform with the default settings for the compiler or interpreter.

Keywords: performance, estimation, algorithms, programming platforms.

ВЪВЕДЕНИЕ

Бързото развитие на езиците за програмиране, води и до промяна на основни параметри на крайните софтуерни приложения. Един от тези параметри е производителността. Тази производителност се влияе както от хардуера, така и от използвания софтуер (език за програмиране), с помощта на който е реализирано приложението. Това влияние се обуславя от различния начин на използване на процесора и паметта. В различните езици резервирането и освобождаването на паметта става по различен начин. Също така различните компилатори или транслатори, предоставят възможности за промяна на количеството използвана памет, като някои от тях предоставят и параметри, влияещи на производителността.

Алгоритмите използвани за повишаване на производителността също имат различно влияние при използването им в различните езици за програмиране. Тази разлика отново идва от различното управление на паметта и използваното процесорно време за изпълнение на операторите реализиращи кеш алгоритъма.

Основната цел на изследването в тази статия е да представи критерии, по които да се осъществява сравнението между бързодействието на различни програмни езици и платформи на база еднакъв по смисъл алгоритъм, реализиран на различните програмни езици.

Останалата част от статията е организирана както следва: в глава II са описани основните съществуващи проблеми при реализацията на алгоритми и оценката за бързодействие в различните програмни среди. В част III са описани експеримента и използваната платформа. В част IV са дадени експерименталните резултати от сравнението и е направен анализ. Статията завършва със заключение и предлага изходния код на използваните алгоритми за осъществяване на сравнението.

ПРЕГЛЕД НА СЪЩЕСТВУВАЩИ ТЕХНИКИ ЗА ОПТИМИЗАЦИЯ

В зависимост от използвания език за програмиране имаме различно време на изпълнение на реализираната програма (дори и минимално в някои от случаите). От тук следва и извода, че в зависимост от използвания алгоритъм, времето на изпълнение би било различно. Ето защо за всеки език за програмиране съществуват, различни варианти с които е възможно да се повлияе на бързодействието на програмата. Тези техники са най-различни и могат да бъдат с голяма сложност, ето защо те са обособени в различни статии за съответните езици за програмиране, към които ще дадем препратки в настоящия доклад.

- Съществуват множество ресурси, свързани с оптимизацията на бързодействието в Java, например [1].
- На следващия адрес [2] са представени техники за повишаване на про-

изводителността, чрез различни алгоритми в езика за програмиране C++.

- Също така GNU организацията предоставя някои възможности за оптимизиране на приложенията [3], чрез използване на опции при компилация:
- За C# Microsoft предоставя също опции, чрез които може да се влияе на производителността [4].

Възможните начини за промяна и повишаване на бързодействието представляват интерес особенно в среди като BIGDATA където са налице изчисления върху огромни структури от данни и нужда от оптимизирани по отношение бързодействие алгоритми алгоритми и езици за програмиране. Тези възможности за подобряване на производителноста са под постоянно обсъждане дори в споделените мрежи за информация каквато е например StackOverflow [5].

ОПИСАНИЕ НА КРИТЕРИИТЕ ЗА ОЦЕНКА НА БЪРЗОДЕЙСТВИЕ

Потвърждаването на предишното твърдение изисква провеждането на екперимент с измерване на използваното време за деклариране на променлива в някои от най разпространените езици за програмиране C, C#, JAVA. За целта е създадена програма, която декларира 8000000 еднотипни променливи (това количество променливи е избрано, защото очакването е разликата да бъде много малка -микро или нано секунди, и посредством натрупването на тази малка разлика да бъде отчетени различията при съответните платформи и езици). Програмата е изпълнена на една физическа хардуерна система с параметри:

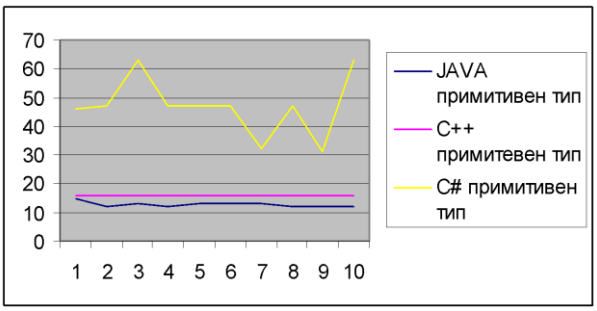
Processor: 2x
Intel® Xeon® CPU L5630 @ 2.13GHz
RAM: 6.00 GB
System type: x64 OS

компилирана на различните езици. При стартиране на програмата се използват стандартните за езиците функции позволяващи отчитането на текущото време. В края на декларирането отново се използват тези функции и разликата дава времето за деклариране на 8000000 променливи в различните езици. Променливите веднъж ще са от примитивен тип, тъй като за тях се очаква компилаторите да имат директно адресиране и тяхното създаване да бъде по-бързо в сравнение с обектните променливи, при които се конструират сложни типове данни. Изпълнява се последователно по 10 пъти всяко едно приложение във вариант с примитивен и референтен тип данни, за да може да се потвърдят резултатите и дори да има промяна във времето, на база на запълване на паметта, това ще проличи в разликата с която се променя времето на изпълнение.

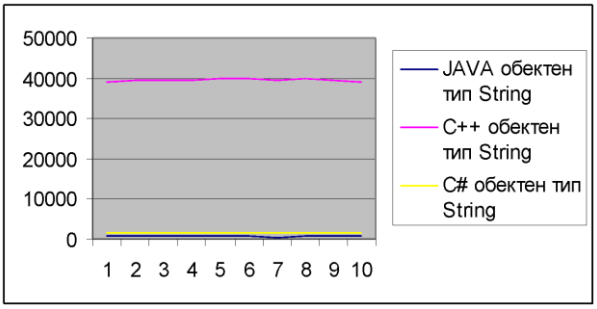
За целта на експеримента се използват алгоритми в следните езици за програмиране JAVA, C++, C#.

ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

След направените експерименти отчитаме следните резултати във времето на изпълнение на тестовите програми измерено в ms (милисекунди) показани на фиг. 1 и фиг. 2:



Фиг. 1. Графика на производителността с използване на примитивен тип данни



Фиг. 2. Графика на производителността с използване на референтен тип данни

1. Заключение

В настоящата разработка беше представен подход за определяне на бързодействието на алгоритми в различни програмни среди. Темата за производителността на

приложенията е актуална и освен бързодействието може да се разглеждат и използваните ресурси по време на изпълнение, В бъдещите разработки на екипа по темата ще бъде засегнат и този аспект от тематиката.

Благодарност

Тази разработка е финансирана по Договор №НИ-01/2016 – "Изграждане на инфраструктура за устойчиво подобряване на качеството на обучение по ОС Линукс и Основи на програмирането във ВУТП" от Висшето училище по телекомуникации и пощи, София.

Литература

1. http://www.javaperformancetuning.com/tips/rawtips.shtml
2. http://www.thegeekstuff.com/2015/01/c-cpp-code-optimization
3. https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html
4. https://msdn.microsoft.com/en-us/library/ms973839.aspx
5. https://stackoverflow.com/questions/6911863/setting-up-a-c-sharp-application-for-max-performance-build
6. Dean, Jeffrey, David Grove, and Craig Chambers. "Optimization of object-oriented programs using static class hierarchy analysis." In European Conference on Object-Oriented Programming, pp. 77-101. Springer Berlin Heidelberg, 1995.
7. Ding, Chen, and Ken Kennedy. "Improving cache performance in dynamic applications through data and computation reorganization at run time." In ACM SIGPLAN Notices, vol. 34, no. 5, pp. 229-241. ACM, 1999.
8. Kusano, Kazuhiro, Shigehisa Satoh, and Mitsuhisa Sato. "Performance evaluation of the omni openmp compiler." In International Symposium on High Performance Computing, pp. 403-414. Springer Berlin Heidelberg, 2000.
9. Stephenson, M., Amarasinghe, S., Martin, M. and O'Reilly, U.M., 2003, June. Meta optimization: improving compiler heuristics with machine learning. In ACM SIGPLAN Notices (Vol. 38, No. 5, pp. 77-90). ACM.
10. Ding, Chen, and Ken Kennedy. "Improving effective bandwidth through compiler enhancement of global cache reuse." In Parallel and Distributed Processing Symposium., Proceedings 15th International, pp. 10-pp. IEEE, 2001.
11. Knauerhase, Rob, Paul Brett, Barbara Hohlt, Tong Li, and Scott Hahn. "Using OS observations to improve performance in multicore systems." IEEE micro 28, no. 3 (2008).
12. Manferdelli, John L., Naga K. Govindaraju, and Chris Crall. "Challenges and opportunities in many-core computing." Proceedings of the IEEE 96, no. 5 (2008): 808-815.
13. J. Cavazos, G. Fursin, F. Agakov, E. Bonilla, M. O'Boyle, and O. Temam. Rapidly selecting good compiler optimizations using performance counters. In Code Generation and Optimization, 2007. CGO '07. International Symposium on, pp. 185–197, 2007

Адрес за контакти:
гл. ас. д-р Страхил Соколов
Катедра „Информационни технологии"
Висше училище по телекомуникации и пощи, София
Гр. София ул. „акад. Стефан Младенов" 1
E-mail: s.sokolov@utp.bg

WEB БАЗИРАНО ПРИЛОЖЕНИЕ ЗА ОБУЧЕНИЕ ПО РОЛЕВАТА ИГРА „THE QUEST ADVENTURER”

Георги Кръстев, Ваня Динева
Русенски университет „Ангел Кънчев”

WEB Based Application for Role-Playing Training „The Quest Adventurer”

Georgi Krastev, Vanya Dineva
„Angel Kanchev” University of Ruse

Abstract: The paper presents elements of the development of WEB application for role-playing training "The Quest Adventurer". The history of Role Play is reviewed in detail and a classification is made. The game presents: history, rules, examples and additional materials. The development is based on HTML 5, JavaScript and CSS 3.
Keywords: Computer Games, Role-playing game, RPG.

ВЪВЕДЕНИЕ

Историята на ролевите игри започва с така наречените „Военни игри”, а средата на миналия век е пикът на тяхната популярност. Историята им обаче започва още през 19-ти век, като в началото нещата тръгват от играта „Kriegspiel“ (буквално преведено – военна игра, фиг. 1) [1], която се появява в Прусия, опитваща се да симулира провеждането на военните действия. Именно „Kriegspiel“ въвежда идеята за използването на маркери, дъска за отбелязване на разположението и зарове за симулация на случайните събития. По-късно, след Френско-Пруската война в Англия се появява тамошна версия на „военната игра“, която започва да се използва масово сред военните за тактическа подготовка и предвиждане на изхода от сраженията.

Фиг. 1. „Kriegspiel”

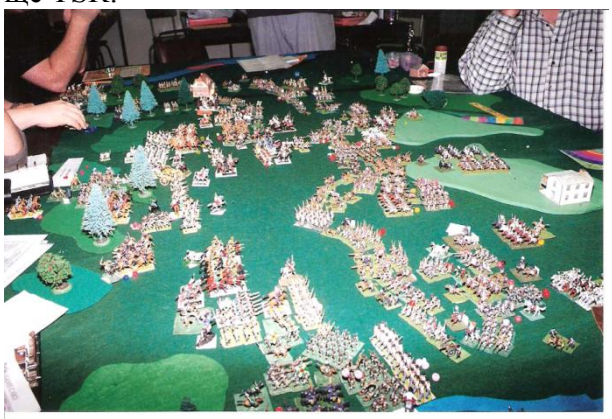
Тук се намесва един от основоположниците на научната фантастика – Хърбърт Уелс. През 1913 г. той издава книгата „Little Wars“(фиг. 2), която представлява сборник с правила за провеждане на военни игри от аматьори [2]. На практика чрез нея военните игри достигат до обикновените хора и се превръщат от средство за симулация на битките в хоби и забавление.

Фиг. 2. „Little Wars”

Книгата добива голяма популярност, но въпреки това до 1953 година военните игри си остават сравнително слабо достъпно хоби. През тази година Чарлз Робъртс издава първата търговско достъпна настолна военна игра. По-късно през 60-те и 70-те години на миналия век, настолните военни игри се радват на изключително голяма популярност, постепено превръщайки се в цяла индустрия. В същото време се появява и тенденцията на „намаляването на скалата“, при което постепенно военните игри от управление на цели армии се свеждат до управлението на отделни отряди и дори самостоятелни еди-

ници. Около военните игри се изгражда цяла субкултура.

Гайгакс, заедно с Джеф Перен и други свои приятели, създава военна игра, която достатъчно точно моделира различните аспекти на средновековното военно дело. Играта се нарича „Chainmail“ (фиг. 3) и се издава от прохождащата компания на Гайгакс, наречена Tactical Studies Riles или както ще стане много по-известна в бъдеще TSR.



Фиг. 3. „Chainmail”

В нейна по-късна версия се появяват за първи път правилата за гиганти, дракони и магически заклинания. Според някои именно „Chainmail“ е първата ролева игра, доколкото в нея присъстват всички нужни правила за адекватното описване на персонаж, но в нея липсва един от основните двигатели за бъдещите ролеви игри – система за развиване на възможностите на персонажите [4]. Правилата на „Chainmail“ обаче лежат в основата на официално считаната за първа ролева игра – „Dungeons & Dragons“(фиг. 4) [3].

Фиг. 4. „Dungeons & Dragons”

През 1968 година Гайгакс и Анерсън членуват в местно общество за средновековно военно дело, наречено „The Castles and Crusades Society“. След изиграването на няколко средновековни военни игри с доста постни, измислени от самите тях правила, един от останалите участници (Дейв Уесли), който се явява и рефер, задава на играчите персонални задачи, които да постигнат, за да разнообрази играта. И именно това се счита за първата стъпка към ролевите игри. В последствие те събират и още няколко техни приятели за да проведат по-дълга средновековна кампания, играейки по този начин.

В началото на 70-те креативността на Анерсън и фантазията на Гайгакс се срещат и двамата започват да комбинират идеите си. Тогава Анерсън използва правилата на „Chainmail“, за да проведе първата ролева игра, която в последствие прераства в кампнията за подземията на Блекмуур. В последствие Гайгакс създава собствен свят за приключенията, който се превръща в Грейхоук. През следващите години двамата играят и тестват по правилата, които в бъдеще ще създадат играта „Dungeons & Dragons“, първата ролева игра.

КЛАСИФИКАЦИЯ НА РОЛЕВИТЕ ИГРИ

Наборът от правилата на една ролева игра се нарича игрова система; самите правила са познати като игрови механики. Въпреки че има игрови системи, ползвани от много игри, например системата d20, много други игри имат своя собствена, обособена система правила.

Почти всички ролеви игри изискват участието на водещ (gamemaster; GM), който създава обстановката за игралната сесия, изпълнява ролята на повечето обитатели на света и действа като арбитър на правилата за играчите. Останалите участници създават и играят обитатели на игровия свят (game setting), и са познати под името игрови персонажи (player characters; PCs). Заедно, игровите персонажи са познати като „група“ („party“).

По време на типична игрова сесия, водещият представя сюжетна цел, която да

постигнат играчите чрез действия на техните персонажи. Често това включва взаимодействие с неигрови персонажи — другите обитатели на игровия свят, които се изиграват от водещия. Много игрови сесии съдържат разрешаване на пъзели, преговори, преследвания и битки. Целта може да бъде явна за играчите от самото начало, или да им се разкрие с течение на играта. Игровите правила определят успеха или провала на действията на даден персонаж. Много игрови системи използват статистики, измерени чрез зарове или други елементи на случайност. В повечето системи, разказвачът използва правилата, за да определи към кое число да се стреми играещият. Играчът хвърля зар, опитвайки се да получи число повече или по-малко от определеното число, в зависимост от игровата система. Повечето системи са обвързани със света на играта, който включват. Въпреки това, някои универсални ролеви системи могат да се адаптират за всички жанрове. Първата игра, която включва подобна система — GURPS, е придружена с множество книги, адаптиращи я за различни жанрове. Системата d20, известна заради ролевата игра Dungeons & Dragons, е използвана и в по-модерни светове като Spycraft и Star Wars Roleplaying Game. На практика обаче, дори универсалните игрови системи често поощряват специфичен стил или жанр, а са просто адаптивни към останалите. Например, въпреки че системата d20 има книги с материали за модерни и футуристични светове, повечето издаден материал за d20 си остава в бойно-ориентираната фентъзи ниша на Dungeons & Dragons.

ТИПОВЕ РОЛЕВИ ИГРИ

Освен традиционния тип настолни ролеви игри, съществуват и някои други разновидности.

- На живо: ЛАРП е форма на ролева игра, свързана с настолните ролеви игри, но подобна повече на импровизиран театър. На български са наричани и „ролеви игри на живо“, тъй като за разлика от настолните такива, при ЛАРП играта се разиграва не „на думи“, във въображението на играчите, а на живо - от тях се изисква да отиграват своите персонажи.
- В електронната медия: Съществува и електронен вариант на ролеви игри, който използва похватите на настолните игри и се различава от компютърните игри преди всички по факта, че се базира на писане на текст от играчите, а не на визуални взаимодействия. Възможни са различни варианти като среда за играта, например имейл, използване на IRC канал, но най-популярни са форумните игри. Различните форуми в мрежата предлагат ролева игра по популярни книги, сериали и истории, където се използват готови персонажи. В някои обаче, светът е фантазен и измислен и всеки има правото да създаде герой по свой вкус.
- Фрийформ: Фрийформ ролевите игри се играят с минимални, формални или никакви правила и са повече фокусирани върху историята на участващите персонажи. Те биват играни както чрез интернет, така и на живо.

РАЗРАБОТКА НА ПРИЛОЖЕНИЕТО

Разработеното WEB базирано приложение се класифицира като асинхронен, мултимедиен сайт за електронно обучение.

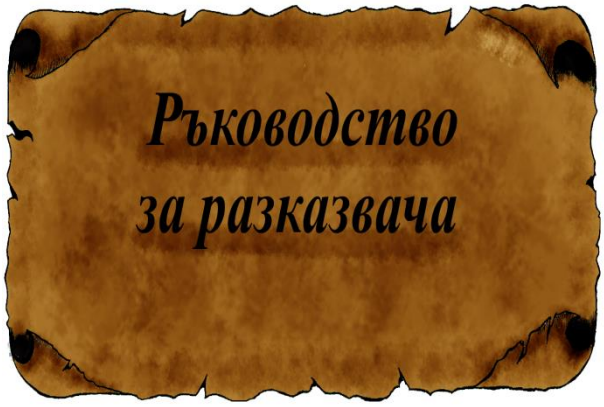
Сайтът съдържа пет раздела:

- „Начало” – предоставя уводна информация, за новодошлите потребители.
- „История” – графична и текстова информация за историята на играта върху която сайтът извършва електронно обучение.
- „Правила” – обобщена графична и текстова информация върху правилата за провеждане на игрова сесия, както и връзки към допълнителни материали, като „персонажен лист”.
- „Ръководство” – пълно ръководство за провеждане на игрова сесия.
- „Примери” – примери и полезни съвети за начинаещите играчи, които желаят да прочетат реални игрови ситуации и да придобият представа как точно „изглежда” една реална игрова сесия.

Съдържанието на сайта е насочено към разработената игра „The Quest Adven-

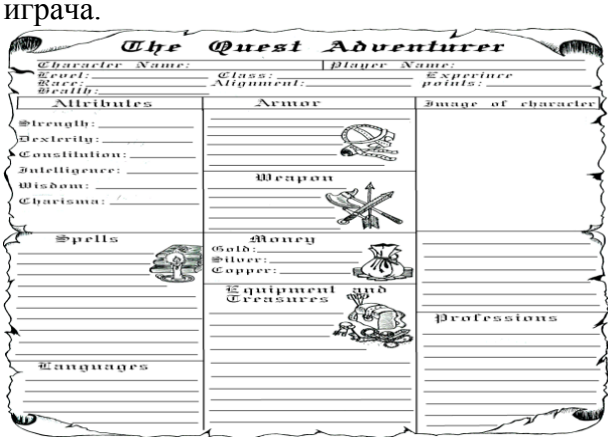
turer", за която са представени история, правила, примери и допълнителни материали. Разработката е реализирана на базата на HTML 5, JavaScript и CSS 3.

В изложението подробно е разгледано ръководството за разказвача.



А. Основни понятия в ролевата игра. Сборът от правилата на една ролева игра се нарича игрова система, самите правила са познати като игрови механики. Въпреки че има игрови системи, ползвани от много игри, например системата d10(десетстенен зар), много други игри имат своя собствена, обособена система правила. Почти всички ролеви игри изискват участието на водещ (gamemaster; GM), който създава обстановката за игралната сесия, изпълнява ролята на повечето обитатели на света и действа като арбитър на правилата за играчите. Останалите участници създават и играят обитатели на игровия свят (game setting), и са познати под името игрови персонажи (player characters; PCs). Заедно, игровите персонажи са познати като „група" („party").По време на типична игрова сесия, водещият представя сюжетна цел, която да постигнат играчите чрез действия на техните персонажи. Често това включва взаимодействие с неигрови персонажи — другите обитатели на игровия свят, които се изиграват от водещия. Много игрови сесии съдържат разрешаване на пъзели, преговори, преследвания и битки. Целта може да бъде явна за играчите от самото начало, или да им се разкрие с течение на играта. Игровите правила определят успеха или провала на действията на даден персонаж. Много игрови системи използват статистики, измерени чрез заровe или други елементи на случайност. В повечето системи, разказвачът използва правилата, за да определи към кое число да се стреми играещият. Играчът хвърля зар, опитвайки се да получи число повече или по-малко от определеното число, в зависимост от игровата система.

Обикновено, персонажите в ролевите игри са представени чрез съвкупност от показатели, отразяващи възможностите му в различни сфери от живота. Много игрови системи откроляват две ключови статистики: характеристики (attributes) и способности (abilities/spells).Преди да започне играта, играчите създават концепция за ролята, която искат да играят. След това те прилагат правилата на игровата система за създаване на персонаж, за да опишат героите си чрез игровите механики. Показателите на персонажа се записват на формуляр, направен специално за въпроса, който се нарича персонажен лист (character sheet). Системата, базирана на шаблони има предимство - създаването на персонаж е лесно и бързо. Това означава, че разказвачът ще прекара по-малко време, одобрявайки баланса на всеки герой. Жертвите са в гъвкавостта и концепцията. Шаблоните по същество представляват предварително изградени персонажи, които са балансирани един спрямо друг. Базираните на класове системи дават малко повече свобода, но все пак изискват от играча да избере от фиксиран брой роли за персонажа си. Силите на персонажа са дефинирани от класа, но специфичните показатели са избрани от играча.



Фиг. 5. Персонажен лист (character sheet)

Общи условия за провеждане на игра. Основните правила на играта са разделени на няколко етапа:

- Основни характеристики на героите и техните класове и особености;
- Създаване на характеристика за герои;
- Правила на играта и необходими материали;
- Създаване на ролевата игра като кампания.

При създаването на една ролева игра като процес е нужно да има следните особености:

- История, която да бъде създадена от Разказвача и да бъде разделена на три основи части: Начало; Среда; Край.
- Да създаде задачи за героите, които участват в кампанията;
- Да използва правилата за провеждане на игра;
- Да използва изобретателност: да добави малки елементи към играта, които да заинтересуват повечето играчите.

Съвети. Разказвачът има право да променя историята на играта, както и правилата. За създаване на история за играта, разказвачът може да прочете книги ориентирани в дадената сфера, която той ще избере. След като има познанията за различните жанрове книги , истории и събития в тях, той може да вземе малки идеи или откъси от тези книги и да създаде нова история, или сам да си измисли история базирана на знание върху дадената сфера.

Ако това затруднява разказвачът, той може да използва само една книга и по нея да създаде история с променени събития.

Също може да се използват и филми (самата история на филма не е толкова добре описана като една книга , но има интересни откъси в самият филм, които ако се използват и се опишат подробно, ще са много добра находка в дадената история и интересен елемент към дадено действие в една сесия)

Както повечето ролеви игри самото създаване на история опира до факта,че няма готова литература към ролевата игра, която разказвачът може да използва. Това са кратки, но и полезни съвети как да си създадеш история.

Б. Основни правила за изграждане на персонажен лист. За изграждането на персонажният лист е нужно да се знаят основните правила и характеристики на героя.

Персонажният лист на ролевата игра „The Quest Adventurer" има следните елементи: В най-горната част на персонажният лист се намират полетата за Име на Героят (Character Name) и Име на Играещият (Player Name). Тези полета са предназначени като информация за разказвачът.

Под тези полета се намират описателни полета за героя: Ниво (Level); Раса (Race) – расови черти; Клас (Class) – класове и основни правила, *таблица 2*; Характер (Alignment) – характер на героя, *таблица 1*; Опит (Expirience points) – таблица за качване на нива и опит, *таблица 6*. Следващите полета са за: Атрибути (Attributes) – това са основните и второстепенни умения на героя, които са: Сила (Strength); Ловкост (Dexterity); Здравина (Constitution); Интелигентност (Intelligence); Мъдрост (Wisdom); Чар (Charisma).

Способности и магии (Abilities and spells) – това са основните умения и магии на дадения герой;

Езици (Languages) – за владеене на езици , *таблица 5*;

Броня (Armor) – за каква броня използва героят;

Оръжие (Weapon) – какво оръжие използва;

Пари (Money) – налични пари в героя;

Екипировка и съкровища (Equipment and treasures) – свободно пространство (торба, раница), в което се описва инвентара на героя и съкровищата;

Изображение на героя (Image of character) – свободно поле, в което може играещият да си нарисува своят герой;

Професии (Professions) – какви професии умее героят, *таблица 3.*

Таблица 1. Характер на героя, нрав (Alignment)

Видове характери		**Черти на характера**
1	Добър	Независимост (свобода) и добротта

2	Справедливо Добър	Цивилизация и заповеди
3	Зъл	Тирания и омраза
4	Хаотично Зъл	*Ентропия и разрушение
5	Неотговарящ характер (нямащ)	Не взима ничия страна (нито е добър, нито е зъл)

* ентропия - мярка за безпорядък (хаос)

Таблица 2. Основни и второстепенни умения (Primary and secondary)

Основни и второстененни умения на класовете			
	Войн	**Стрелец**	**Магьосник**
1	Сила: основно (Strength: primary)	Ловкост: основно (Dexterity: primary)	Интелигентност: основно (Intelligence: primary)
2	Ловкост: второстепенно (Dexterity: secondary)	Сила: второстепенно (Strength: secondary)	Сила: второстепенно (Strength: secondary)
3	Интелигентност: второстепенно (Intelligence: secondary)	Интелигентност: второстепенно (Intelligence: secondary)	Ловкост: второстепенно (Dexterity: secondary)

Primary – използва се за атака (основно умение на дадения клас герой);
Атака: primary + weapon damage + level – defence;
defence (защита):
1) при различни класове:
Войн = strength + armor;
Стрелец = dexterity + armor;
Магьосник = intelligence + armor
2) при еднакви класове: armor
HP = constitution + level * 2
Charisma = за убеждаване, влюбване
(твоята Charisma + 1d4) – (вражеският чар +1d4)
Магии - за успешно заклинание:
твоят intelligence + d12 + твоят level – вражеският dexterity + d12 + вражески level;
Използване на професии:
1d20: при хвърляне на зар, за да се брои за успех от 15 на горе, а при неуспех от 14 на долу.

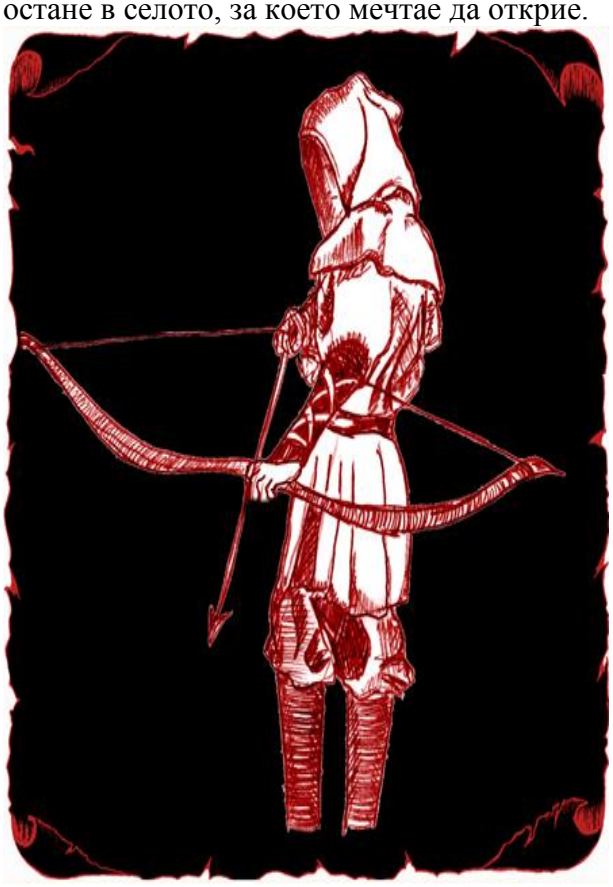
Физически характеристики на човек:

- + 2 точки на едно умение по избор;
- Движение: 6 квадратчета;
- Видимост: нормална;
- Езици: общ + 1 език по избор.

Кратък справочник на персонажите.

Войн: Наемник, който продава себе си на баронети, херцози и принцове. Бронята му е практична, без никакви декорации, и мечът му е предназначен само за битка, а не като украшение. Той се моли на Бога на Войната, но не и извън границите на вяра към този бог, защото знае че смъртта е неизбежна и се надява да може да бъде отложено колкото се може повече.

Стрелец: Очите му винаги гледат към хоризонта. Страстта го движи към разпознаване и към нови места. Мечтае да създаде ново селище, място на което хората могат да живеят в мир и свобода. Моли се на Бога на Мира и на Бога на Мечтите, за да осъществи своите мечти. Той е създание на дивото, но мечтата му е да го опитоми. Една част от него се чуди дали ще може да остане в селото, за което мечтае да открие.



Магьосник: Единственото за което жадува е могъщество. Той пътешества за да намери древни знания и древни артефакти, които ще му помогнат да си увеличи мощта на магията му. Той се моли на Бога на Мощта, защото търси знание, което е ключ към могъщество, но се чуди дали да се моли и на Бога на Тайните, защото не знае какво може да му предложи в замяна на вярност.

Таблица 3. Видове професии

Видове Професии		
Ковач	**Шивач**	**Писател и четец**
Притежава великолепни знания в областта на ковачеството и чрез тези знания може да обработва метал, да създава инструменти, оръжия, брони. Когато работи върху нещо не обича да бъде обезпокояван, поради факта че работи с нагорещен метал.	Притежава великолепни умения в областта на шиенето и чрез тези знания може да направи различни видове облекла, камуфлажи и много други неща свързани с обработка на платове и кожи. Обожават да шият през повечето време, защото обичали да носят шапки от животинска кожа, щавена с живак.	Притежават великолепни умения в областта на писането и разчитането на книги, които могат да са на древен или чужд език , но им отнема повече време за превеждане на книгата. Обожават да четат най-различни жанрове книги и понякога се задълбочават толкова много, че изключват външният свят (не го отразяват). Също притежават и допълнителни знания в областта на Историята и Науката.
Бонус Умение :	**Бонус Умение :**	**Бонус Умение :**
Strength + 3	Dexterity + 3	Intelligence + 2; Wisdom + 1

Таблица 4. Скала за получаване на точки след нападение

Брой точки (получаване)	Ниво на врага	Видове врагове/чудовища	Кръв на врага/чудовище	Здравина на врага/чудовище	Кърви	Рефлекс	Клас на бронята
25	1	звяр	12	3	6	4	4
36	2	звяр	16	4	8	5	5
39	3	звяр	19	5	9	5	5
44	4	звяр	23	6	12	6	6
50ч	5	звяр	26	8	13	7	7
63	6	звяр	30	9	15	8	8
75	7	звяр	33	10	17	9	9
88	8	звяр	36	11	18	10	10
100	9	звяр	40	12	20	11	11
125	10	звяр	44/46	13	22/23	12	12

Таблица 5. Видове езици

Видове езици		**За разговаряне с различни раси**
1	Общ	Хора, полуръстове,тифлинги
2	Драконов	Дракони, хора-дракони, хора-гущери
3	Джуджешки	Джуджета, огнени джуджета, каменни джуджета
4	Елфически	Елфи, елфически създания, неживи елфи, тъмни елфи
5	Оркски	Орки, огрета, пуло-орки
6	Примордиал (език на елементите)	Елементи, древни елементи, създания с тяло състоящо се от елементи (природни : вода, въздух, огън, лед, земя)

Таблица 6. За качване на нива

Ниво	Опит	Всички класове	Клас Войн	Клас Стрелец	Клас Магиьосник
1	0	primary + 5 secondary + 2	Con= 10 Wis= 2 Char= 2	Con= 7 Wis= 2 Char = 4	Int= 4 Con= 4 Char= 2
2	150	Constitution + 1	-	-	-
3	500	Primary + 2 Secondary + 1	Con + 2	Char + 2	Int + 2
4	1000	Constitution + 1	-	-	-
5	1750	Primary + 2 Secondary + 1	Con + 2	Char + 2	Int + 2
6	3250	Constitution +1 Primary + 3 Secondary + 2	Char -1	Wis – 1	Con – 1
7	6300	Constitution + 1	-	-	-
*8	9800	Primary + 2 Secondary + 1	Con + 2	Char + 2	Int + 2
9	13850	Constitution + 1	-	-	-
*10	18850	Primary + 3 Secondary + 3	Con + 3	Char + 3	Int + 3

* Бонуси при развитие на героя

Ниво *8	Героят може да получи злато, като макцималният брой златни монети е на стойност 570 , при условие, че успее да метне 1d20 (пример: 1d20 – мята – получава число 20 ,което е максималната стойност на зара – получава 570 злато)
Ниво *10	Героят може да се научи да язди кон, срещу заплащане .

Терминологичен речник за табл. 6:

- Primary: това е основното умение на даденият клас, което получава бонус точки на повечето нива;
- Secondary: това е второстепенното умение на даденият клас, което получава бонус точки на повечето нива;
- Constitution: съкращение „Con”, здравина – определя здравината на героя;

- Wisdom: съкращение “Wis”, Мъдрост – като умение на дадения клас се използва;
- Charisma: съкращение “Char”, Чар – като умение на дадения клас се използва;
- Intelligence: съкращение “Int”, Интелигентност – като умение на дадения клас се използва.

Видове зарове: D4 – зар с четири страни; D6 – зар с шест страни; D8 – зар с осем страни; D10 – зар с десет страни; D12 – зар с дванадесет страни; D20 – зар с двадесет страни.

ЗАКЛЮЧЕНИЕ

В настоящата статия са представени елементи от разработката на WEB приложение за обучение по ролевата игра „The Quest Adventurer”. Направен е обзор на съществуващите решения, в който подробно е разгледана историята на Ролевите игри и е направена класификация.

Разработката е реализирана на базата на HTML 5, JavaScript и CSS 3.

Предвижда се в бъдеще функционалността на WEB приложението да бъде подобрено с включване на допълнителни раздели като:

- готови сценарии за провеждане на игра;
- класове и герои;
- външни връзки съдържащи полезна информация за играча;
- софтуерни реализации на играта.

С разработката се създават условия за по-голяма ефективност и качество при трансферирането на научното знание в това направление.

Литература

[1] http://r-s-g.org/kriegspiel/rules.php
[2] http://www.bbc.com/news/magazine-22777029
[3] http://dnd.wizards.com/
[4] Gygax, Gary; Perren, Jeff (1971). Chainmail. Evansville IN: Guidon Games. p. 29.

Адрес за контакти:
проф. дн Георги Кръстев
Катедра „Компютърни системи и технологии“
Русенски университет “Ангел Кънчев”
E-mail: gkrastev@ecs.uni-ruse.bg

ДИЗАЙН НА ИНФРАСТРУКТУРА ОБСЛУЖВАЩА СРЕДА ЗА ЕЛЕКТРОННО ОБУЧЕНИЕ ПО ИНФОРМАЦИОННИ ТЕХНОЛОГИИ ВЪВ ВУТП

Страхил Соколов, Стефан Влаев, Асен Захариев, Михаил Вукадинов
Висше училище по телекомуникации и пощи

Infrastructure Design for IT e-Learning Environment at the University of Telecommunications and Post

Strahil Sokolov, Stefan Vlaev, Asen Zahariev, Mihail Vukadinoff
University of Telecommunications and Post - Sofia

Abstract: This paper presents the technical architecture of the proposed new technology e-learning environment that will be of help for the students in IT at the University of Telecommunications and post in Sofia, Bulgaria. Hardware requirements are proposed as well as software components which consist of portals for teacher and students, master database, storage and backup mechanisms.
Keywords: e-Learning, students, IT.

ВЪВЕДЕНИЕ

Напредъкът в информационните и комуникационните технологии позволява до огромна степен да работим и учим от разстояние. Широко е възприето, че ученето може да става извън класната стая. Съвременни автори [2], [3], [5] се обединяват около идеята, че ключов фактор в обучението е да бъде поддържана непрекъснатата пряка връзка между студентите и техните обучители.

Електронното обучение има значително въздействие върху висшето образование. Това е непрекъснато разширяващ се свят на знанието без граници и поле на безкрайни възможности. Това, което се опитват да постигнат най-съвременните системи за електронно обучение, не е само връзка с преподавателя - те също така могат да предскажат и насочат специфичните нужди на самите обучаващи се. Ето защо е важно такава среда да бъде непрекъснато достъпна [1] и да може бързо да доставя от разстояние всички възможни учебни ресурси за студентите [8]: като се започне от учебните материали и основните предпоставки за осъществяване на занятията, когато студентът работи отдалечено.

Тази статия продължава както следва: в следващия раздел е описана предложената архитектура и насочената към потребителя част на средата за обучение (Frontend). В раздел 3 са описани механизмите на компонентите за балансиране на натоварването и централната база данни. Раздел 4 съдържа описание на избраната виртуализация и автоматизация на средата.

ОПИСАНИЕ НА ПРЕДЛОЖЕНАТА АРХИТЕКТУРА

Целта на тази част е да покрие дизайн на инфраструктура, основана на съвременни технологии и инструменти. В следващата си част, статията е разделена на два основни потока - Front-end и Back-end.

Насочената към потребителите част (Front-end) на системата за обучение ще се състои два портала - учител и студент. Всеки портал ще обслужва на предварително определена група хора.

Портал на преподавателите (ПП) - Основната цел на този портал е да осигури лесна употреба и да управлява функционалността за преподавателите и техните асистенти.

Тук се осъществява прегледа и се добавят оценки, резултати от задачи, както и да се заключат/отключат курсове. Една от основните характеристики е, че обучителите могат да извършват одит на извършените от учениците стъпки, докато са изпълнявали задачата. Например кои команди са въведени, за да инсталирате и конфигурирате MySQL и Apache сървъра.

Всичко това ще бъде достъпно в централна база данни (или MasterDB), съдър-

жаща необходимата информация. Повече подробности ще бъдат предоставени в частта от статията, посветена на MasterDB.

Студентски портал (СП) - Основната цел на този портал е да осигури достъп на студентите, така че да могат да преглеждат оценки, резултати от задачи и обяд. Например, предварително конфигурирана виртуална машина с всички необходими инструменти, за да може студентът да изпълни задачата, без да губи ценно време за допълнителни конфигурации.

След стартирането на упражнението студентът ще получи подробности за достъпа до машината - IP, потребителско име и парола. Допълнително към това може да се осигури кратко ръководство.

Фиг. 1 Компоненти на предложената инфраструктура във ВУТП

БАЛАНСИРАНЕ НА НАТОВАРВАНЕТО И СЪХРАНЕНИЕ НА ДАННИТЕ

Предложено е хардуерно изпълнение във вариант с два сървъра, например HP ProLiant® G6, върху които ще бъде изградена останалата част от инфраструктурата.

За балансиране на натоварването (LB) и Failover (FO) ще се използва High Availability Proxy (HAProxy).

HAProxy[5] е софтуер с отворен код за TCP / HTTP натоварване за балансиране на натоварването и за проксиране на базата на Linux / Unix операционни системи. Той обикновено се използва за подобряване на производителността и надеждността на услугата чрез разпределяне на работното натоварване на няколко сървъра (например уеб, приложения и бази данни).

HAProxy ще бъде инсталиран на Linux RedHat/Centos виртуална машина с хардуерни изисквания, които ще варират в зависимост от трафика, който преминава през HAProxy.

Дадени са примери за ниско ниво, средно и високо ниво на натоварване.

За ниско ниво на натоварване се предполага, че трафикът по TCP и/или HTTP ще бъде до 1000 потребителски връзки и с много нисък трафик на файлове за компресиране на SSL и/или gzip.

Това натоварване може да се управлява от виртуална машина хардуерен сървър с поне 1 CPU и 1GB RAM.

За средно ниво на натоварване се счита, че трафикът по TCP и/или HTTP (включително HTTP обработка) ще бъде до 4000 връзки и с нисък трафик на файлове за компресиране на SSL и/или gzip.

Това натоварване би могло да бъде обработено от виртуална машина или хардуерен сървър с поне 2 CPU и 1GB RAM.

За високото натоварване ще предположим, че трафикът по TCP и/или HTTP (включително HTTP обработка) ще бъде до 20 000 връзки и с 10% запасен трафик на SSL и/или gzip файлове за компресиране.

Това натоварване може да бъде обработвано от виртуална машина или хардуерен сървър с поне 2 CPU и 4GB RAM.

Другата характеристика, която може да подобри практическата работа на потребителите, е предоставената от HAProxy – проверки на състоянието. С тази функция HAProxy може да определи дали обслужващия сървър (или уеб сървърът) е свързване на конфигурирания порт. HAProxy ще изпрати TCP заявка до уеб сървъра и ако няма отговор трафикът ще се пренасочи към активния (уеб) сървър.

ЦЕНТРАЛНА БАЗА ДАННИ

Предложено е порталите да бъдат обслужвани от уеб сървър на Apache. Основната (MasterDB) база данни ще разчита на MySQL[6] или MariaDB[7].

MasterDB ще бъде инсталиран на две самостоятелни виртуални машини с минимум 2 CPU и 4 GB RAM.

MasterDB ще съдържа информация за идентификационния номер на студента, име и фамилия, текущия семестър, курсовете, в които студентът се занимава и точките от неговите упражнения и степен за курса.

Потребителите от ПП могат да добавят нови студенти (име, фамилия и фамилия), да задават курсове, да дават оценки въз основа на резултатите от курса и като цяло дават оценка за студент.

MasterDB ще бъде инсталиран в клъстер с конфигурация master-slave, който ще осигури резервно копие на основната DB в случай на бедствие/повреда на данните.

В последващото развитие на архитектурата конфигурацията на клъстерите може да бъде оптимизирана с клъстера Galera[7], който ще осигури активна активна конфигурация, която повишава производителността и премахва конфликтите. Друго подобрение е да включи в конфигурирацията сървър с копие на DB за четене и друг – с копие DB за записване. Сравнението на вариантите и целесъобразността на предложенията предстои да бъдат разгледани в бъдещи разработки.

ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

Съществуват много инструменти и методи, използвани за провизиране, внедряване и автоматизация. За нуждите на предложената инфраструктура е избрана виртуализация от типа виртуална машина в ядрото - Kernel Virtual Machine (или KVM [9]), Docker [10] и Puppet [11].

Виртуалната машина, базирана на ядрото, е хипервайзор, който осигурява пълна виртуализация на Intel и AMD архитектурата. Веднъж инсталиран, KVM може да създаде отделни виртуални машини, всяка от които запуска своя собствена операционна система, собствена конфигурация за съхранение, мрежа и сигурност. Голямата полза от използването на KVM е, че може да поддържа разнообразие от гост-операционни системи - Linux дистрибуции, Microsoft® Windows®, BSD и Solaris.

Docker е контейнер, осигуряващ софтуер, характеризиращ се с олекотено изпълнение, базиран на отворени стандарти и повишена сигурност. Под олекотено изпълнение следва да се разбира, че контейнерите споделят една и съща операционна система и могат да стартират много бързо. Тази технология с отворен код може да бъде стартирана както на Linux, така и на Windows OS и при това е безплатна. Контейнерите са изолирани едни от други, въпреки че използват една и съща операционна система.

Puppet е инструмент за управление на конфигурацията на сървър-клиент както за Unix/Linux, така и за операционната система Windows. Използвайки присъщия Puppet декларативен език или Ruby, потребителят може да определя действията по настройка, системния ресурс и тяхното желано състояние.

В предложената инфраструктура KVM и Docker ще се използват за осигуряването на отделните OS, с които ще работят студентите - Linux и Windows. С KVM ще бъдат осигурявани виртуални машини с Windows за курсове или задачи, където е необходимо. От семейството на операционната система Windows KVM поддържа от Server 2003 до Server 2012. С помощта на опцията KVM шаблон (template) може да се създаде предварително дефинирани образи, които да обслужват предназначения. Метаданните на шаблона се съхраняват в XML файл, от който се създават няколко предварително дефинирани шаблони с минимален обем данни.

Docker ще бъде основният източник за Linux операционни системи. Предварително дефинираните контейнери ще съдържат необходимото приложение/пакети. Конфигурацията на новите се контейнери ще бъде осигурена чрез редактиране на техния Docker файл.

С помощта на Puppet ще бъде манипулирана конфигурацията на шаблона на KVM и контейнерния файл на Docker. Също така ще се използва за внедряване на допълнителен софтуер при вече стартираните виртуални машини. В Puppet модул ще се съдържа информация, имена на ресурси и действия, които Puppet трябва да

изпълни преди и след създаването на виртуалната машина. Puppet също ще бъдат използвани като инструмент за внедряване на нови версии на порталите (ПП), (СП). Свързването на Puppet с GitHub би могло да осигури лесна употреба и управление на хранилища за контейнери и предстои да бъде разгледано в бъдещи разработки.

ЗАКЛЮЧЕНИЕ

В настоящата разработка беше представена техническата архитектура за среда за електронно обучение, базирана на уеб сървъри и HAproxy за управление на високата надеждност на предоставяните услуги и балансирането на натоварването. Дискутирана беше характеристиката на система да бъде автоматизирана до голяма степен, особено в областта на разгръщането и управлението на конфигурацията на виртуалните машини. Беше представена идеята да се използва Puppet и контейнери за виртуализация в ядрото на Linux, за да се подсигурят необходимите ресурси от операционни системи. Бъдещата работа на екипа ще бъде фокусирана върху реализацията, подобряването на предлаганата инфраструктура и популяризирането на платформата за доставка на електронно обучение.

Благодарност

Тази разработка е финансирана по Договор №НИ-01/2016 – "Изграждане на инфраструктура за устойчиво подобряване на качеството на обучение по ОС Линукс и Основи на програмирането във ВУТП" от Висшето училище по телекомуникации и пощи, София.

Литература

1. Kumar, S., Gankotiya, A.K. and Dutta, K., 2011, April. A comparative study of moodle with other e-learning systems. In Electronics Computer Technology (ICECT), 2011 3rd International Conference on (Vol. 5, pp. 414-418). IEEE..
2. Kalyuga, S., 2007. Enhancing instructional efficiency of interactive e-learning environments: A cognitive load perspective. Educational Psychology Review, 19(3), pp.387-399..
3. Masud, M.A.H. and Huang, X., 2012. An e-learning system architecture based on cloud computing. system, 10(11), pp.255-259.
4. Clark, R.C. and Mayer, R.E., 2016. E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. John Wiley & Sons.
5. W. Tarreau. (2012, April). Haproxy: the reliable, high performance tcp/http load balancer. [Online]. Available: http://haproxy.1wt.eu/.
6. MySQL, A. B. "MySQL: the world's most popular open source database." http://www. mysql. com/ (2005).
7. Bartholomew, D., 2014. MariaDB Cookbook. Packt Publishing Ltd.
8. Garrison, D.R., 2011. E-learning in the 21st century: A framework for research and practice. Taylor & Francis..
9. CUI, Z.Y. and ZHAO, H.Q., 2011. Research and Application of Virtualization Based on KVM [J]. Computer Technology and Development, 6, p.030.
10. Merkel, D., 2014. Docker: lightweight linux containers for consistent development and deployment. Linux Journal, 2014(239), p.2.
11. Turnbull, James, and Jeffrey McCune. Pro Puppet. Apress, 2011.

Адрес за контакти:
гл. ас. д-р Страхил Соколов
Катедра „Информационни технологии"
Висше училище по телекомуникации и пощи, София
Гр. София ул. „акад. Стефан Младенов" 1
E-mail: s.sokolov@utp.bg

СЪЗДАВАНЕ НА КОМПЮТЪРНА ИГРА

Лазар Хекимов
Русенски университет „Ангел Кънчев”

Creating a Computer Game

Lazar Hekimov
„Angel Kanchev” University of Ruse

Abstract: Creating a computer game. The game is a software application, which lets the user to control the course of action. The game by it's own offers the participation of at least one player, who can interact with the environment and his surroundings. The game has a goal which the player thrives to accomplish by being sufficiently inventive, creative and has a fast mind. There are games in a variety of genres. The purpose for development of the game „A tale of a Knight“, is to apply modern methods and technics for painting graphics and programing. Made with the help of: Unity, Freephototool and Audacity, with uncommercial purpose.
Keywords: Computer game, Unity3D, Freephototool, Audacity.

ВЪВЕДЕНИЕ

Компютърната игра е софтуерно приложение, което позволява на потребителя да контролира хода на действието. Играта, сама по себе си, предполага участието на поне един играч, който може да влияе на средата, в която е поставен. Тя има цел, към която геймърът се стреми и която може да постигне стига да бъде достатъчно находчив, ловък или смел [3, 17]. Срещат се игри в най-разнообразни жанрове [9, 13, 21, 23]: екшън от първо лице, екшън от трето лице, ролеви игри, приключенски игри, симулационни игри, спортни игри, стратегически игри (походови стратегии и реално-времеви стратегии), platform игри, casual игри и др.

Човеко-машинната комуникация е интердисциплинарна задача, при която към практиките от компютърните науки трябва да се приложат например теории и принципи от психологията [1], социологията и антропологията, както и подходи в сферата на дизайна и индустриалния дизайн за създаване на интерактивни продукти.

Човеко-машинният интерфейс (ЧМИ) дефинира начина, по който хората и компютрите си взаимодействат. Визуално-базираната човеко-машинна комуникация буди най-голям интерес сред изследователската общност в сферата на ЧМИ. Типичен представител е разпознаването на жестове [10, 11, 12]. Те се използват за пряко взаимодействие човек–машина в релация на задаване на команда и последващо действие/реакция.

Сензорно-базираната човеко-машинна комуникация има широка област на приложение. Характеризира се с използването на поне един физически сензор за осъществяването на комуникация, като сензорите могат да бъдат както много примитивни, така и много сложни [8].

Интерфейсът човек-компютър чрез анализ на активността на мозъка (Brain-Computer Interaction Interface - BCII) е създаден през последните години [5, 6, 7]. Става дума за уникалното взаимодействие между човек и компютър, което се осъществява след регистриране и обработка на сигнали с произход от човешкия мозък (най-често това са електроенцефалографските сигнали - ЕЕГ). Основното им предназначение е за използване в игрите. Чрез тях може да се „разбират“ нервните състояния на човека като страх, напрегнатост, спокойствие и по този начин игрите стават много по-реални. Вълнуващите и страшни моменти в игри, като F.E.A.R. Penumbra ще затруднят много повече играчът, защото страхът реално ще му въздейства. Курсорът се управлява с движение на очите, а въздействието върху обектите става с помощта на концентрация.

Целта на настоящата разработка е създаване на играта „A tale of a Knight“,

като се прилагат съвременни техники за рисуване и програмни технологии.

ПРОГРАМНО ОСИГУРЯВАНЕ

Unity е среда (междуплатформен игрови енджин) за създаване на компютърни игри и предлага възможност за разработване на такива в режими 2D и 3D [4, 16, 18].

В началото през 2005 г. на световната конференция за програмисти на Епъл е обявен само за OS X. Оттогава е разширен за повече от петнадесет платформи (фиг.1).

Енджинът представлява съвкупност от компоненти, чрез които могат да се управляват различните обекти в дадена сцена, като към тях се включи и програмната логика, сцената се превръща в ниво от играта. При това поддържаните езици за програмиране основно са C#, JavaScript и UnityScript.

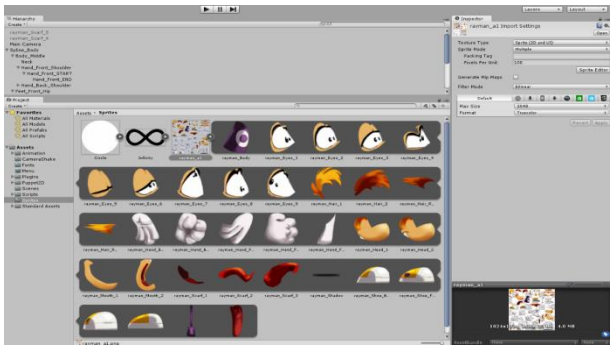
Unity предлага две версии – безплатна (за лично ползване и без право на разпространение) и платена версия Pro, която е за комерсиално ползване. Всичко, направено с Unity би работило по същия начин, както ако е направено с Unity Pro. Причината за това е потребителите да имат възможност за преминаване от безплатната към Pro версията, без да променят нищо.

Фиг. 1. Поддържани платформи

Unity 2D е дял на Unity, предназначен за разработката и обработката на двуизмерни обекти, като има различни инструменти, пригодени за по-лесното им управление и имплементиране във виртуалния свят.

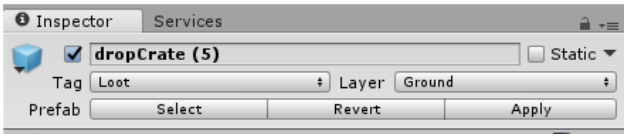
Спрайтовете са основните графични елементи, от които е изградена една игра, като могат да бъдат както двуизмерни, така и триизмерни. Те могат да изпълняват ролята не само на статични графични елементи, но и обекти, които да бъдат програмирани чрез вграждането на скриптове в тях, както и задаване на определени стойности на изграждащите ги елементи.

Фиг. 2. Unity 2D среда



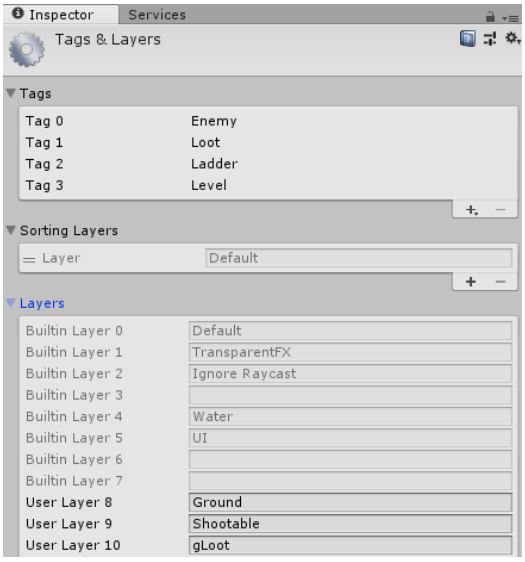
Фиг. 3. Примерни спрайтове

Таговете и слоевете се използват в голяма степен при програмирането на игрите, като чрез тях енджинът разбира кои обекти са земя, кои са противници и т.н.. Така чрез скриптовете може да се определя какво да се случва с играча при допир с обект с таг-земя или при сблъсък на projectile с обект със слой-Shootable. Също така слоевете се използват и за сортиране на обектите, от които е изградена играта или нейния фон.

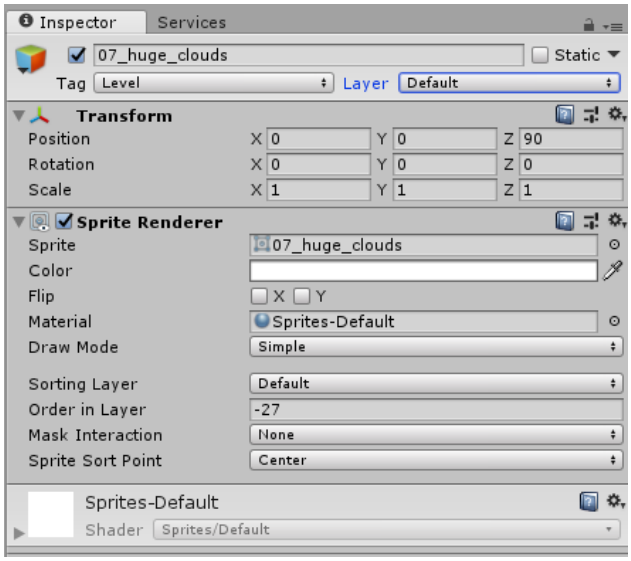


Фиг. 4. Тагове и слоеве

Слоевете имат подредба, в която да се генерират, като чрез нея се определя кой обект пред кой е и кой зад кого е. Така, ако имаме два обекта, които са на едно и също място на координатата „Z“ и искаме единия да е пред другия, трябва да му зададем стойност на подредбата на слоеве да е по-голяма от тази на задния.



Фиг. 5. Таг и слой на обект



Фиг. 6. Ред на слоевете

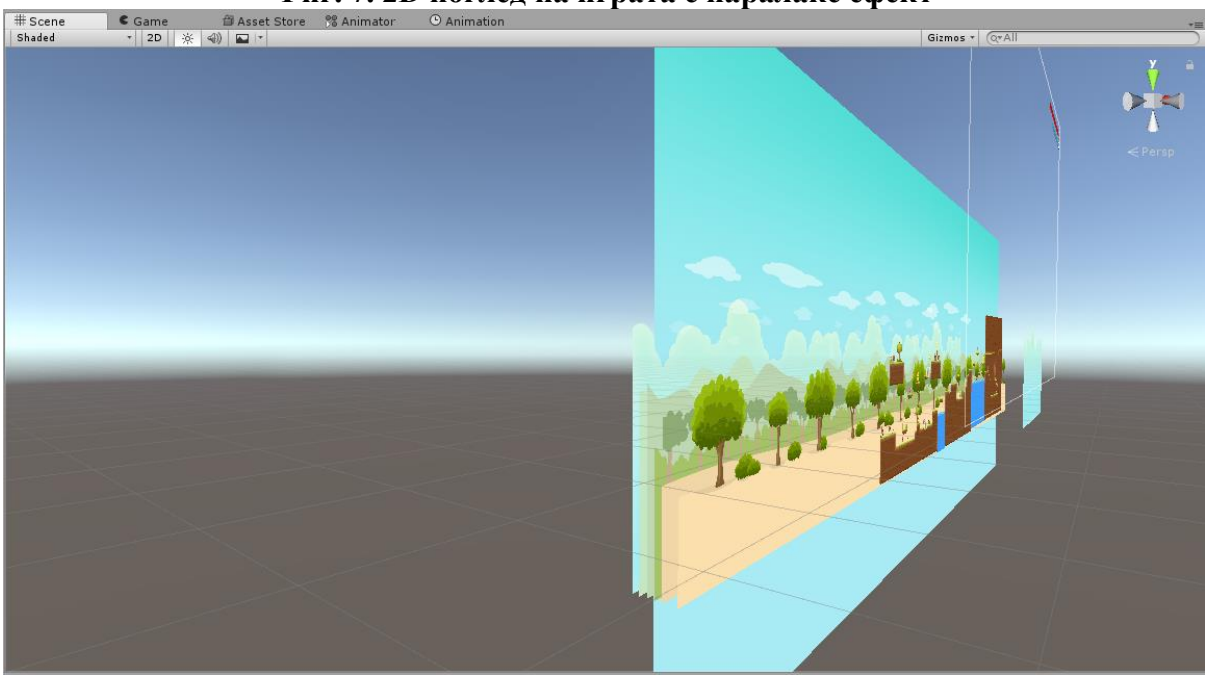
ПАРАЛАКС ЕФЕКТ

Паралакс е термин, който характеризира видимото изместване на положението на един наблюдаван обект, дължащо се на неговото наблюдение от две различни точки. За постигането на този ефект е нужно задният фон да бъде изграден от повече от един слой, като всеки от тях е поставен на различно разстояние по „Z“ координатата от предностоящия слой. По този начин, при предвижване на камерата с перспективен обектив в посока „X“ или „Y“ всеки слой се измества с различна скорост, като по-близките слоеве се изместват по-бързо, а по-далечните се изместват по-бавно.

СКРИПТОВЕ

До голяма степен скриптовете са основната и градивна част от създаването на една видео игра. Чрез тях се създават интеракциите между обектите присъстващи в дадена игра, описват се взаимодействията между тях, създават се зададени обекти или ефекти при взаимодействието на елементи от играта, и се задават различни условия и стойности на обектите. Без скриптове игрите няма да имат никаква форма на управление или интеракции между различните обекти, по скоро игра без скриптове би била просто филмче със статични анимации, без възможност за управление.

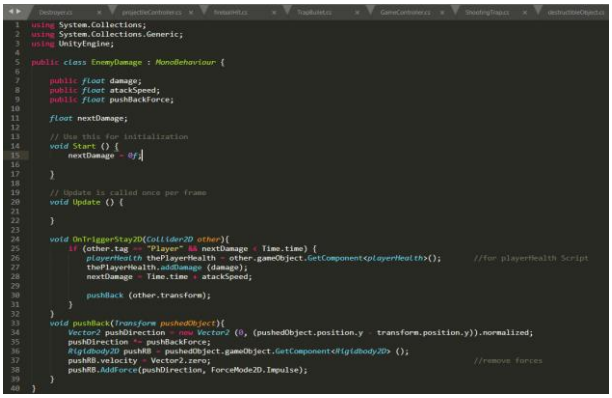
Фиг. 7. 2D поглед на играта с паралакс ефект



Фиг. 8. 3D изглед на играта с паралакс ефект

UnityEngine е библиотека, включваща физиката, класовете, ефектите и функциите, които енджинът поддържа. Чрез програмния код в скриптовете се създава и контролира поведението на различните елементи, определя се обработката на събития и последователността на действията. Като се програмира всяко едно възможно събитие се създава и интерактивността на игрите, но не винаги можем да определим дали всички събития са програмирани.

Ако съществува събитие, което не е програмирано, а се случи, то се нарича „бъг“. Бъговете водят до странни резултати при взаимодействието на обектите един с друг, било то старт на грешна анимация, движение с необичайна скорост или дори изчезване на елементи от играта. В зависимост от сериозността на бъг-а играта може да продължи, но може и да спре да функционира и да се наложи да бъде рестартирана. В повечето случаи появата на бъг, който не застрашава целостта на играта, се нарича глич (glitch), чрез който играчите могат да преминат до места, които не са предназначени за тях или да получат нещо, което може трудно да бъде намерено в дадена игра.



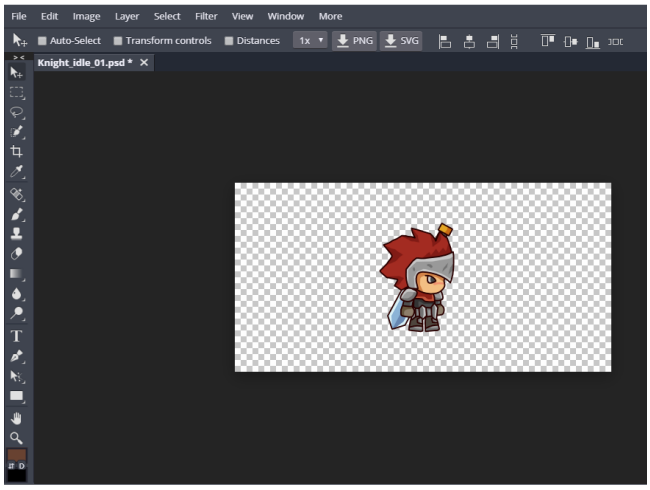
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnemyDamage : MonoBehaviour {

    public float damage;
    public float atackSpeed;
    public float pushBackForce;

    float nextDamage;

    // Use this for initialization
    void Start () {
        nextDamage = 0f;
    }

    // Update is called once per frame
    void Update () {

    }

    void OnTriggerStay2D(Collider2D other){
        if (other.tag == "Player" && nextDamage < Time.time) {
            playerHealth thePlayerHealth = other.gameObject.GetComponent<playerHealth>();        //for playerHealth Script
            thePlayerHealth.addDamage (damage);
            nextDamage = Time.time + atackSpeed;

            pushBack (other.transform);
        }
    }
    void pushBack(Transform pushedObject){
        Vector2 pushDirection = new Vector2 (0, (pushedObject.position.y - transform.position.y)).normalized;
        pushDirection *= pushBackForce;
        Rigidbody2D pushRB = pushedObject.gameObject.GetComponent<Rigidbody2D> ();
        pushRB.velocity = Vector2.zero;                                                      //remove forces
        pushRB.AddForce(pushDirection, ForceMode2D.Impulse);
    }
}
```

Фиг. 9. C# скрипт

ДОПЪЛНИТЕЛЕН СОФТУЕР

Freephototool

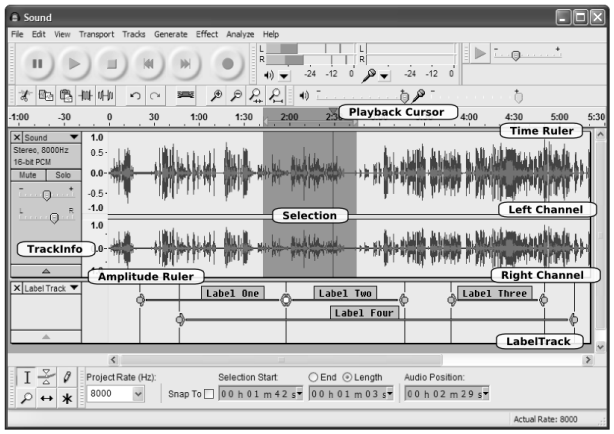
Това е алтернативен редактор подобен на Photoshop, специализиран за създаване и обработка на графични обекти. Той разполага с голям набор от инструменти и филтри, чрез които манипулацията и обработката на различните изображения е много достъпна. Възможни са множество от формати на запазване, започвайки от стандартния .PSD формат, в който се запазва целия проект заедно с всички слоеве, трансформации и папки, продължавайки с широко известните .JPEG (няколко версии), .PNG, .TIFF, както и някои типове на студио “Pixar” и др. Чрез него могат да се създават и експортират анимирани картинки (GIF).



Фиг. 10. Прозорец на freephototool

AUDACITY

Audacity е безплатен софтуер с отворен сорскод използван за създаване и обработка на аудио елементи. Той позволява обработката на аудио елементи чрез голям набор от настройки - контролиране амплитудата на звука, неговата сила и посока на изпълнение и позволява наслагване на аудио звуци за едновременното им изпълнение. Audacity позволява компресиране под различни аудио формати като .mp3, .wav, както и контролиране на bit rate-а и честотата на звука [24, 25].



Фиг. 11. Прозорец на Audacity

АРХИТЕКТУРА НА ИГРАТА

Логиката на играта е описана в множество класове и скриптове, написани на програмния език C#. Класовете са свързани чрез обектно-ориентирана структура, като връзките между тях са реализирани чрез „наследяване“, „имплементиране“, „асоциация“ и „насочена асоциация“.

Всеки скрипт е добавен към един или повече игрови обекти в дадена сцена за придаване на определените функционалности, тъй като обектите могат да взаимодействат помежду си, а в класовете трябва да се опише тяхното взаимодействие.

● Връзка от тип „наследяване“

Връзките от тип наследяване се използват от скриптовете прикачени към обектите които могат да бъдат изстреляни (като огнената топка) и от обектите, които могат да бъдат събирани (като монетите и колбите за живот и мана). FireballHit наследява от projectileController, чрез който се управлява положението и скоростта на огнената топка, както и trapBullet наследява projectileController.

● Връзка от тип „имплементиране“

Връзките от тип имплементиране се използват когато един клас изпълнява интерфейс и приема неговите методи и константи.

Имплементирането се използва от играча и противниците основно за приемане и нанасяне на щети чрез функцията addDamage, като класовете които го имплементират са enemyDamage, fireballHit и MeleeAtack.

● Връзка от тип „асоциация“

Тази връзка се използва от скрипта cameraFollow и playerHealth. При скрипта на камерата се касае за нейното свързване с главния герой, като тя взема неговите координати и ги задава като свои с цел следване на героя. А в скрипта playerHealth се използва за временното показване на екрана, индикиращ нанесени щети върху героя.

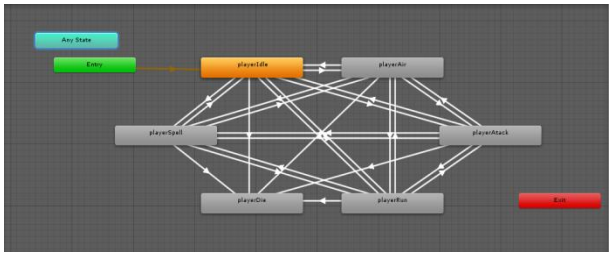
● Връзка от тип „насочена асоциация“

Връзките от този тип се използват често, тъй като повечето скриптове използват други обекти. Насочената асоциация се използва за създаване на игрови обекти, като се създават обекти в скрипта, които вземат за стойност даден обект от играта, които желаем да създаваме на дадено място. За да работи правилно играта и за да може всички взаимодействия между обектите да се извършват, трябва скриптовете да са свързани правилно, и разбира се, трябва всички обекти от класовете да имат стойност, т. е. да не са null, защото средата

Unity веднага извежда на конзолата изключение от тип NullPointerException.

Логика и структура

Героят притежава жизнени и магически точки, които определят неговото състояние, като при положителни жизнени точки героят е жив. При всеки удар нанесен върху героя се проверява стойността на моментният му живот и се изпълнява звуков сигнал както и визуален сигнал чрез премигване регулиране на прозрачността на картина зададена в DamageScreen от тип image върху екрана, а при достигането им до нула вследствие удари от противници или капани, булевата променлива dead приема стойност true и аниматора изпълнява анимацията за смърт. На фиг.12 е показан контролера на анимациите на главния герой, който подлежи на доразвиване с добавянето на допълнителни функционалности.

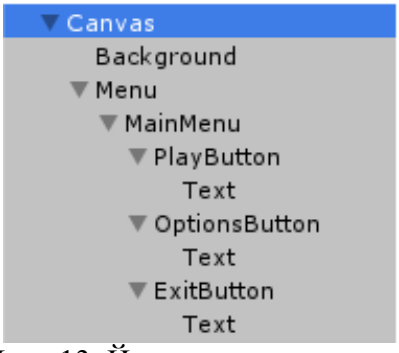


Фиг. 12. Състояния и действия на главния герой

Главен екран (GUI)

При стартирането на играта се зарежда първата сцена, от която може да бъде стартирано първото ниво в играта, да се влезе в опциите на играта или да излезе от нея чрез бутоните Play, Options и Exit. Подредени в йерархична структура, бутоните са изградени от UI елемент, създаден предварително от обектите вложени в юнити и текстови обекти от тип TextMeshPro, които позволяват голяма персонализация на текста и неговите свойства и визия [15, 19, 20].

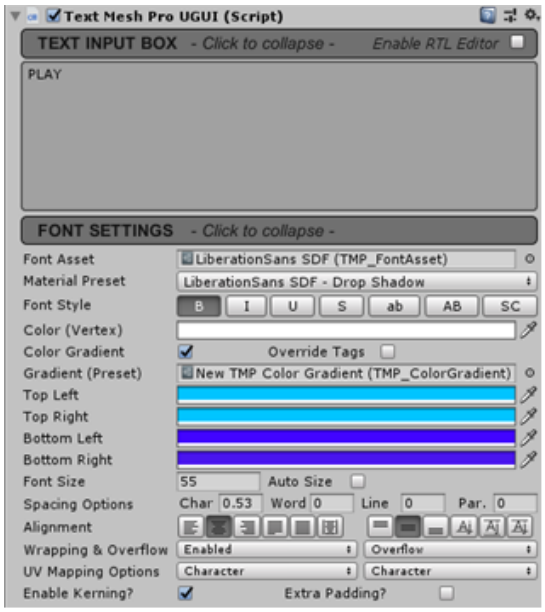
За създаването на текста са избрани подходящ шрифт, размер и стил като е включена опцията за преливане на цветовете на текста. Горната и долната страна на текста са оцветени в различни нюанси на синия цвят и е регулирано междубуквеното разстояние, както и центриране на текста.



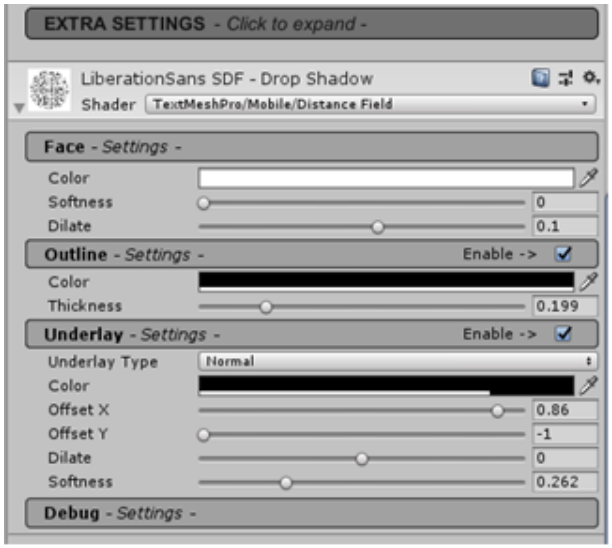
Фиг. 13. Йерархична структура на основното меню

Фиг. 14. Фон на менюто

Фиг. 15. Главно меню

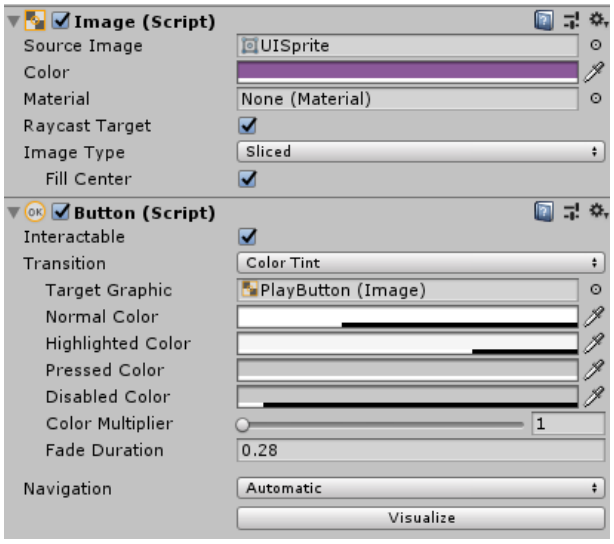


Фиг. 16. TextMeshPro настройки



Фиг. 17. TextMeshPro настройки

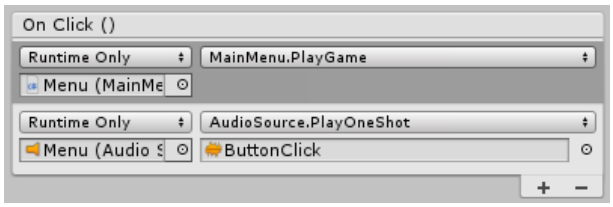
Чрез котви са позиционирани бутоните така, че при регулиране и оразмеряване на екрана, те да не губят своята позиция и съотношение на екрана.



Фиг. 18. Елементи на бутоните на главното меню

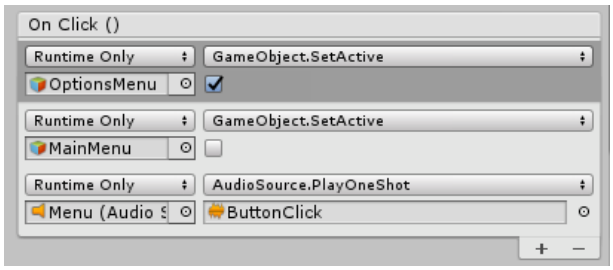
Бутоните на главното меню притежават еднакви свойства, като след създаването на единия бутон неговите елементи са дублирани и добавени към елементите на другите бутони. Разликата в бутоните е в тяхната функционалност при натискането им:

➢ **Play-** При натискането на бутона Play се извиква функцията PlayGame() от скрипта MainMenu, която зарежда следващата сцена зададена в мениджъра на активни сцени и се възпроизвежда звук, индикиращ натискането на бутона.



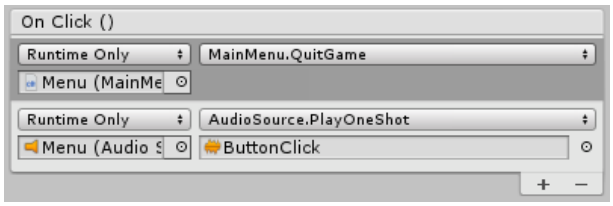
Фиг.19. OnClick() функционалности на бутона Play

➢ **Options-** При натискането на бутона Options се извиква вградената функция в библиотеката на юнити за контрол на сцените, чрез която се деактивира игровия прозорец MainMenu и се активира прозореца OptionsMenu, като се изпълнява и звуков сигнал при натискането на бутона.



Фиг. 20. OnClick() функционалности на бутона Options

➢ **Exit-** При натискането на бутона Exit се извиква функцията QuitGame() от скрипта MainMenu, чрез която се излиза от приложението и се изпълнява аудио звук за натискането на бутона.



Фиг. 21. OnClick() функционалности на бутона Exit

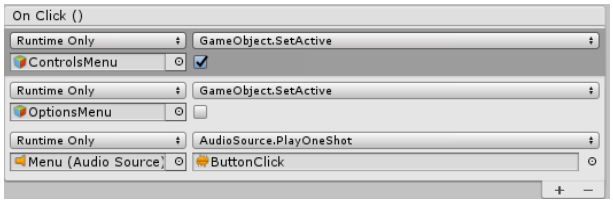
Екран за опции (GUI)

След натискането на бутона за опциите се деактивира активния екран на главното меню и се активира екрана представляващ опциите.

Менюто за опции е съставено от текст, обозначаващ текущото активно меню, както и два бутона, които са създадени по същия начин като бутоните от главното меню.

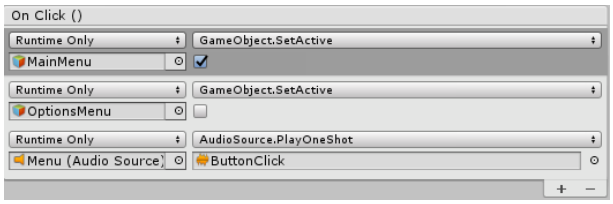
Фиг. 22. Екран на опциите

➢ **Controls-** При натискането на бутона Controls се деактивира екрана за опциите и се активира екрана, съдържащ контролите на играта, както и звук, индикиращ натискането на бутон.



Фиг. 23. OnClick() функционалности на бутона Controls

➢ **Back-** При натискането на бутона Back се деактивира текущия прозорец и се активира прозореца на главното меню, както и аудио сигнал за индикация на натискане на бутон.



Фиг. 24. OnClick() функционалности на бутона Back

ЗАКЛЮЧЕНИЕ

Играта е въображаема, нереална, безопасна среда, в която човек изпробва, тества, усвоява, тренира, репетира знания, умения и стратегии, необходими за оцеляването и развитието си. Тази среда е подчинена на определени правила, които се приемат доброволно от играчите. Тя може да представлява въображаема ситуация, пространство, терен, функционален или виртуален свят. Играта е забавна, докато поставя пред играчите предизвикателства, които е трудно да бъдат преодолени. Доброто представяне в играта осигурява на човека или на тима по-висок статус и евентуално други придобивки, които повишават мотивацията на играчите. Конкретната игра е създадена с помощта на Unity, с некомерсиална цел. Първоначално ще бъде пуснато демо на играта и ако има достатъчно голям интерес, тя ще бъде завършена и споделена в Интернет за безплатно ползване.

ЛИТЕРАТУРА

1. Динева, В. За валидността на диагностичните критерии при компютърна зависимост и дисфункционална употреба на Интернет. Научни трудове, т. 54, сер. 6. Русенски университет, 2015, с. 74-78.
2. Кръстев, Г., М. Теодосиева, Структури от данни, използвани при програмиране на игри за мобилни информационни устройства. НС, РУ „А. Кънчев”, Научни трудове, т. 46, серия 3.1. Комуникационна и компютърна техника и технологии, Русе, 2007.
3. Кръстев, Г. Методика за създаване на компютърни игри. Печатна база при Русенски университет, 2014.
4. Кръстев, Г., В. Динева. Създаване на компютърни игри с Unity 3D. Печатна база при Русенски университет, 2015.
5. Кръстев, Г., Технически решения за човеко-машинен интерфейс чрез анализ на активността на мозъка. Научни трудове, т. 53, сер. 3. 2. Русенски университет, 2014, с. 54-58.
6. Кръстев, Г., В. Динева. Програмно осигуряване за неврофийдбек. Комуникация със слушалки MindSet. Научни трудове, т. 54, сер. 3. 2. Русенски университет, 2015, с. 47-51.
7. Кръстев, Г., В. Динева. Визуализация на мозъчни вълни получени от слушалки MindSet. Научни трудове, т. 54, сер. 3. 2. Русенски университет, 2015, с. 62-66.
8. Кръстев, Г., В. Динева. Човеко-машинен интерфейс с ръкавица Peregrine. Известия на съюза на учените, т. 13, с. 1. Русе, 2016, с. 48-55.
9. Кръстев, Г., В. Динева. Web базирано приложение за обучение по ролевата игра „The Quest Adventurer”. Известия на съюза на учените, т.13, с. 1. Русе, 2016, с. 56-65.
10. Кръстев, Г., В. Динева. Човеко-машинен интерфейс с Leap Motion. Извес-

тия на съюза на учените, т. 14, сер. 1. Русе, 2017, с. 38-46.
11. Krastev, G., Magdalena Andreeva. A Softwaere Tool for Experimental Study Leap Motion.// International Journal of Computer Science & Information Technologies (IJCSIT), 2015, No6, vol.7.
12. Krastev, G., V. Voinohovska, S. Tsankov, V. Dineva. Controlling a 2D computer game with a Leap Motion. IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 19, Iss. 6, Ver. II, 2017, pp. 81-93.
13. Krastev, G., V. Voinohovska. Designing Educational Computer Games to Enhance Teaching and Learning.// IOSR Journal of Computer Engineering (IOSR - JCE), 2016, No Volume 18, I, pp. 01-10.
14. Krastev, G., V. Voinohovska, Sv. Tsankov. Web Based Reality With Structure Sensor.// International Journal of Computer Science and Technology - IJCST, 2016, No 7, Issue 4, pp. 90-93.
15. Krastev, G., V. Voinohovska, Sv. Tsankov. Designing the User Interface of a Logic Game.// IOSR Journal of Research & Method in Education (IOSR-JRME), 2016, No 6, Issue 4 V.
16. https://docs.unity3d.com/Manual/Unity2D.html
17. http://brackeys.com/
18. https://assetstore.unity.com/
19. https://itch.io/game-assets/tag-2d
20. https://kenney.nl/assets?q=2d
21. https://lifestyle.bg/archive/kratka-istoriya-na-elektronnite-igri-chast-1-galeriya.html
22. https://www.youtube.com/
23. https://en.wikipedia.org
24. https://www.audacityteam.org/
25. https://freesound.org/
26. https://stackoverflow.com

КИБЕРСИГУРНОСТ - ЗАПЛАХИ ЗА СИГУРНОСТТА И УЯЗВИМОСТИ

Иван Иванов
Висше училище по телекомуникации и пощи

Cyber Security - Threats to Security and Vulnerabilities

Ivan Ivanov
University of Telecommunications and Posts

Abstract: This article presents the results of cybersecurity analysis and research. The purpose is to acquaint the readers with some of the security issues on the Internet and to propose solutions in the fight against them. The social engineering where approved practices for protection are recommended is highlighted..

Keywords: cybersecurity, information security, network security, computer viruses, social engineering.

ВЪВЕДЕНИЕ

В днешно време почти всяко наше електронно устройство е свързано с Интернет, било то преносим компютър, мобилен телефон, таблет и т.н. Наивно е да се счита, че с появата на Интернет няма да се появят и редица проблеми, породени от злонамерени действия, насочени от потребител към Интернет и обратно, общо казано. Това налага вземането на мерки да се защитят виртуалното пространство и неговите обитатели, които често небрежно, разсеяно и невнимателно прекарват време там.

Целта на статията е да запознае читатели с част от проблемите по сигурността в Интернет пространството и да представи решения в борбата с тях.

ИЗЛОЖЕНИЕ

Информационна сигурност

Информационна сигурност – това е защита на информацията и информационните системи от неоторизиран достъп, използване, разкриване, промяна, прочитане, запис и унищожаване. Опазване на конфиденциалността, целостта и наличността на информацията. Терминът е достатъчно общ, за да бъде използван независимо от формата, която може да имат данните (напр. електронна, физическа).

Според определението на Европейската комисия информационната сигурност е защита на мрежите и информационните системи срещу човешки грешки, природни бедствия, технически неизправности или злонамерени атаки.

IT сигурността, наричана понякога компютърна сигурност, означава информационна сигурност, приложена към техниката (най-често под формата на компютърна система) [1]. При това компютър не се ограничава до персонален компютър, а е всяко устройство с централен процесор и компютърна памет. Това могат да са както самостоятелни, несвързани с други, прости устройства като калкулатори, така и свързани в телекомуникационна мрежа мобилни устройства като смартфони и таблети. Поради естеството и ценността на данните във всяко по-голямо предприятие има отдел по IT сигурност. Той отговаря за опазването на компанията от злонамерени атаки, които се опитват да се доберат до критична чувствителна информация или да осъществят контрол върху вътрешни системи.

Информационната защита (на английски: Information assurance)[2] е предотвратяването на загуба на данните при критични ситуации. Такива могат да включват, без ограничение: природни бедствия, повреда на компютри/сървъри, кражба или всеки друг случай, когато има риск от загуба на информацията. Тъй като в наши дни повечето от информацията се съхранява на компютри, за информационната защита най-често се грижат специалистите по IT сигурност. Един от най-честите методи за информационна защита е дублирането на

информацията и съхраняването на резервни копия на различно място.

Заплахите за сигурността на информацията приемат най-различна форма. Някои от най-честите са софтуерни атаки, кражба на интелектуална собственост, кражба на самоличност, кражба на устройство или информация, саботаж и манипулиране на информацията. Повечето хора са изпитали някакъв вид софтуерна атака: вируси, червеи, фишинг и троянски коне са примери за такива атаки. Кражбата на интелектуална собственост е предмет на загриженост за много IT компании, като най-честа е кражбата на софтуер. При кражбата на самоличност атакуващият се опитва да получи достъп до лична информация, за да се възползва от нея по злонамерен начин. Кражбата на устройства или информация е често срещана днес поради факта, че все повече информационни устройства са мобилни. Мобилните телефони са чест обект на кражба и са все по-желани с увеличаването на обема на съхраняваната информация. Саботажът може да приеме формата на повреждане на уебсайта на компанията в опит да се причинят вреди на потребителите. Манипулирането на информация се прави с цел да се изнудва собственика да заплати възстановяването на вярната информация или да получи обратно собствеността си [3].

Правителствата, военните, корпорациите, финансовите институции, болниците и частните компании натрупват голямо количество конфиденциална информация за своите поданици, служители, клиенти, продукти, изследвания и финансови операции. В наши дни по-голямата част от тази информация се събира, съхранява и обработва с помощта на компютри и се предава по компютърни мрежи на други компютри [4].

Съвременни кибер заплахи

Като цяло съвременните кибер заплахи, могат да се разделят в пет основни групи (фиг.1):

1. Физически заплахи и уязвимости;
2. Мрежово базирани заплахи;
3. Заплахи и уязвимости към безжичните мрежи;
4. Софтуерно базирани заплахи;
5. Социално инженерство.

Физически заплахи и уязвимости

Физическа сигурност – осигуряване физическата сигурност на периметъра, за да се предотврати неоторизиран физически достъп чрез внедряване на механизми за физически контрол. Този тип заплахи могат да се разделят на следните категории [5];

- Вътрешни заплахи – недоволни служители или служители преди напускане;
- Външни заплахи – прекъсване на захранването или други услуги;
- Природни бедствия – наводнения, пожари, високи температури и влажност;
- Причинени от човека – умишлени и неосъзнати.

Мрежово базирани заплахи

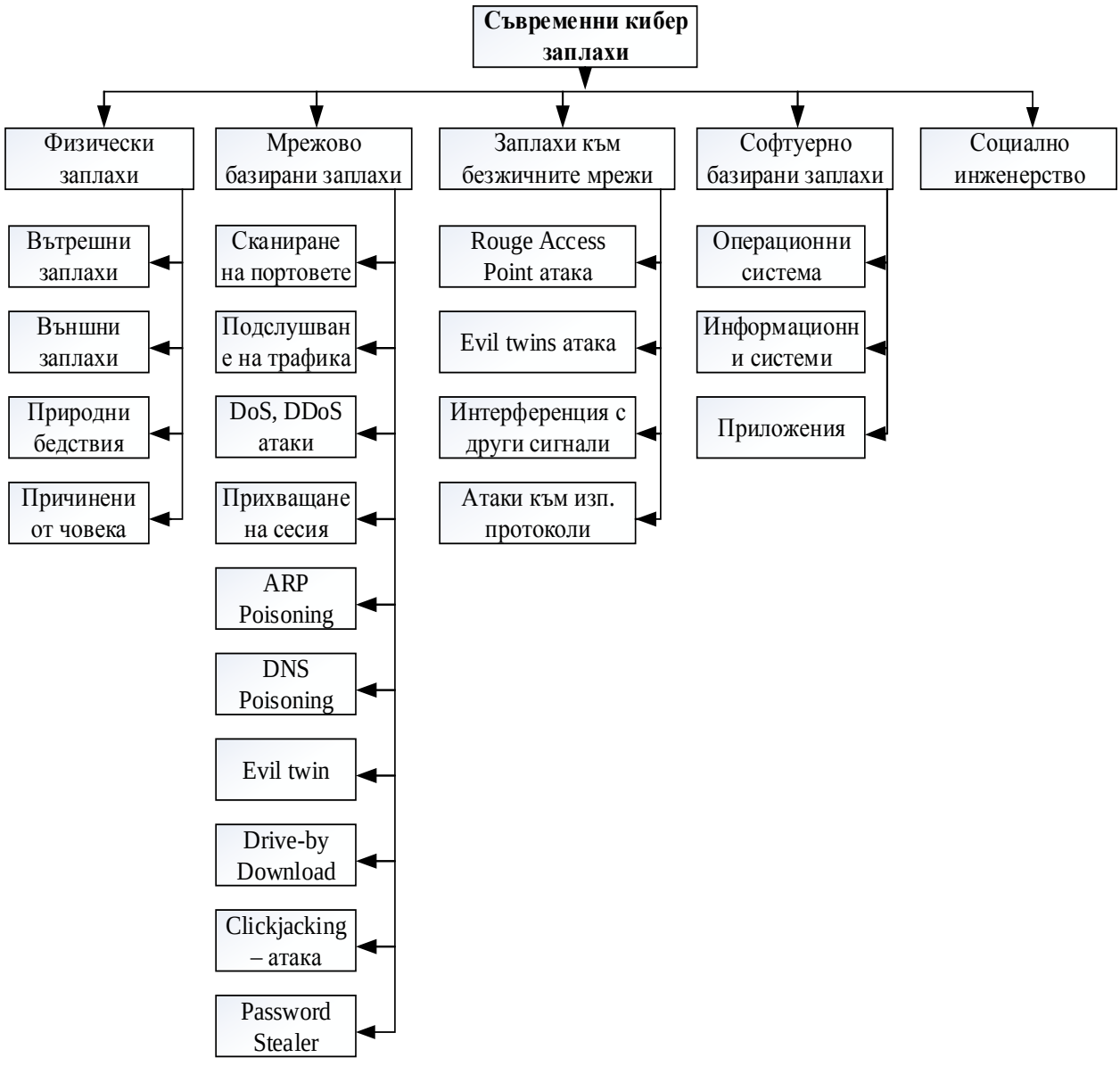
Най-разпространените в днешно време атаки, базиращи се на различни методи, подходи, протоколи и други техники за достъп [6]. Те се разделят на:

- Атаки чрез сканиране на портовете (Port Scanning Attacks);
- Атаки чрез подслушване на трафика (Eavesdropping);
- DoS, DDoS атаки - Атака за отказ на услуга (на английски: denial-of-service attack, съкратено DoS attack — DoS атака) е опит даден ресурс, предоставян от компютър (наричан жертва), да бъде направен недостъпен за целевите му потребители. Атаката може да бъде чрез изтощаване на ресурси или чрез възползване от грешка в софтуера на жертвата. Най-често биват атакувани популярни уеб сървъри, като целта е те да станат недостъпни от Интернет. Според Борда на архитектите на Интернет, това действие е компютърно престъпление, нарушаващо Етиката в Интернет;
- Прихващане на сесия – Man-in-the Middle атака – атакуващият установява точка между двете страни, като по този начин може да модифицира данните;
- ARP Poisoning (Address Resolution Protocol) – MAC адреси към IP адреси. При тази атака атакуващият пренасочва IP адреса към друг MAC адрес;

- Атаки към DNS сървъри – DNS Poisoning – Модифициране на DNS записите;
- Атака на Злия Близнак (Evil twin) – създаване на фалшив профил на истинска личност и свързване с приятелите на атакувания, събиране на лична информация;
- Drive-by Download – отвеждане на потребителя от социалната мрежа към сайт със зловреден софтуер;
- Clickjacking – атака, която принуждава потребителя на кликне върху злонамерен линк;
- Крадец на пароли (Password Stealer) – тип софтуер, който се инсталира от социалната мрежа, прихващащ паролите на потребителите.
-

Заплахи и уязвимости към безжичните мрежи

От всички мрежи безжичните са най-уязвими и подходящи за атака, поради това че могат да бъдат достъпни извън физическите граници на организацията. Всички безжични устройства са с настройка по подразбиране за свързване към мрежата с най-силен сигнал, която може да е изградена от атакуващия.



Фиг. 1. Класификация на съвременните кибер заплахи

Една **корпоративна WLAN система** трябва да включва следните функционалности [7]:

- Аутентификация на потребителите - възлите в мрежата;
- Решение на проблема с криптирането на данните;
- Идентифициране на външни точки за достъп;
- Детектиране на ad hoc или злонамерени потребители;
- Позволяване на потребителски достъп от тип „гост" (guest access);
- Планиране и управление на RF обхвата;
- Локализиране на източника на пасивни и активни атаки от тип DoS and man-in-the-middle.

От своя страна, **безжичните атаки и уязвимости** се разделят на:

- Rouge Access Point атака – неоторизиран access point поставен в мрежата на организацията;
- Evil twins атака – access point устройства, които изглеждат легитимни, обикновено установени в публични зони, които примамват потребителите да се свържат със тях;
- Интерференция с други сигнали – обикновено в домашна обстановка, което кара сигналът да прекъсва;
- Атаки към използваните протоколи за защита.

Контролът на достъпа и криптирането са жизнено важни за осигуряването защитата на данните и предотвратяването на атаките в безжичните локални мрежи. Дискредитираният статичен WEP (Wired Equivalent Privacy), вече е заменен от множество опции за шифриране, които дават възможност на IT специалистите за избор на най-добрата налична стратегия и планиране [8]. Проектирането на безжична локалната мрежа, както и приложните инструменти за разпознаване на вмешателство са съществени стъпки за построяване на сигурна WLAN

Докато 802.1X е широко приет стандарт, като решение за аутентификация при една безжична LAN, то са налични множество опции за безжично криптиране. Те включват Wired Equivalent Privacy (WEP) със статични и динамични ключове; Wi-Fi Protected Access (WPA) 1.0, който използва 802.1X и протокол за временната цялостност на ключовете - Temporal Key Integrity Protocol (TKIP); WPA 2.0, който използва 802.1X и стандарт за подобрено криптиране - Advanced Encryption Standard (AES); и Ipsec виртуални частни мрежи (VPN), стандарт който може да бъде разработен като стандартно VPN решение или като ултра сигурен стандарт за обработване на държавна информация Federal Information Processing Standard (FIPS) 140-2 ниво 1 или ниво 2, който определено е предназначен за правителствени реализации. IEEE организацията се очаква да завърши 802.11i стандарта през 2004, който дефинира и включва както TKIP така и AES.

Софтуерно базирани заплахи

Атаки извършени срещу определен софтуер – операционна система, информационна система или приложение.

Цел – да се спре функционирането на даден обект или да се наруши обработваната от него информация.

Компютърни вируси - Дефиниция

Вирус е само размножаваща се програма, която се разпространява като вмъква копия от себе си в друг изпълним код (програми) или документи. Това дава и името на този вид програми, тъй като подобно поведение е сходно с това на биологичен вирус, който се размножава като се вмъква в живи клетки. По аналогия, вмъкването на вирус в програма често се нарича „инфекция" [9]. Вирусите са само един от видовете злонамерени програми, но в разговорния език термина често се използва да обозначава и представители на другите видове, като троянски коне и червеи. Както всяка друга програма, компютърният вирус следва предварително зададени логически инструкции за действие. Една вирусна програма може да има различни версии, като те могат да са от един и същ или различни автори.

Компютърните вируси могат да се разделят на следните видове [10]:

- Boot секторен вирус - Този тип вируси инфектира boot записа на хард диска. Отначало той премества или презаписва оригиналния boot запис като го заменя с инфектиран. Преместеният оригинален запис се записва в сектор, който вирусът маркира като повреден, за да не се използва повече (антивирусните приложения няма да го сканират, защото е повреден). За да се заразите с такъв вирус трябва да стартирате компютъра си от инфектирана дискета, CD, DVD, Blu-Ray, флашка или друг носител на данни. При проверка на boot сектора, вирусът заблуждава антивирусната програма като я насочва да сканира чистото копие вместо заразеното.
- Файлов вирус - Това е един от най-разпространените видове вируси. Тези вируси търсят файлове с определено разширение (обикновено изпълними файлове като *.com и *.exe) и ги инфектират. Когато програмата бъде отворена, вирусът се стартира и инфектира още файлове.
- Макро вируси - Тези вируси използват специални програми и поддържаните от тях файлове, за да се размножават. Макро вирусите обикновено заразяват файлове на MS Excel, но могат да инфектират и други файлове, които използват програмен език. Те не инфектират програми. Когато бъде отворен заразен документ, вирусът се разпространява и на други документи. Макро вирусите могат да нанесат големи поражения на документите, които се съдържат в заразения компютър. Неприятното е, че могат да се разпространяват между различни операционни системи.
- Полиморфични вируси - Тези вируси променят своя код с всяка инфекция, което ги прави трудни за засичане. Мултиполиморфичните вируси пък от своя страна инфектират boot записи и изпълними файлове. Тези вируси могат да комбинират предимствата на полиморфичните и стелт вирусите.
- Стелт вируси - Лесно избягват сканиранията на антивирусните програми (чрез криптиране на програмния си код или се крият в паметта) и им пречат да открият промените в заразените от тях файлове, като им предоставят стари данни за същите файлове. Това ги прави трудни за засичане.
- Rootkit или руткит - представлява зловреден код, който се разпространява като инфекция чрез мрежи за обмен на данни от типа равен-с-равен (peer-to-peer) или от повредени уеб-сайтове, които са инфектирани с такъв зловреден код. Обикновено този тип малуер работи с някаква технология за покриване, което прави неговото детектиране и отстраняване значително усложнено. Добре работещият руткит може да бъде неоткриваем. Той не оставя следи по логовете. Показва фалшифицирани конфигурационни файлове и/или ключове в регистратурата, от които не личи неговото стартиране. Показва фалшиви данни за работещите процеси и натоварването на процесора. Не дава да се видят, променят или трият неговите файлове. Накратко: работещият на този компютър вижда не истинското положение, а това, което му показва руткитът.
- Спайуерът (Spyware) е софтуер, който може да извърши определени действия на Вашият компютър, без да знаете за тях, като показване на реклами, събиране на персонална информация или промяна на настройките на компютърът Ви. Други видове нежелан софтуер могат да изменят Вашият компютър в различна степен - от незначително дразнене до голямо забавяне или замръзване на системата. Тези програми имат способността да променят началната страница на Вашият браузър, страницата за търсене или да добавят допълнителни компоненти към браузъра, които не са ви необходими или не искате. Също така тези програми могат да направят много трудно връщането на настройките към първоначалните им стойности. Тези видове нежелани програми се наричат спайуер.
- Адуер (на английски: adware) е софтуер, който се разпространява основно с легитимни програми и предназначението му е да показва рекламни съобщения и/или банери, като по този начин да генерира печалба както за автора на рекламния софтуер, така и за автора на програмата, с която се е инсталирал той. В повечето случаи рекламният софтуер е безвреден. Досаден, но безвреден. Има и изключения обаче.

Някои рекламни приложения съдържат програми за запис на клавиши, събират някаква информация за потребителя и други.

- Рансъмуерът е тип зловреден софтуер, който блокира достъпа на жертвите до компютъра им и изисква плащането на откуп. Откупът и официалната причина да бъде платен зависи от вида вирус. Някои версии на рансъмуера твърдят, че плащането трябва да бъде направено, за да се избегне наказание от държавна власт (обикновено ФБР или местна агенция), други информират, че това е единственият начин да декриптирате данните си. Допълнителни дейности, които могат да извършват рансъмуерните паразити, включват крадене на лична информация, блокиране на легитимен софтуер (антивирусни, анти-спайуер и т.н.), показване на предупреждения и други нежелани дейности. Един от най-често срещаните е криптиращият рансъмуер който главно се разпространява с помощта на троянци. Веднъж инфилтрирал се, той намира често ползвани файлове и ги криптира. Заключените файлове включват снимки, музика, видео, изкуство, бизнес и други данни, които се смятат за важни за жертвата. В допълнение подобен рансъмуер започва да показва голямо съобщение за предупреждение, което твърди, че единственият начин да декриптирате данните си, е да платите откупа. Всъщност са прави, защото повечето подобни малуери изтриват и шадоу копията на файловете, за да предотвратят възстановяване.
- SPAM, SPIM - Спам (spam) е използване на среда за електронни комуникации за масово изпращане на нежелани съобщения. Най-известната форма на спам е под формата на съобщения с рекламно съдържание по електронната поща. Спамът, разпространяван чрез услугите за моментни съобщения е по-известен като SPIM.
- Phishing, Pharming - Фишинг или в превод риболов е една от най-разпространените измами в Интернет. При него се използва социално инженерство или директна кражба на информация.
- Червеи - Червеят е програма, която се размножава, но не инфектира други програми. Той заразява компютри независимо дали са част от мрежа или не. Червеят се копира от и на флопи дискети, CD, DVD, Blu-Ray и "флашки", както и на различни дялове на хард диска. Ако заразеният компютър е част от мрежа, той може да инфектира и другите компютри в нея. Червеите често крадат и унищожават данни и се разпространяват основно чрез Интернет.
- Троянски коне - Троянският кон е зловредна програма, която е скрита в безобидна такава. Когато тази програма бъде стартирана, се стартира и троянският кон, за да изпълни определена задача. Троянските коне могат да откраднат лична информация (пароли, потребителски имена), да изтрият файлове, да форматират твърдия диск и др.
- Droppers (контейнери) - "Контейнерът" е такава програма, която е създадена да преодолее антивирусната защита на компютъра - обикновено чрез криптиране, което пречи на антивирусната програма да го засече. Тя се спотайва в компютъра до настъпването на определено събитие и тогава го инфектира с вируса, който съдържа, т.е. тя действа като "контейнер за вируси". Оттам идва и наименованието й.
- Бомби - Бомбата е вреден скрипт или програма, която се задейства при изпълнението на определени условия. Някои бомби се активират на определени дати като използват системния часовник. Например, бомбата може да бъде програмирана да изтрие всички *.doc файлове на Нова година. Друга бомба може да изчака да бъде отворена за 17 път и тогава да се задейства.
- Експлойт - Ако следваме дефиницията, то експлойтът съвсем не е вирус или зловреден код. Защото експлойтът е приложение, което е създадено със специалната цел да използва (експлоатира) уязвимост в друго приложение - или иначе казано, да се възползва от грешка в програмната архитектура. Например, ако става въпрос за пробойна в операционната система, целта на експлойта обикновено е да даде на създателя му администраторски права върху атакувания компютър - и съответно, пълен контрол.

Социално инженерство

Само по себе си, социалното инженерство рядко има връзка с интернет или

мрежовите технологии, разбиването на пароли или кибер атаките, но въпреки това, може да свали гарда на всяка компания, дори да разполага с превъзходни системни администратори, VPN, защитни стени, силни пароли, най-новите security протоколи, системи за мониторинг на мрежи и т.н. Винаги има едно непредвидимо звено в системата и то се нарича "човешки фактор".

Както казва Кевин Митник, може би най-известният (бивш) хакер в областта на Social Engineering, "много по-лесно е да подлъжеш някой да ти даде паролата за дадена система, отколкото да загубиш часове в усилия да я разбиеш". На последната DEF CON Hacking Conference това се потвърди по един нестандартен начин.

Преди време Social-Engineer.org организираха състезание между участниците, в рамките на което всеки можеше да тества social engineering уменията си, опитвайки се да събере важна вътрешна информация от компании като Coca Cola, Shell, Ford, дори ИТ гиганти като Google, Microsoft, Apple, Cisco и други. Изненадващо, теста се оказва доста успешен - представяйки се за журналисти, бизнесмени и IT специалисти, хакерите успяват да получат информация като спецификациите на проекти със затворен код или вида и версията на браузъра и антивирусния софтуер, използвани от служителите. Някои от участниците дори успяват да убедят отсрещната страна да посети URL, продиктувано по телефона.

Разговорът е само един от методите на социалното инженерство. Наред с него "досието" на жертвата може да се попълни и с търсене в Google, следене на Facebook, Twitter, LinkedIn и др. онлайн профили на жертвата, личен контакт, комуникиране с приятели и колеги на служителя и т.н.

Все пак положението не е съвсем безнадеждно - съществуват редица утвърдени практики за защита срещу този род атаки. По-долу са описани част от тях. Според Крис Хаднаги, съосновател на Social-Engineer.org, ако служителите не познават типовете атаки и основните методи на прилагането им, няма как да се защитят срещу тях. Двата най-често използвани и ефективни подхода за получаване на информация са представяне за вътрешен служител или за външно лице, нает за одит или проучване във фирмата - така въпросите не изглеждат странно и извън контекста, а служителите рядко търсят потвърждение от ръководното ниво.

Утвърдени практики за защита срещу социално инженерство:

- Внимателно споделяне

В корпоративния живот няма маловажна информация. Често пъти това важи и за личния живот на служителя - независимо дали споделяте "очи в очи" или през социалните мрежи, един необмислено изнесен факт може да ви донесе неприятности. Учудващо е колко много потребители в социалните мрежи споделят с напълно непознати къде живеят, кога ходят на работа, дори какъв е достъпа до работното място или какъв софтуер за защита ползват. Специалисти по сигурността съветват да не се предоставя информация (независимо от характера и), когато няма реална нужда от нея.

- Определяне на най-ценните активи за хакерите

Дори компании, които инвестират стабилно в защита от социално инженерство, често допускат грешката да се фокусират върху защита на погрешни страни. Когато компаниите мислят за защита на активите си, те го правят от гледна точка на бизнеса. Хакерите, обаче не винаги гледат от същата перспектива. За тях е важно да имат информация, от която могат да извлекат материална или друг вид изгода. Когато организациите вземат решение кое е важно да се защити, трябва да обмислят нещата внимателно. Често най-добрата оценка се дава от външни независими специалисти.

- Фирмени политики

Когато знаем кои са най-ценните ни активи, както и по какъв начин злонамерени лица могат опитат да се сдобият с тях, напишете политика за сигурност, включваща конкретни мерки за защита на информацията. За да сме сигурни, че ще бъдат спазвани е добре да се проведе обучение на персонала При това обучението трябва да обхваща всички служители на компанията, независимо от позицията и отговорностите. Едно от най-атакуваните звена са call-

центровете или отделите за връзка с клиента. Политиката трябва да включва точни и ясни инструкции за поведение в широк спектър от ситуации. Повечето служители предоставят твърде лесно информация в желанието си да помогнат или съдействат на хакера.

- Често обновяване на софтуера

Понякога хакерите използват социално инженерство за да разберат дали могат да се възползват от остарели версии на различни приложения или такива, на които не са добавени критични пачове и допълнения. Дори да се доберат до такава информация, редовният ъпгрейд и мониторинг на софтуера свежда риска от атаки до минимум.

- Неангажиращо отношение от страна на служителите

Както мнозина специалисти по човешки ресурси биха потвърдили, паричните стимули далеч не са най-важни за ефективността на служителите. Много компании не успяват да събудят съпричастност и чувство на колективност в персонала си. В резултат голяма част от служителите не смятат, че сигурността на компанията ги касае пряко и не считат за необходимо да познават в детайли политиката за сигурност. Важно, обаче е да се разбере, че хакера не поставя граници и служебната информация трябва да се пази както в офиса, така и вкъщи.

- Техники при искане на информация

Хората инстинктивно се стремят да бъдат максимално полезни и изчерпателни, когато някой помоли за помощ. Социалните инженери също са наясно с този инстинкт и ефективно се възползват от него. Въпреки това, компаниите трудно биха си позволили да загубят потенциални клиенти, отклонявайки въпросите им поради политики за сигурност. Затова се налага предварително да се определят границите на релевантните въпроси и на тези, които излизат извън контекста.

- Следване на протокола

Най-добрия начин за защита от социалното инженерство е стриктно да следвате разписаните политики за сигурност. Често пъти служителите се поддават на настойчиви молби и дават информация, която иначе не биха споделили. Ако служител не е убеден , че трябва да отговаря на зададените въпроси, по-добре да насочи клиента към мениджърите си или отдела, който е най-близо до темата на разговора. Повечето хакери се отказват, ако бъдат пренасочени към друг източник. Друга много ефективна мярка е компанията да има политика въпросите да се изпращат по електронен път - това също отказва социалните инженери.

Заключение

В днешно време всяка една организация, трябва да обръща изключително внимание на защитата на информацията, като създаде необходимата структура, документи и организация, влагайки достатъчно средства и периодично обучение на персонала. Всичко това е задължително да се реализира след много задълбочен анализ на уязвимостите и възможните заплахи за сигурността.

Литература

[1] Meeuwisse R. Cybersecurity for Beginners, Cyber Simplicity Ltd; 2 edition March 14, 2017.

[2] Donaldson S, Siegel S, Williams C, Aslam A. Enterprise Cybersecurity: How to Build a Successful Cyberdefense Program Against Advanced Threats, Apress; 1st ed. edition May 20, 2015.

[3] Johnson T. Cybersecurity: Protecting Critical Infrastructures from Cyber Attack and Cyber Warfare, CRC Press; 1 edition April 16, 2015.

[4] Hubbard D, Seiersen R, Geer D, McClure S. How to Measure Anything in Cybersecurity Risk, Wiley; 1 edition July 25, 2016.

[5] Smith R. Elementary Information Security 2nd Edition, Jones & Bartlett Learning; 2 edition March 8, 2015.

[6] Stamp M. Information Security: Principles and Practice 2nd Edition, Wiley; 2 edition May 3, 2011.

[7] Andress J. The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice 2nd Edition, Syngress; 2 edition, May 20, 2014.

[8] Kim D, Solomon M. Fundamentals of Information Systems Security 3rd Edition,

Jones & Bartlett Learning; 3 edition (October 26, 2016).
[9] Ludwig M, Noah D. The Giant Black Book of Computer Viruses, American Eagle Books November 17, 2017.
[10] Filiol E. Computer Viruses: from theory to applications, Springer; Pap/Cdr edition March 30, 2006.

Адреси за контакти:
гл. ас. д-р инж. Иван Иванов
Катедра Информационни технологии
Висше училище по телекомуникации и пощи
E-mail: i.ivanov@utp.bg

СИМУЛАЦИОННО ИЗСЛЕДВАНЕ НА ПРОТИВОПОЖАРНА СИСТЕМА ЗА РАБОТА В ОБЛАЧНА СТРУКТУРА

Филип Цветанов[1], Мартин Пандурски[2]
Югозападен университет, Благоевград
Висше училище по телекомуникации и пощи

Simulation Research of the Fire System Working in Cloud Structure

Filip Tsvetanov[1], Martin Pandurski[2]
[1]South-West University, Faculty of Engineering, Blagoevgrad, Bulgaria
[2]University of telecommunication and post, Faculty of telecommunication, Sofia, Bulgaria

Abstract: Cloud technologies enable users to integrate their equipment into a single system, manage it more easily, minimize technical interruptions, significantly reduce hardware, software, and human resources costs, and increase productivity seriously. An important prerequisite for the use of cloud technologies is the fast-paced Internet technology of things, which leads to an increase in the number of objects that connect via real IP addresses and the exchange of information between them. The Internet of Things helps the collection and analysis of large volumes of information and the automation of various processes. Cloud technologies as a paradigm for a new business technology model are of great interest to the research community. This article proposes an algorithm for simulating the operation of an early-warning fire alarm system built from smart devices working on the Internet of Things Technology and transmitting, storing data in a cloud structure..
Keywords: Internet of Things, fire protection systems, sensors.

ВЪВЕДЕНИЕ

Интернет свързва не само компютри, но и множество други устройства, с които се взаимодейства. В бъдеще много от устройствата в дома ни също ще могат да се свързват с интернет, за да се наблюдават и конфигурират от разстояние [8].

IoT е технология, която създава четири взаимосвързани компонента от хора, процеси данни и устройства, като по този начин правят мрежовите връзки по-полезни от всякога. Информацията от тези връзки води до решения и действия, които създават по-богат опит и безпрецедентни икономически възможности за хора, фирми и държави [9].

Взаимодействието между елементите в четирите компонента създава многообразие от нова информация. Компонентите си взаимодействат по начин, който създава три основни връзки в средата на IoT: хората комуникират с хора (P2P), машините комуникират с хората (M2P) и машините комуникират с машини (M2M).

P2P връзките са съвместни решения, които използват съществуващата мрежова инфраструктура, устройства и приложения, за да позволят безпроблемна комуникация и сътрудничество между хората.

В M2P връзките, техническите системи взаимодействат с физически лица и организации, за да предоставят или получават информация.

M2M се отнася до всяка технология, която позволява на мрежовите устройства да обменят информация и да извършват действия без участие от страна на хората. Комуникацията M2M отговаря за формата на предаване на данни, която позволява на машината да комуникира с машина по жичен или безжичен път. M2M използва устройства за събиране и обработка на събития чрез сензори или измервателни уреди, които се предават през мрежата.

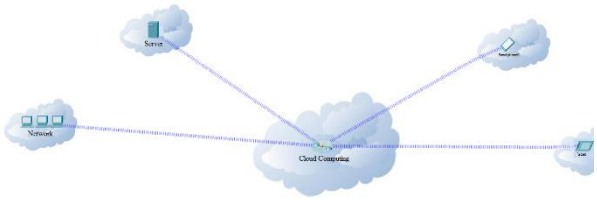
I) Модел на облачни структури (Cloud)

Технологията на облачните структури, дава почти безкрайни възможности за бързо съхранение и обработка на данни, както и адаптивност към различни приложения, което я прави привлекателно решение за събиране на данни, генерирани от различни интегрирани системи, предназначени да наблюдават различни природни явления и да предават тези данни в реално

време [8].

Облачните структури са различни от модела клиент- сървър в който сървъри и услуги са разпръснати по цялото земно кълбо в разпределени центрове за данни. Според авторите в работа [9], облачните структури позволяват на крайните потребители да получават достъп до приложения в облака намиращи се в сървъри вместо да изискват устройство на крайния клиент.

Облачните структури, имат възможност да съхраняват копия от информация в много сървъри. Различни институции и организации се абонират за различни услуги в облака (фиг.1)



Фиг.1. Облачните структури позволяват достъп до приложения от сървъри, намиращи се в облака.

Отделните организации вече не са отговорни за поддържане на актуализациите на приложения, сигурността и архивите. Това се превръща в отговорност на организацията, предлагаща облачната услуга [1].

2)Крайни устройства

Крайните устройства се свързват с Интернет, с цел изпращане на данни. Клетъчните телефони, лаптопи, компютри, принтери и IP телефони са примери на крайни устройства, използващи Интернет протокол (IP). Днес има нови типове крайни устройства, които обработват информация, но използват различни протоколи като IEEE 802.15 и NFC. Крайните устройства са или източникът, или дестинацията на данни, предавани по мрежата. За да се различи едно крайно устройство от друго, всяко крайно устройство в мрежата се идентифицира чрез адрес. Когато крайното устройство инициира комуникация, то използва адреса на устройството получател, за да му изпрати съобщението [9].

Сървърът е крайно устройство, което има инсталиран софтуер, който му позволява да предоставя информация, като имейл или уеб страници, до други крайни устройства в мрежата. Например, сървърът дава заявка за уеб софтуер за да предоставя уеб услуги в мрежата. Клиентът е като крайно устройство, което има инсталиран софтуер, за да може той да изисква и показва информацията, получена от сървъра.

3) Сензори

Сензорът е устройство, който измерва физически явления, и конвертира тази информация в електрически или оптичен сигнал, който може да бъде обработван от компютри [2]. Сензорът трябва да бъде свързан с мрежата за данни и да отчитат: топлина, тегло, движение, налягане, влажност и др. Сензорите обикновено са програмирани за измерване на определена физична или химична величина [2]. Някои сензори могат да бъдат конфигурирани, за да се промени тяхната степен на чувствителност или честотата на обратната връзка. Настройката на чувствителността показва до каква степен изхода на сензора се променя, когато се променя измереното количество. Контролер, който може да включва графичен потребителски интерфейс, се използва за промяна на настройките на сензора, локално или дистанционно.

В табл.1 са показани приложения на сензорите в зависимост от решаваната задача

Таблица 1

Нефт и газ	Градове	Транспорт	Селско стопанство	Медицина
Сензорите отчитат различни химически нива, като въглероден двуокис.	Сензорите отчитат налягане температура и влажност, запрашеност, замърсяване на въздуха.	Сензори измерват използване на гориво, повреди на двигателя,	Сензори засичат влага на почвата, слънчева радиация, не и др.	Сензори измерват: кръвно налягане, пулс, температура

Параметрите на сензорите, които играят важна роля за определянето на областта на тяхното приложение са:

Чувствителност (Sensitivity): промени на входа на сензора водещи до промяна на сигнала на изхода му;

Обхват (Range): минималната и максималната стойност на измерваната физична величина;

Прецизност (Precision): Сензора подава на изхода си едни и същи данни при едни и същи входни величини;

Точност (Accuracy): Разликата между действителната и измерената стойност от сензора;

Резолюция (Resolution): Възможността на сензора да „усети“ промяна във входният параметър, която да изведе в изходния сигнал;

Време за реакция (Response time): времето за което сензора променя изходната стойност при промяна на входната.

4) Изпълнителни устройства

Изпълнителните устройства се използват, за преместване или контролиране на механизми или системи, основани на определен набор от инструкции. Изпълнителните устройства могат да извършат физически функции [6]. Един тип изпълнително устройство е електрическа намотка, използвана за контрол на хидравлика [7].

Има три вида изпълнителни устройства, използвани в Интернет на нещата: хидравлични, пневматични и електрически.

Независимо от това как тези уреди създават движението, основната им функция е да получат сигнал, и въз основа на този сигнал, да извършват набор от действия. Те обикновено не могат да обработват данни. Вместо това, резултатът от действията им се основава на приет сигнал. Действието извършено от тях, обикновено се създава от сигнал на контролер [6].

5) Контролери

Контролерът е отговорен за събирането на данни от сензорите и предаване на обработените данни по мрежата. Контролерите може да вземат незабавни решения, базирани на програмния им код, или да изпратят данни на по-мощен компютър за анализ [3].

Освен изпращането на основна информация в конфигурацията на 1оТ, контролерите са в състояние да извършват по-сложни операции. Контролерите могат да приемат и обработват информация от множество сензори или да извършват основни анализи на получените данни. Информацията, събирана от сензорите и контролерите може допълнително да се анализира и да бъде достъпна чрез мобилни и отдалечени устройства [7].

АЛГОРИТЪМ ЗА ИЗГРАЖДАНЕ НА ПРОТИВОПОЖАРНА СИСТЕМА

Автоматизация на дома представлява контролиране на домашните устройства дистанционно или автоматично. Част от процесът по автоматизация включва и откриването на пожари. Пожарът е едно неочаквано събитие, което би могло да доведе до сериозни материални загуби и дори загуба на човешки живот.

С цел да се избегнат такива загуби нашето изследване предлага една гъвкава система за сигурност, базирана на Интернет на неща. Тя позволява в реално време да се наблюдават условията в околната среда и при поява на пожар да реагира автоматично.

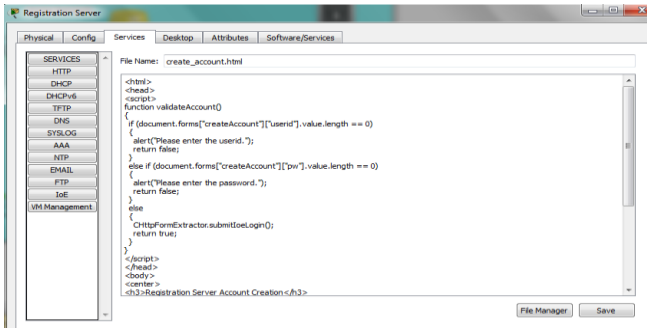
Предложената система дава възможност за получаване на информация при поява на пожар в реално време. Системата може да следи за промени в околната среда (температура), чрез сензори, свързани към контролер. Когато контролерът получи сигнал от сензора, той дава команда на изпълнителните устройства да извършат определено действие. Контролерът е свързан безжично към шлюз, който е свързан към Интернет. Данните, които контролерът изпраща се ъплоудват на сървър, като това позволява да се наблюдава състоянието на сензорите в системата през смартфон.

В предложената система се симулира модел на противопожарна система, който непрекъснато предава данни през шлюз към сървърната среда в облачна структура. Данните се събират и натрупват, анализират и предоставят на потребителя в реално време.

За провеждане на симулационните изследвания се използва симулационен софтуер Cisco Packet Tracer като се използва следният алгоритъм:

Конфигуриране на основни мрежови настройки и свързване на устройствата с MCU: включва IP адресиране на сензорите и настройки за сигурност за шлюза. Свързв-

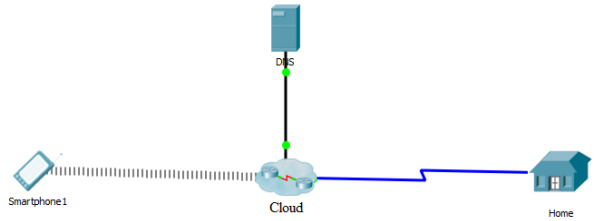
ването на устройствата към шлюза и регистрирането в регистрационния сървър е показано на фиг.2. Устройствата се свързват към шлюза и се регистрират към регистрационния сървър.



Фиг.2 Настройки на регистрационния сървър

1) Създаване на DNS регистрационен сървър.

Създаването на този сървър изисква „включване“ от раздела Услуги в DNS страницата на сървъра. За да бъде свързан към мрежата му задаваме адреси за: IP Address, Subnet Mask, Default GateWay, DNS Server



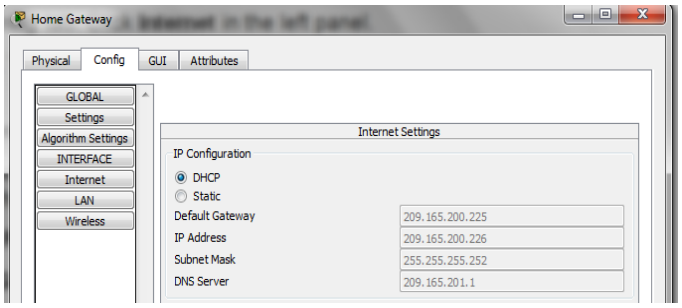
Фиг.3. Схема на мрежата

2) Свързване към Облак

Облакът се свързва с рутера, съгласно предоставените възможности на стимулационният софтуер, както е показано на фиг.3. Модемът се свързва с шлюз, като по този начин свързваме предложената система към WAN

3) Конфигуриране на шлюз за свързване на устройствата в мрежа (фиг.4.)

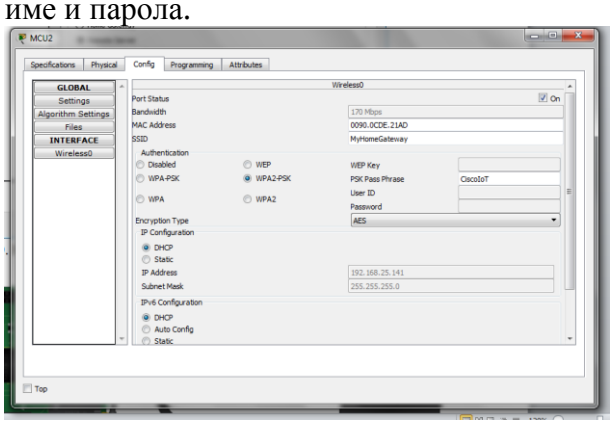
В раздела за конфигуриране на *Home Gate-way* се избира *DHCP* за *IP Configuration.*



Фиг.4. Настройки на IP-адреса на шлюза

4) Регистрация на MCU контролера

Добавя се безжичен конектор към MCU контролера и се конфигурират безжичните настройки за свързване към шлюза и съответно към мрежата. В раздел конфигурации, се конфигурира адреса на дистанционния DNS сървър със съответните име и парола.



Фиг.5**.** Конфигуриране на безжичните настройки на MCU платката.

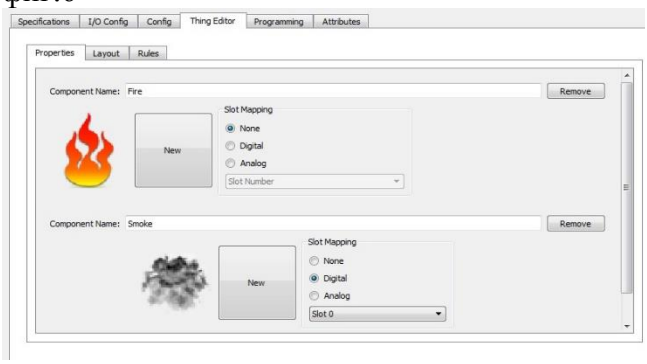
5) Създаване на Приложения/ Компоненти

Всички приложения и компоненти в Packet Tracer използват един и същ базов компонент: Неща (Thing). Уникалното при тях, е конфигурацията за Вход /Изход (I/O Config), редакторите (Thing Editor) и програмирането (Programming tabs). Възможно е също така да се вземе съществуващ компонент и да се промени, за да се създаде по-различен или изцяло нов компонент.

6) Редактор на компоненти

Редакторът определя как компонентите изглеждат на работното пространство. В раздела свойства, може да се добавят елементи, които допълват вида на компонента. Така например, в изображението по-долу е създаден пожарен симулатор. Симулаторът е програмиран да имитира пожар, с цел да се тества противопожарната система,

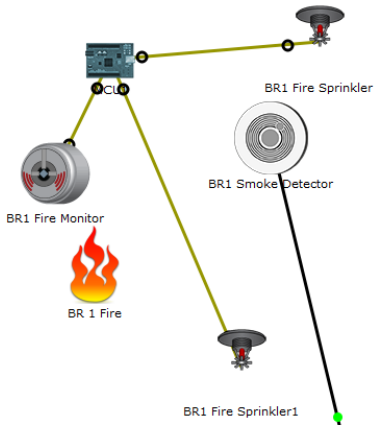
изградена от сензори за дим и пръскачки - фиг.6



Фиг.6. Редактор на компоненти

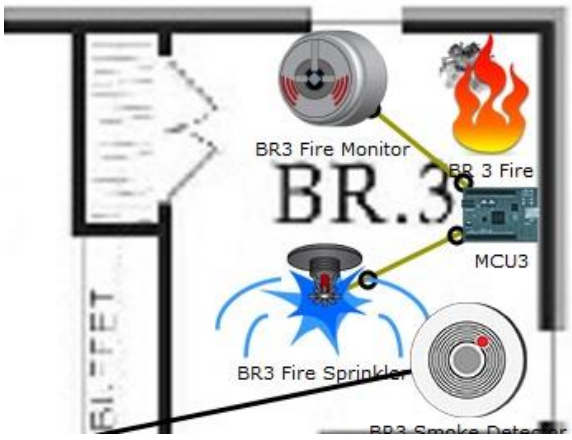
7) Изграждане на противопожарна система

Изграждането на противопожарната система се извършва чрез присъединяване на сензорите за дим и огън и разпръсквателите на вода към *MCU* контролера, съгласно фиг.7. В случай на пожар, дори и да не сме в къщи, сензорите за температура и дим ще дават информация на смартфона, за наличието на пожар.



Фиг.7. Противопожарна система

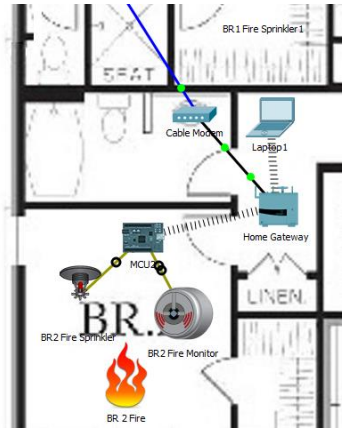
8) Симулиране на пожар -фиг. 8.



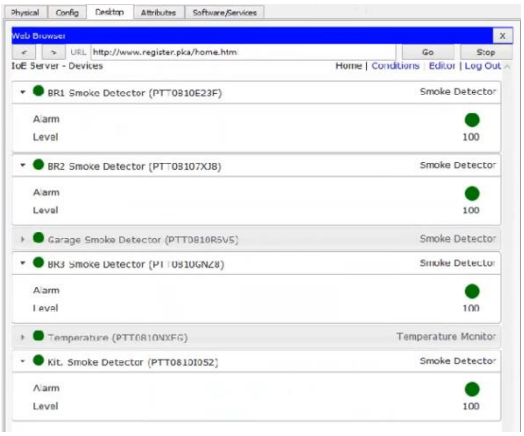
Фиг. 8. Симулиране на пожар

Чрез команда в симулатора се симулира пожар и противопожарната система задейства системата като цяло и пуска в действие разпръсквателите на вода.

9)Изпращане на съобщение за възникнал пожар- фиг.9.



Фиг.9. Свързване на контролера към шлюз и изпращане на съобщение



Фиг.10. Сензори за дим, наблюдавани през смартфона

За наблюдение работата на противопожарната система през мобилното устройство се въвежда www.register.pka в уеб браузъра. За влизане в локалния сървър на Интернет на нещата е необходима идентификация чрез потребителско име и парола. Информацията, за състоянието на димните сензори е показана на фиг. 10.

ЗАКЛЮЧЕНИЕ

Ранното предупреждение и незабавното реагиране при възникване на пожар са единствените начини да избегнем материални и нематериални щети. Следователно, най-важните цели при поява на пожар са бързото и надеждното откриване, локализиране и определяне стадия на пожара. В тази статия е симулирана работата на про-

тивопожарна система, базирана на технологията Интернет на нещата. Предложен е алгоритъм за симулиране със симулатора *Cisco Packet Tracer*, чрез който се известява за работата на противопожарната система към мобилно устройство. Симулиран е пожар за активиране работата на системата. Системата предприема автоматично необходимите действия по известяване и задействане на изпълнителни механизми за предотвратяване на възникналия пожар. Чрез изградената система потребителите могат да получат чрез *on-line* мониторинг, ранна и визуално точна информация за развитието на пожара.

Резултатите показват, че умните устройствата, които използва технологията IoT може успешно да се прилагат за предотвратяване и минимизиране на материални и нематериални щети, причинени от пожари

ЛИТЕРАТУРА

[1] F. Tsvetanov, K Tsvetanov. I.Georgieva, E. Ivanova. Modelling and Simulation of Communication Efficiency in Low-Speed Networks, Proceedings of the International conference, TELFOR'201,4 p.p. 178-182.

[2] G.Hristov and I.Georgieva, Opportunities of the Virtual Sensors, Proceeding of the Student and Doctoral Scientific Session of the Technical Faculty, 2017, ISSN 2367-9441, p.p, 89-96, http://engstudents.swu.bg/?page_id=80

[3] I. Kaur "Microcontroller Based Home Automation System With Security" IJACSA Vol. 1.

[4] Exploring C for Microcontrollers: A Hands on Approach By Jivan S. Parab.

[5] T.,Demirel H., 2009. Fire detection in video sequences using a generic color model, Fire Safety Journal, Volume 44, Issue 2, pp. 147-158

[6] Sachin Sharma, Gaurav Chitranshi, Basanta Mahato, Atul Kumar Srivastava, "Control of Home Appliances through IR interface utilizing web(GPRS) empowered Mobile Phones," International Journal of Advanced Engineering Sciences Technologies.

[7] A. R. Krishna, G. S. Bala, A. Sastry, B. B. Sarma and G. S. Alia, "Design And Implementation Of A Robotic Arm Based On Haptic Technology," International Journal of Engineering Research and Applications (IJERA), vol. 2, no. 3, pp. 3098-3103,2012.

[8] D. Giusto, A. Iera, G. Morabito, L. Atzori (Eds.), The Internet of Things, Springer, 2010. ISBN: 978-1-4419-1673-0.

[9] Sirisilla Manohar, D. Mahesh Kumar, "E-Mail Interactive Home Automation System", International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 4, Issue. 7, July 2015, pg.78 – 87.

[10] S. W. Kim, "Sensor network research and development and commercialization practices", Natural IT Industry Promotion Agency.

ИЗСЛЕДВАНЕ ЕФЕКТИВНОСТТА НА ПОСТОЯННОТОКОВ ДВИГАТЕЛ С НЕЗАВИСИМО ВЪЗБУЖДАНЕ

Васил Иванов
Университет „Проф. д-р Асен Златаров“ – Бургас

Study of the Efficiency of a Separately Excited DC Motor

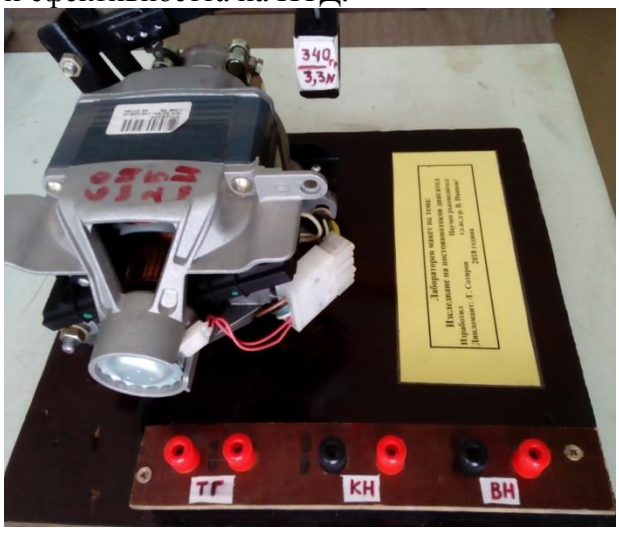
Vasil Ivanov
University Prof. Asen Zlatarov of Burgas

Abstract: A stand was made for the experimental study of a universal collector motor from an automatic washing machine. The terminals for power supply of the excitation coil (HV) and the armature coil (CN) are separated. A tachogenerator for speed control is built into the DC motor (PDD). It has a mounted mechanism with shoulder and weight. In this way, it is possible to simulate an adjustable mechanical load by means of frictional friction, under the pressure of gravity.
Keywords: DC Motor, carrying conductor, simulation.

ИЗЛОЖЕНИЕ

За експерименталното изследване на универсален колекторен двигател от автоматична пералня е изработен стенд (фиг.1). Отделени са изводите за захранване на възбудителната намотка (ВН) и котвената намотка (КН). Към постояннотоков двигател (ПТД) е вграден тахогенератор за контрол на скоростта. Той е с монтиран механизъм с рамо и тежест. По този начин е създадена възможност чрез фрикционно триене, под натиск на тежестта да се имитира регулируем механичен товар.

С използването на различен намагнитващ (възбудителен) ток и степен на натоварване са изследвани характеристиките и ефективността на ПТД.



Фиг.1. Снимка на изработения стенд

Целта на статията е чрез експериментално отчитане на електромеханичните характеристики за доказване ефективността на работа на ПТД чрез увеличаване на възбудителния ток.

Основни характеристики на постояннотоковите машини [1, 2, 4, 5, 6, 7]**.**

● Основните механични величини при ПТД са: **въртящ момент M** и **скорост на въртене ω** с техните изменения.

Механичната характеристика ω=f(M) на постояннотоков двигател (ПТД) може да бъде получено от уравнението за равновесието на е.д.н. във веригата на котвата при установен режим на работа.

▪ **Механичният въртящият момент M** на цилиндричен ротор е:

$$M = Fr \quad (1)$$

или

$$M = F\frac{D}{2}, \quad (2)$$

където

F е периферна тангенциална сила създадена от въртенето на ротора,

r - радиус на ротора,

D – диаметъра на ротора.

▪ **Електромагнитната сила F** се дължи на взаимодействието на тока в ротора с магнитното поле на статора:

$$F=B.l.I \quad (3)$$

където **B** е **магнитната индукция** на полето, която действа **на проводник с дължина l**, когато през него **протича ток I**, при **$\alpha = 90^0$** (**α** - ъгълът между **тока и магнитните линии**, определен от техните посоки).

▪ **Приложеното напрежение U** към изводите на ПТД се уравновесява от индуктираното в котвената намотка **е.д.н. E** и от **загубата на напрежение** във веригата на котвата **$I_a.R_a>0$**:

$$U - E - I_a R_a = 0 \quad (4)$$

$$E = \frac{pN}{60a}\Phi n = c_E \Phi n \quad (5)$$

или

$$E = \frac{pN}{2\pi a}\Phi\omega = c'_E \Phi\omega, \quad (6)$$

където:

n - скорост на въртене на котвата на двигателя, об/min;

ω - ъглова скорост на въртене на котвата, rad/s;

Φ - магнитен поток на двигателя, Wb;

p - брой на чифтовете полюси на двигателя;

a - брой на паралелните клонове на котвената намотка;

N - брой на активните проводници на котвената намотка;

$c_E = \frac{pN}{60a}$ или $c'_E = \frac{pN}{2\pi a}$ са постоянни коефициенти, които зависят от конструктивните параметри на двигателя.

Ако заместим стойността на е. д. н. от (5) и (6) в (4), получаваме израза

$$U = c_E \Phi n + I_a R_a \quad (7)$$

или

$$U = c'_E \Phi\omega + I_a R_a \quad (8)$$

Като решим (7) и (8) относно **скоростта на двигателя**, намираме уравнението на механичната характеристика

$$n = \frac{U - I_a R_a}{c_E \Phi} = \frac{U}{c_E \Phi} - \frac{R_a}{c_E \Phi} I_a \quad (9)$$

$$\omega = \frac{U - I_a R_a}{c'_E \Phi} = \frac{U}{c'_E \Phi} - \frac{R_a}{c'_E \Phi} I_a \quad (10)$$

▪ **Въртящият момент** на ПТД, изразен в N.m, се дава с формулата:

$$M = c_M \Phi I_a \quad (11)$$

където

$$c_M = \frac{pN}{2\pi 9.81a} \quad (12)$$

или

$$c'_M = \frac{pN}{2\pi a} \quad (13)$$

е константа на двигателя.

Тогава за е. д. н. E и момента на двигателя M можем да запишем

$$E = c\Phi\omega \quad (14)$$

$$M = c\Phi \mathrm{I}_a \quad (15)$$

Изразът (15) показва, че въртящият момент е пропорционален на магнитния поток, който се създава от магнитен поток и тока в котвата.

▪ Оборотите на електродвигателят се определят от индуцираното е.д.н. в котвата чрез (5) и (6)

$$n = \frac{E}{c_E \Phi} \quad (16)$$

или

$$\omega = \frac{E}{c'_E \Phi}, \quad (17)$$

където:

c_E и c'_E – константи (от 5 и 6);

E – е.д.н., което може да се изрази чрез приложеното напрежение U и пада на напрежението в котвата, като се използва равенството (4)

$$E = U - I_a R_a \quad (18)$$

● Механична мощност

Както е известно от механиката, механичната мощност на едно въртящо се тяло е**:**

$$P_m = M\omega = \frac{2\pi.n}{60}M \qquad (19)$$

От изрази (19), (11) и (5) за **механичната мощност** се получава

$$P_m = \frac{2\pi.n}{60}M = c_E \Phi n I_a = EI_a \qquad (20)$$

тъй като множителят c_E е същият, както в израза (5).

Израз 20 показва връзката между електрическа и механична мощност.

Извод: Получената механична мощност се определя от електродвижещото напрежение E, а не от приложеното напрежение U. То определя електрическата мощност, която двигателят ще консумира от мрежата.

Само частта $\mathbf{E.I_a}$ се превръща в механична мощност. Останалата част отива за покриване на различните загуби.

● Ефективност на ПТ двигател

За да се изчисли ефективността на двигателя, механичната изходна мощност се разделя на **електрическата входна мощност.**

$$\eta = \frac{P_m}{P_e}, \qquad (21)$$

където

η ефективност на преобразуване на енергията (коефициент на полезно действи) к.п.д.,

P_e електрическа входна мощност,

P_m механична изходна мощност.

P_e = UI,

където

U входно напрежение,

I входен ток,

P_m = M.ω = 0,1047.M.n,

M изходен въртящ момент,

ω изходна ъглова скорост.

Често **η** се представя в проценти и представлява **к.п.д.**

● Видове двигатели в зависимост от начина на присъединяването на възбудителната намотка към котвата:

▪ **Двигатели с паралелно или независимо възбуждане**, при които възбудителната намотка се включва паралелно на котвата или към независим източник за постоянен ток. При тях **магнитният поток не зависи от товара (Ф=const) и следователно ω=const;** механичните им характеристики са **твърди**.

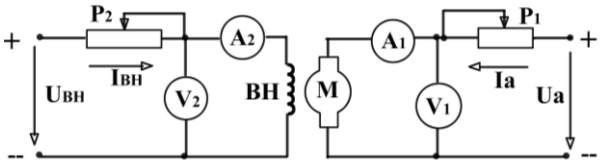
▪ **Двигатели с последователно възбуждане**, при които възбудителната намотка е включена последователно на котвата. **Магнитният поток зависи от товара $\Phi=f(I_a)$** и механичните им характеристики са меки.

▪ **Двигатели със смесено възбуждане**, при които има две възбудителни намотки - паралелна и последователна.

МЕТОД НА ИЗСЛЕДВАНЕ

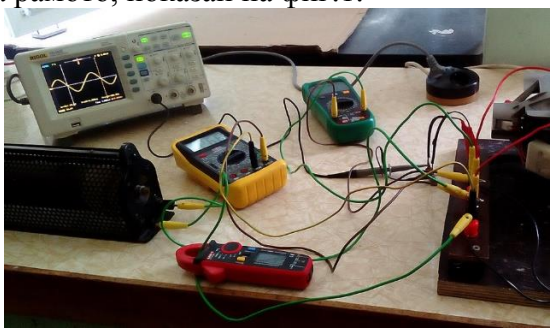
За провеждане на експерименталните изследвания е изпълнена схемата от фиг.2 [3].

Схема на опитната постановка



Фиг.2. Принципна схема на опитната постановка – ПТД в схема на възбудителна намотка ВН с независимо захранване.

Постояннотоковият електродвигател е с разделени електрически вериги за захранване на възбудителната намотка (ВН) и котвената намотка (КН). Монтираният към вала на двигателя фрикционен механизъм създава изкуствен регулируем съпротивителен товар според закрепването му по оста на рамото, показан на фиг.1.



Фиг.3. Снимка от лабораторния стенд в процес на отчитане оборотите чрез тахогенератор (показано на осцилоскопа)

● Номинални технически данни на двигателя:

Номинално напрежение U = 230V;

Номинален ток $I_a = I_в = 1{,}65A$;

Номинална мощност P = 360W;
Електрически параметри на котвената намотка: R_a=2.0 Ω, L_a=23mH
Електрически параметри на възбудителна намотка: $R_{вн}$=2Ω, $L_{в}$н=57mH

● Изчислителна част: ПТД с независима възбудителна намотка.

▪Измерване на ъгловата скорост (честота)

$\omega=(\pi/30).n=0,1047.n,\ s^{-1}$ (24)

На опитната постановка измерваме чрез **осцилоскоп** ъгловата скорост ω=2.π.f, като данните се отчитат от вградения тахогенератор на вала на ПТД. Използвайки **ω** измерения параметър отчитаме оборотите **n**:

$n = (30/\pi).\ \omega=9.55.\ \omega$ (25)

▪ Изчислената механична мощност получена на вала на ПТД $P_м$:

$P_м=0,1047.M.n=M.\omega$, [N.m] (26)

▪Изчислената активна мощност **Ра изч.**се изразява чрез:

$P_{а\ изч.} = U_a.\ I_a$ [W], (27)

където,

U_a [V] - измереното напрежение на клемите на котвената намотка,

I_a [A] – измерен ток във веригата на котвената намотка.

▪**В експеримента е отчетено** активното съпротивление на намотките, което определя загубите на мощност изразени чрез **$P_{заг.}$** Това се отразява чрез отделяне на топлина в ПТД.

$P_{заг.}= I_a^2.\ R_a$ [W] (28)

▪**Изходна активна мощност $P_{а\ изх.}$,** която реално се използва за работа на ПТД се получава като от **$P_{а\ изч.}$** се извадят загубите **$P_{заг.}$** в котвената намотка:

$P_{а\ изх.} = P_{а\ изч.} - P_{заг.}$ [W] (29)

▪Въртящият момент на вала на ПТД:

$M_{върт.}=9,55.\ P_{а\ изх.}/n= P_{а\ изх./}\omega$ [N.m], (30)

▪ **Коефициентът на полезно действие η** определя ефективната работа на ПТД:

$\eta = 100 \cdot P_{а\ изх.}/ P_{а\ изч}$ [%] (31)

ЕКСПЕРИМЕНТАЛНИ РЕЗУЛТАТИ

По схемата от фиг.2 се свързват захранващите източници и измервателната апаратура към лабораторния стенд.

● Задават се входните параметри определящи режимите на работа:

▪ **Режим на работа на ПТД при празен ход** със зададен възбудителен ток **$I_{вн1}$=1A.**

Примерни изчисления за първото измерване:

$P_{а\ изч.} = U_a.\ I_a = 6,50.\ 0,38= 2,47$[W], активна изчислена мощност

$P_{заг.} = Ia^2.\ R_a=0,38^2.2= 0,29$ [W], мощност-загуби

$P_{а\ изх.} = Pa_{\ изч.} - P_{заг.} = 2,47 - 0,29 = 2,18$ [W], реална изходна активна мощност

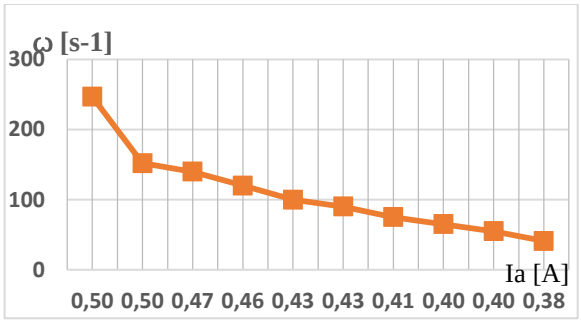
$M_{върт.} = P_{а\ изх.\ /}\ \omega = 2,18\ /41 = 0,053$[N.m], въртящ момент на вала на ПТД

η (к.п.д) .= $100.P_{а\ изх\ .}/\ P_{а\ изч} = 100.\ 2,18\ /\ 2,47 = 88$ [%].

▪ Графично представяне механичните характеристики при работа на ПТД с независимо възбуждане без товар (празен ход) по данните от табл.1.

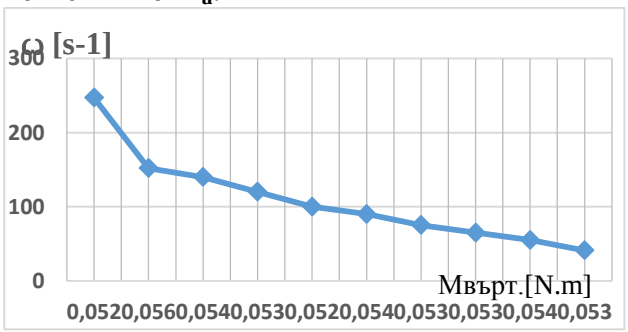
Таблица 1 данни при **$I_{вн1}$**

измерени			**изчислени**				
Ua	**Ia**	**ω**	**Ра изч.**	**Р заг.**	**Ра изх.**	**$M_{върт.}$**	**к.п.д.**
V	**A**	**s^{-1}**	**W**	**W**	**W**	**N.m**	**%**
6,50	0,38	41	2,47	0,29	2,18	0,053	88
8,23	0,40	55	3,29	0,32	2,97	0,054	90
9,34	0,40	65	3,74	0,32	3,42	0,053	91
10,50	0,41	75	4,31	0,34	3,97	0,053	92
12,10	0,43	90	5,20	0,37	4,83	0,054	93
12,90	0,43	100	5,55	0,37	5,18	0,052	93
14,80	0,46	120	6,81	0,42	6,38	0,053	94
16,90	0,47	140	7,94	0,44	7,50	0,054	94
18,00	0,50	152	9,00	0,50	8,50	0,056	94
26,80	0,50	247	13,40	0,50	12,90	0,052	96



Фиг.4. Диаграма от ъглова скорост **ω** и котвен ток $\mathbf{I_a}$ при фиксиран $\mathbf{I_{вн1}}$.

Извод: От фиг.4. се наблюдава увеличаване на ъгловата скорост **ω** при нарастване на котвения ток $\mathbf{I_a}$.



Фиг.5. Диаграма на ъглова скорост и въртящия момент

Извод: От фиг.5 е отчетен стабилен въртящи момент $\mathbf{M_{върт.}}$ при зададения среден възбудителен ток.

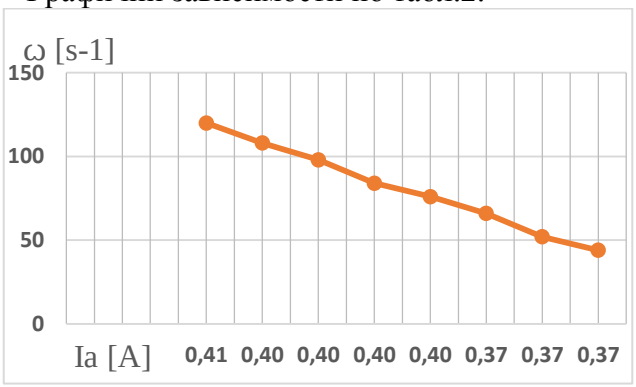
▪ Режим на празен ход с номинален фиксиран $\mathbf{I_{вн2}}$ = 1,5A

Таблица 2. данни при номинален $\mathbf{I_{вн2}}$

измерени			**изчислени**				
Ua	**Ia**	**ω**	**Pa** изч.▪	**P** заг.	**Pa** изх▪	**$M_{върт.}$**	**к.п.д.**
V	**A**	**s^{-1}**	**W**	**W**	**W**	**N.m**	**%**
8,70	0,37	44	3,22	0,27	2,95	0,067	91
10,10	0,37	52	3,74	0,27	3,46	0,067	93
12,00	0,37	66	4,44	0,27	4,17	0,063	94
13,30	0,40	76	5,32	0,32	5,00	0,066	94
14,20	0,40	84	5,68	0,32	5,36	0,064	94
16,20	0,40	98	6,48	0,32	6,16	0,063	95
17,70	0,40	108	7,08	0,32	6,76	0,063	95
19,30	0,41	120	7,91	0,34	7,58	0,063	96

В табл. 2 се наблюдава нарастване на въртящия момент $\mathbf{M_{върт.}}$ при увеличение на възбудителния ток $I_{вн}$, което се определя от нарастващия магнитен поток Ф. **Следователно** е добавена допълнителна магнитна енергия за преодоляване на съпротивителния момент от механични загуби в ПТД.

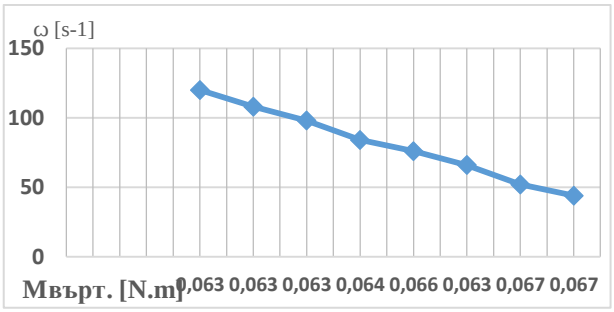
● Графични зависимости по табл.2.



Фиг.6. Изменение на ъглова скорост **ω** и котвен ток **Ia** при постоянен $\mathbf{I_{вн2}}$.

Анализ на резултатите от експеримента:

При работа на ПТД без товар показва едни и същи механични характеристики отчетени от фиг.5 и фиг.6. Регистрирано е стабилизиране на въртящия момент при повишаване на възбудителния ток до номинални стойности.



Фиг.7. Диаграмата отчита при намаляване на ъгловата скорост **ω** нараства на въртящия момент $\mathbf{M_{върт.}}$

▪ Режим на работа с включен товар G=3,24N и фиксиран възбудителен ток $I_{вн3}$=1,6A

Механичният товар се определя от израза в механиката при натоварване на въртящ обект (ротора на ПТД) с механична сила:

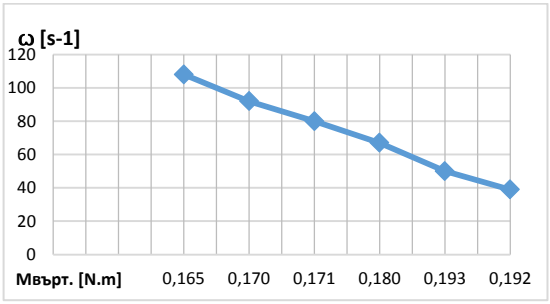
Мв= F.r [N.m]

F= g.m = 9,81. 0,33= 3,24 N (33)

Таблица 3 при товар G=3,24N и възбудителен ток $I_{вн3}$=1,6A

измерени			**изчислени**				
Ua	**Ia**	**ω**	**Pa** изч.▪	**P** заг.	**Pa** изх▪	**$M_{върт.}$**	**к.п.д.**
V	**A**	**s^{-1}**	**W**	**W**	**W**	**N.m**	**%**
10,70	0,83	39	8,88	1,38	7,50	0,192	84
12,50	0,90	50	11,25	1,62	9,63	0,193	86
14,50	0,96	67	13,92	1,84	12,08	0,180	87
16,20	0,96	80	15,55	1,84	13,71	0,171	88
18,20	0,96	92	17,47	1,84	15,63	0,170	89
20,50	0,96	108	19,68	1,84	17,84	0,165	91

По табл.3. се построява механичната характеристика на ПТД с независимо възбуждане.

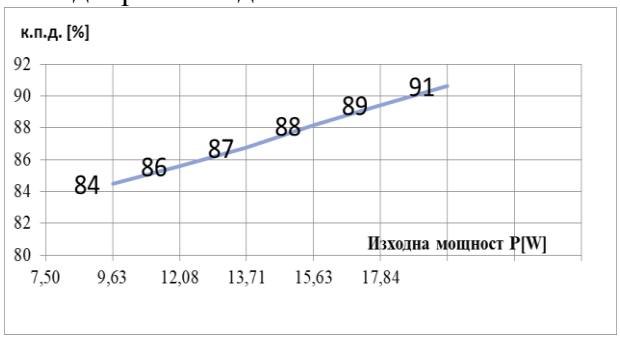


Фиг.7. Механична характеристика при включен товар на вала на котвата.

Графично представената на фиг.7 механична характеристика е с **повишен въртящ момент** в резултат от натоварването с механична тежест на рамото от стенда. **Извод:** Ъгловата скорост **ω** намалява пропорционално на нарастващия въртящ момент по израза:

$Мв=P_{изх}./ \omega$ [N.m] (3.11)

При повишаване на мощността $P_{изх}$ се подобрява к.п.д.



Фиг.8. Диаграма на к.п.д. от изходна мощност $P_{изх}$

АНАЛИЗ НА РЕЗУЛТАТИТЕ

От получените резултати мога да направя следните изводи :

▪ Повишаването на котвеното напрежение води до повишаване на оборотите в режим на празен ход. Повишаване на възбудителния ток до номинална стойност стабилизира въртящия момент, повишава се мощността и като цяло се подобрява ефективността на работа на двигателя.

▪ Графичните резултати от фиг.8 показват, че при повишаване на изходната мощност се подобрява ефективната работа на ПТД като к.п.д. достига $\eta = 91\%$.

▪ От експериментите чрез лабораторния макет и построените диаграми се доказва, че най-ефективна е работата на ПТД при номинално натоварване и номинален възбудителен ток.

ЛИТЕРАТУРА

[1] Маринов, „Електротехника и електроника“, Варна,2012

[2] „Електромеханични устройства и системи“, В. Яков, В. Стефанов, Русе, 2013г.

[3] Сотиров, дипломна работа „Изследване на постояннотоков двигател“, Бургас, 2018

[4] Rachev S., L. Dimitrov, D. Koeva. FEATURES UPON CONVERTING A CONVENTIONAL CAR INTO AN ELECTRIC CAR, TRANS MOTAUTO WORLD, Issue 2/2018, pp. 52-56. ISSN 2367-8399.

[5] Рачев С., М. Рачева, Д. Коева, Л. Димитров. Текущо състояние и съвременни тенденции в развитието на електрическите превозни средства – Част I: общ преглед на разработването, Енергиен форум 2018, 26-29 юни 2018, к.к. „Св. Св. Константин и Елена”, Сборник, 285-300. ISSN 2367-6728.

[6] Рачев С., М. Рачева, Д. Коева, Л. Димитров. Текущо състояние и съвременни тенденции в развитието на електрическите превозни средства – част II: специфики в България, Енергиен форум 2018, 26-29 юни 2018, к.к. „Св. Св. Константин и Елена”, Сборник, 301-314. ISSN 2367-6728.

[7] Димитров Л., Рачев С., Ст. Филчев. Изследване на електрозадвижването на компактно електрическо транспортно средство, Четвърта научна конференция с международно участие „Комуникации, електрообзавеждане и информатика в транспорта – КЕИТ 2018”, 28-30 септември.2018, гр. Банско.

Адрес за контакти:
гл. ас д-р Васил Борисов Иванов
Университет „Проф. д-р Асен Златаров“, Бургас
e-mail: vasil_bi@abv.bg

ИЗСЛЕДВАНЕ КАПАЦИТЕТА НА АВТОНОМНА ФОТОВОЛТАИЧНА СИСТЕМА, ЗАХРАНВАЩА IP ВИДЕОКАМЕРА В НЕПРЕКЪСВАЕМ РАБОТЕН РЕЖИМ

Васил Иванов, Христо Иванов
Университет „Проф. д-р Асен Златаров“ – Бургас
Областна администрация- Бургас

Investigation of the Capacity of an Autonomous Photovoltaic System Supplying an IP Video Camera in Continuous Operation

Vasil Ivanov, Hristo Ivanov
University Prof. Asen Zlatarov of Burgas
Областна администрация- Бургас

Abstract: The capacity of an autonomous photovoltaic power supply system when operating in continuous mode with a minimum load (IP camera) has been determined. The change of the parameters of the accumulator batteries during the period of charging during the day and discharging and at night was observed. The data from the controller is recorded at certain periods of time. The required installation capacity of PVS has been determined
to operate in continuous mode.
Keywords: photovoltaic system, IP camera, battery, operation mode.

ВЪВЕДЕНИЕ

Фотоволтаичната захранваща система е с автономен работен режим, защото е независима от обществената енергийна система. Обекти на изследването са източниците на електрическа енергия: соларните панели (СП) и акумулаторните батерии (АБ) [1, 5, 6].

1. Автономна фотоволтаична захранваща система (АФЗС)

Фиг. 1. Обща структура на автономна фотоволтаична захранваща система (АФЗС) за малка мощност състояща се от соларни панели, акумулаторни батерии и постояннотоков товар - уеб камера

▪ **Соларни панели (СП). Технически данни** [2]**:**

Соларен панел (СП) модел: Luxor Solo Line 100 - LX-100M 125-36”: изходно напрежение при STC: U_{MPP} = 18.70V; изходен ток при STC: I_{MPP} = 5.39A; напрежение на празен ход при STC: U_{OC} = 21.6V; - ток на късо съединение при STC: I_{SC} = 5.87A; пикова изходна мощност при STC: P= 100Wp; при свързани в паралел 2 броя СП обща пикова мощност P= 200Wp;

Фиг. 2. Изглед на монтираните СП от АФЗС

▪ **Акумулаторна батерия (АБ) модел:** SW 12240 (12V 24Ah)
- номинален капацитет $C_{АБ\,1}$ = 24Ah;
- номинално напрежение $U_{АБ}$ = 12V;

- номинален капацитет от две батерии в паралел $C_{АБ\ 1,2}$= 48Ah;

Фиг.3 Снимка на едната акумулаторна батерия от АФЗС.

▪ **Микроконтролер и дисплей за управление модел** [3]: BlueSolar PWM- Pro Charge Controler 12/24V – 10A; BlueSolar PWM-Pro Charge Controllers
- Battery Voltage: 12/24V with automatic system voltage detection;
- Rated charge current: 10A;
- Maximum solar voltage: 28V / 55V; for 12V use 36 cell Solar panels;
- Protection: Battery reverse polarity (fuse); Output short circuit;

Фиг.4. Снимка на заряден контролер и на дисплея за визуализиране на данните

▪ **IP видеокамера модел** [4]: (AZISHN Super Mini Wireless, Security, 720P IP Camera, Wifi, Night Vision, Two Way Audio) Захранващи параметри на IP видеокамера: U_{in} = +5V; I_{in}= 0,15A;

Фиг. 5. Снимка на IP видеокамерата използвана в опитната постановка.

IP видеокамерата от фиг.5 е съвременен модел с дистанционна комуникация на изображение и звук и дистанционно управление посоката. Има възможност за нощно наблюдение чрез вградените инфрачервени диоди, които се включват автоматично при затъмняване. Основната и функция е наблюдение и запис на мобилен телефон или компютър с предварително инсталирано приложение.

▪ **Стабилизиращ преобразувател DC/DC от понижаващ вид:** U_{out}=+12/+5V; I_{out}= 2A;

2. Определяне на основните характеристики от АФЗС.

А. Изходни данни за източниците и консуматорите:

Соларни панели СП с обща изходна мощност: $P_{СП}$ = 200Wp; $U_{СП}$ = 18.70V; $I_{СП}$ = 10,78A; $\eta_{СП}$ =15,4%;

Акумулаторни батерии с общ номинални капацитет и напрежение:

$C_{АБ}$= 48 Ah; $U_{АБ}$=12V; $\square_{АБ}$=85%; $W_{АБ} = C_{АБ} . U_{АБ} . W_{АБ}$ = 490 W.h

Преобразувател и стабилизатор номинални параметри:

U_{in} = 12V, I_{in}=2,5A

U_{out} = 5V, I_{out}=1,5A

IP видеокамера като консуматор:

U_{ip} = 5V, I_{ip}=0,4A

Б. Основни формули за изчисляване параметрите от АФВС

Капацитетът на акумулаторната батерия АБ **($C_{АБ}$)** е необходимото количество електричество захранващо постояннотоков консуматор (крушка, мотор, реле, стабилизатор и др.) за определен период от време. Той се означава в амперчасове [Ah] и се разделя на: теоретичен капацитет, действителен капацитет и номинален капацитет.

Капацитетът $\mathbf{C_{АБ}}$ в АБ не е фиксирана стойност. Той е пряко свързан, температурата, изходното напрежение $\mathbf{U_{АБ}}$, капацитета на разреждане $\mathbf{C_{АБ\ разр.}}$ (или времето за разреждане $\mathbf{t_{\ разр.}}$) и разрядния ток $\mathbf{I_{\ разр.}}$ Капацитетът $\mathbf{C_{АБ}}$ на АБ намалява с понижаване на температура $\mathbf{t^oC}$, защото тя оказва сериозно влияние върху вискозитета и устойчивостта на електролита.

През 1898 Peukert предлага уравнение за определяне на разряден капацитет на АБ. Колкото е **по-голям разрядният ток $\mathbf{I_{разр}}$**, толкова по-малка е дълбочината на действие, толкова по-малко е използването на активния материал, **толкова по-малък е капацитетът на акумулатора**.

Законът на Peukert свързва $\mathbf{C_{АБ}}$ със скоростта на разреждане (разряден ток $\mathbf{I_{разр}}$ и време на разреждане $\mathbf{t_{разр.}}$):

$$\mathbf{C_{АБ\ разр.} = I^{k}_{разр.} \bullet t_{разр.}\ [Ah];} \quad (1)$$

където:

$\mathbf{C_{АБ\ разр.}}$ - капацитета на АБ при скоростта на разреждане 1А за 1час;

$\mathbf{I^{k}_{разр.}}$ - разряден ток в ампери;

t - време на разреждане в часове;

к – коефициент на Peukert в обхвата на 1,1 до 1,3;

● Енергия при заредена АБ:

$$\mathbf{W_{АБ\ зар.} = C_{АБ\ зар.} \bullet U_{АБ\ зар.}\ [W.h];} \quad (2)$$

● Капацитет при заредена АБ:

$$\mathbf{C_{АБ\ зар.} = I_{зар.}.t_{зар.}\ [Ah];} \quad (3)$$

● Енергията при разредена АБ:

$$\mathbf{W_{АБ\ разр.} = C_{АБ\ разр.} \bullet U_{АБ\ разр.}\ [W.h];} \quad (4)$$

● Капацитет на разреждане на АБ:

$$\mathbf{C_{АБ\ разр.} = I_{разр.} \bullet t_{разр.}\ [Ah];} \quad (5)$$

КПД на АБ:

$$\boldsymbol{\eta} = \frac{\boldsymbol{W}_{\text{АБ разр}}}{\boldsymbol{W}_{\text{АБ зар}}} = \frac{\text{C}_{\text{АБ разр}}\ .\mathbf{U}_{\text{АБ разр}}.}{\text{CАБ зар}\ .\mathbf{U}_{\text{АБзар.}}}. \quad (6)$$

Процесът на зареждане/разреждане за АБ е цикличен. Съществуват определен брой цикли, които са свързани с основните експлоатационни характеристики и гаранционните условия за конкретната АБ. За оловно-киселинните АБ броя дълбоки цикли е около 300.

2 Определяне на параметрите за изчисляване: сумарен капацитет и енергия при зададени номинални характеристики на източниците АБ и СП.

Оловни акумулаторни батерии АБ:

Сумарният капацитет $\mathbf{C_{АБ}}$ за оловно киселинните АБ от АФЗС е отбелязан на корпуса и е:

$\mathbf{C_{АБ}}$ = 48[Ah].

Номинален капацитет на АБ включва и к.п.д. = 0,85:

$$\mathbf{C_{АБ\ зар.} = I_{зар.} \bullet t_{зар.} \bullet 0{,}85}\ [\text{Ah}] \quad (7)$$

▪ Реалният капацитет с включен к.п.д. за оловна АБ на изследваната АФЗС при средна стойност на разпределение за 1час е:

$\mathbf{C_{АБ\ зар.} = C_{АБ} \bullet 0{,}85 =}$ 48.0,85 = 40,8 [Ah];

▪ Електрическата енергия съхранен в АБ с възможност за отдаване в рамките на1час върху консуматор с активна мощност 530 W е:

$$\mathbf{W_{АБ\ зар.} =}\ 40{,}8.13 = 530\ [\text{W.h}] \quad (8)$$

● Генерираната и равномерно отдадена през нощта мощност $\mathbf{P_{АБ\ средно}}$ от АБ, определя средната ѝ стойност на разреждане. Отдадената енергия в консуматора е нощно време, защото тогава СП не работят.

$$\mathbf{P_{АБ\ средно} = W_{АБ\ зар.}\ /\ t_{нощ}}\ [\text{W}]; \quad (9)$$

$\mathbf{P_{АБ\ средно}}$= 530/17 = 31,2 [W] .

Соларни панели СП в АФЗС.

▪За България средните стойности на периодите на редуване ден и нощ са с продължителност, което е важно за определяне периодите от време за генериране на електрическа мощност от СП:

$\mathbf{t_{ден}}$**= 7часа** и $\mathbf{t_{нощ}}$ **= 17часа,** като **T**= $t_{ден}$ + $t_{нощ}$ = 24часа (период на денонощието)

▪Теоретичната максимална мощност $\mathbf{P_{СПмаксSTC}}$, генерирана от два еднотипни СП, свързани в паралел при стандартни условия на тестване $P_{СП\ STC}$ е отбелязана на корпуса на панелите и за АФЗС е:

$\mathbf{P_{СП\ макс\ STC}}$ = 200Wp ;

Тя се генерира при пикови стойности на излъчваната от слънцето оптична плътност на мощността, когато то е в зенита си на попадане върху земната повърхност.

Коефициент на полезно действие на СП:

$\eta_{СП\ STC} = P_{СП\ STC} / G_{СПмакс} \bullet S_{СП} = 200/1000.1,3 = 0,154;$

където,

$G_{СПмакс}$ - максимален интензитет на слънчевата радиация;

$S_{СП} = 1,3$ м^2 -обща площ на инсталираните СП;

Генерираната реална номинална мощност $P_{СП\ STC\ ном}$ за два еднотипни СП, свързани в паралел включва и к.п.д. на СП:

$P_{СП\ STC\ ном} = P_{СП\ STC} \bullet \eta_{СП\ STC} = 200.0,154 = 30,8$ Wp;

Номиналният капаците $C_{СП\ STC}$ генериран от СП се определя от максималния ток $I_{СП\ STC}$ при стандартни условия на тестване STC и средно време на ден $t_{ден}$, определено за нашия регион, както и от к.п.д. :

$C_{СП\ STC} = I_{СП\ STC} \bullet \eta_{СП\ STC} \bullet t_{ден} = 10,78.0,154.7 = 11,62$ [Ah]; (10)

Средната стойност на енергия $W_{СП\ STC}$ от СП се изчислява за средния период от време на слънцегреене $t_{ден}$;

$W_{СП\ STC} = C_{СП\ STC} \bullet U_{СП\ STC.} = 11,62 \bullet 18,7 = 217,3$ [W.h]; (11)

Баланс на мощностите между АФЗС и консуматора.

Условието АФЗС да работи в непрекъсваем режим е свързано с баланса между инсталиран (заряден) $C_{АБ\ зар.}$ и консумиран (разряден) $C_{АБ\ разр.}$ капацитети на АБ като:

$C_{АБ\ зар.} \geq C_{АБ\ разр.}$; (12)

Така ще бъде гарантиран непрекъсваем енергиен капацитет за продължителен период от време в процеса на захранване на консуматорите.

При оловно-киселинните АБ $C_{АБ\ разр.}$ се реализира 85% от $C_{АБ\ зар.}$ в процеса на зареждане. Това определя коефициент на полезно действие к.п.д. при разряд:

$\eta_{UАБ\ разр.} = 0,85 \bullet \eta_{UАБ\ зар.}$ (13)

Принципът на работа на АФЗС е в използването на СП за захранване на консуматорите и зареждане на АБ през светлата част от деня $t_{ден}$ и изключването им от системата през нощта. Тогава се използва натрупаната енергия от СП в АБ, която захранва консуматорите.

Процесът на разреждане през нощта на АБ е свързан с отдаване на заряда през консуматора-IP камера.

Изчисляването на $C_{АБ\ разр.\ нощ}$ е свързан с използване на к.п.д. и капацитет на АБ при разреждането ѝ за средната стойност на консумирания ток от IP камера през ноща $t_{нощ}$:

$$C_{АБ\ разр.\ нощ} = \frac{I_{разр.средно} . t_{нощ}}{\eta_{UАБ\ разр.}}, Ah \quad (14)$$

Средната стойност на необходимия **капацитет** $C_{консум.\ ден}$ на АФЗС за захранване на IP камера през деня се определя от:

$C_{консум.\ ден} = I_{разр.средно} \bullet t_{ден}$, Ah; (15)

Средната стойност на необходимия **капацитет** $C_{консум.\ нощ}$ на АФЗС за захранване на IP камера през ноща се определя от:

$C_{консум.\ нощ} = I_{разр.средно} \bullet t_{нощ}$, Ah; (16)

Средната стойност на необходимия общ капацитет за денонощие:

$\Sigma C_{консум.} = C_{консум.\ нощ} + C_{консум.\ ден}$, Ah (17)

Средната стойност на необходимата обща **енергия за денонощие**:

$W_{консум} = P_T . T, Wh$ (18)

Максимална консумирана мощност от АБ при зареждането ѝ:

$P_{ТАБ\ макс} = U_{ТАБ\ макс} . I_{ТАБ\ макс}, W$

Средната стойност на $P_{ТАБ}$ за $t_{нощ}$

$P_{ТАБ\ средно} = P_{ТАБ\ макс} . t_{нощ} / T, W$

Сумарна консумирана мощност:

$\Sigma P_{ТАБ} = P_T + P_{ТАБ\ средно}, W$

Необходимата площ на СП за да работи системата с този консуматор за продължителен период от време е:

$S_{СП} = \Sigma P_{ТАБ} / G_{СПсредно}$, m^2 (19)

Средната стойност на интензитета на слънчевата радиация:

$G_{СПсредно} = G_{СПмакс} . t_{нощ} / T, W / m^2$ (20)

$G_{СПсредно} = G_{STC} . \eta_{СП} . t_{нощ} / T, W / m^2$

след заместване в $S_{СП}$ получаваме изчислената теоретична площ на СП:

$S_{СП} = \Sigma P_{ТАБ} / G_{СПсредно}$, m^2

Необходимата **средната стойност на сумарния капацитет** от СП **през деня** включва капацитета $C_{АБ\ разр.нощ}$ и $C_{консум.ден}$:

$\Sigma C_{СП} = C_{АБ\ разр.\ нощ} + C_{консум.\ ден}$, Ah; (21)

От тук се определя и средната стойност на изходния ток $I_{СП.средно}$:

$$I_{СП.средно} = \frac{\sum C_{СП}}{t_{ден}}, A \quad (22)$$

Соларните панели СП са неподвижно монтирани като средната стойност на изходния ток $I_{СП.средно}$ се изменя по си-

нусоидален закон. Интегрираме данните от мониторинга при слънчев ден:

$$\mathbf{I_{СП.средно}} = \frac{\mathbf{2}}{\boldsymbol{\pi}} \bullet \mathbf{I_{СП.макс.}} = 0{,}64 \bullet I_{СП.макс} \quad (23)$$

Определяне на максималната стойност на тока $\mathbf{I_{СП.макс}}$ от СП, като се пренебрегват загубите от разликите в напреженията $U_{СП} \leq U_{АБ}$ след заместване в израза за средната стойнот на тока:

$$\mathbf{I_{СП.макс}} = \mathbf{I_{СП.средно}} / 0{,}64, \text{ A} \quad (24)$$

ЕКСПЕРИМЕНТАЛНА ЧАСТ

Методика на измерване и изследване капацитета на автономна фотоволтаична система с минимален товар [7]. Изследването се базира на анализ на резултатите от измерване капацитета на АБ и СП в зависимост от продължителността на слънце греенето. През периода на измерване ще към АФВС ще бъде включена да се захранва IP камера в режим на непрекъснато наблюдение.

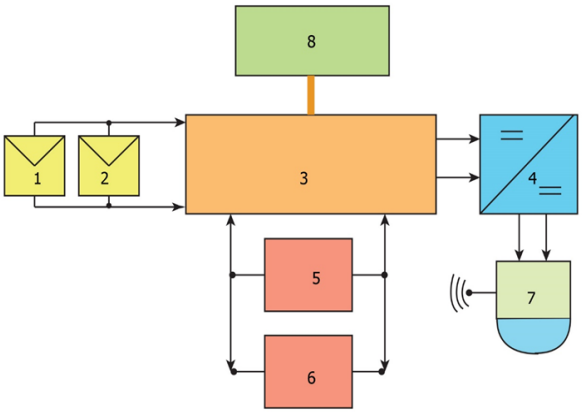
Изчислява се капацитета при работа на IP камера в непрекъсваем режим. Ще бъде направена оценка на капацитета на АФВС и показателите на соларните панели.

Етапи от експериментите и диагностиката.

Измерване показателите на АФЗС в непрекъсваем режим на натоварване с консуматор IP камера за период от едно денонощие (фиг. 5)**.** Измерват се ток и напрежение и се изчисляват мощностите, енергията и капацитет.

Фиг. 5. Снимка на **опитната постановка** и примерен екран от камерата.

Блок схема на АФЗС

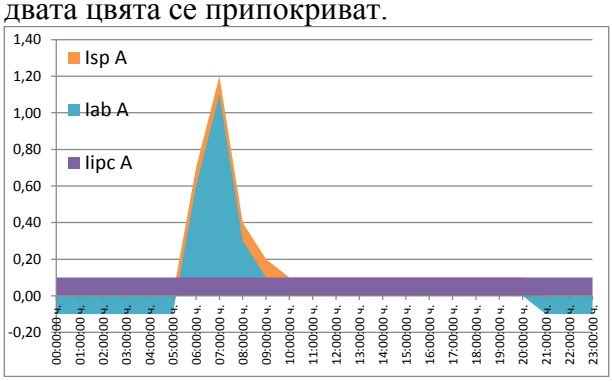


Фиг.6. Структура от свързани елементите от АФЗС за малка мощност: 1,2-соларни панели; 3-контролер заряд/разряд; 4-DC/DC 12V/5V; 5,6- акумулаторни батерии, 7 - IP видеокамера; 8- дисплей.

Таблица 1. Токове от СП, АБ и IP камера

Час	Isp	Iab	Iipc
h	A	A	A
0:00:00	0,00	-0,10	0,10
1:00:00	0,00	-0,10	0,10
2:00:00	0,00	-0,10	0,10
3:00:00	0,00	-0,10	0,10
4:00:00	0,00	-0,10	0,10
5:00:00	0,00	-0,10	0,10
6:00:00	0,70	0,60	0,10
7:00:00	1,20	1,10	0,10
8:00:00	0,40	0,30	0,10
9:00:00	0,20	0,10	0,10
10:00:00	0,10	0,00	0,10
11:00:00	0,10	0,00	0,10
12:00:00	0,10	0,00	0,10
13:00:00	0,10	0,00	0,10
14:00:00	0,10	0,00	0,10
15:00:00	0,10	0,00	0,10
16:00:00	0,10	0,00	0,10
17:00:00	0,10	0,00	0,10
18:00:00	0,10	0,00	0,10
19:00:00	0,10	0,00	0,10
20:00:00	0,10	0,00	0,10
21:00:00	0,00	-0,10	0,10
22:00:00	0,00	-0,10	0,10
23:00:00	0,00	-0,10	0,10

Трите тока са обединени на една графика, показана на фиг.7. Площта в зелен цвят от диаграмата е генерираният капацитет от СП през светлата част от деня. Графиката в зелен цвят продължава в периода от 5,30ч до 20ч паралелно с необходимия капацитет за захранване на IP камерата (площта в жълт цвят), като в този интервал от време двата цвята се припокриват.



Фиг.7. Графично представени ток от СП (I_{sp}), ток от АБ (I_{ab}) и консумиран ток от IP камера (I_{ipc}).

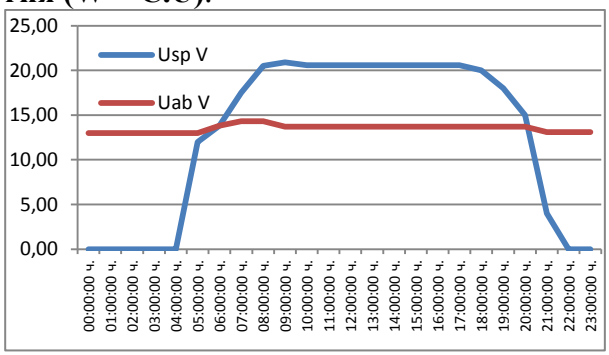
По този начин капацитета от СП се използва за захранване на IP камерата и зареждане на АБ (пиковата площ в оранжев цвят). През нощта АБ (площта в син цвят в отрицателната част на диаграмата) се разрежда през IP камерата и тогава капацитета ѝ намалява. Това са периодите от време, когато СП не генерира енергия, а се използва съхранения капацитет от АБ. Пиковата стойност на диаграмата (площта в син цвят) е получен от 5,30ч. до 9,00ч е процес на зареждане на АБ, която е захранвала IP камерата цялата нощ и се е разредила частично.

● Таблични данни от **измерените на напрежения** от АФЗС за един характерен слънчев ден по часове за 24ч.

Таблица 2. Нива на напрежението

Час	**Usp**	**Uab**
h	**V**	**V**
0:00:00	0,00	13,00
1:00:00	0,00	13,00
2:00:00	0,00	13,00
3:00:00	0,00	13,00
4:00:00	0,00	13,00
5:00:00	12,00	13,00
6:00:00	13,80	13,80
7:00:00	17,50	14,30
8:00:00	20,50	14,30
9:00:00	20,90	13,70
10:00:00	20,60	13,70
11:00:00	20,60	13,70
12:00:00	20,60	13,70
13:00:00	20,60	13,70
14:00:00	20,60	13,70
15:00:00	20,60	13,70
16:00:00	20,60	13,70
17:00:00	20,60	13,70
18:00:00	20,00	13,70
19:00:00	18,00	13,70
20:00:00	15,00	13,70
21:00:00	4,00	13,10
22:00:00	0,00	13,10
23:00:00	0,00	13,10

По данните от табл.2 е построена диаграма **на напреженията U за едно денонощие** от работата на АФЗС. Графиката е показана на фиг.8 и от нея могат да се проследят **генерираната и отдадена енергия (W = C.U)**.



Фиг.8. Графично представени напрежения получени от СП (U_{sp}) и напрежение от АБ (U_{ab}).

Напрежението произведено от СП U_{sp} от фиг.8. със син цвят е със стойности достигащи 20V при висок интензитет на слънчевата радиация в периода от 9ч. до 18ч. За да се използва за захранване на IP камерата и зареждане на АБ се включва стабилизатора от силовата част на контролера с максимално ниво на напрежението до 14,4V. Зарядният контролер не позволява презареждане на АБ при повишаване на

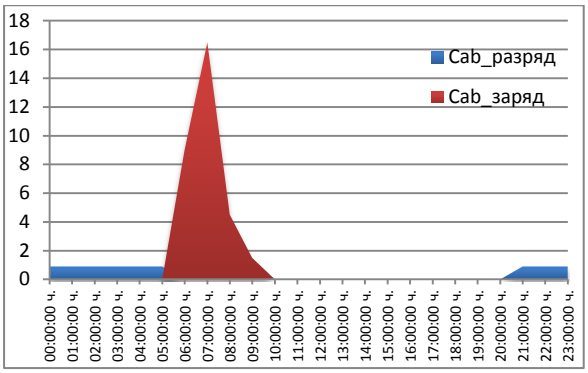
Usp. Захранващата система притежава и механизъм за регулиране широчината на импулсите (PWM) при динамичното изменение процеса на натоварване съгласувано с генерираната енергия от СП.

● Таблични данни за **капацитета на АБ** от АФЗС за един характерен слънчев ден по часове за 24ч (табл. 3).

Таблица 3. Динамика на капацитета на АБ през денонощието

час	**Cab_razregdane**	**Cab_zaregdane**
h	**Ah**	**Ah**
00:00:00 ч.	0,9	0
01:00:00 ч.	0,9	0
02:00:00 ч.	0,9	0
03:00:00 ч.	0,9	0
04:00:00 ч.	0,9	0
05:00:00 ч.	0,9	0
06:00:00 ч.	0	9,0
07:00:00 ч.	0	16,5
08:00:00 ч.	0	4,5
09:00:00 ч.	0	1,5
10:00:00 ч.	0	0,0
11:00:00 ч.	0	0,0
12:00:00 ч.	0	0,0
13:00:00 ч.	0	0,0
14:00:00 ч.	0	0,0
15:00:00 ч.	0	0,0
16:00:00 ч.	0	0,0
17:00:00 ч.	0	0,0
18:00:00 ч.	0	0,0
19:00:00 ч.	0	0,0
20:00:00 ч.	0	0,0
21:00:00 ч.	0,9	0,0
22:00:00 ч.	0,9	0,0
23:00:00 ч.	0,9	0,0

Изчислените стойности на табл. 3 показват посоката на обмен на енергия т. е. зареждане и разреждане на АБ през различните периоди от време за едно денонощие 24часа.



Фиг.9. Графично представени капацитет на разреждане и зареждане на АБ.

Диаграмата от фиг.9 визуализира в реално време **Cab_ разряден** (в син цвят) и **Cab_ заряден** (червен цвят) капацитети на АБ. От графиката се наблюдава пикови стойности на заряд от 6ч до 8,30ч. (с максимум16Ah). Този максимален режим на зареждане е когато АБ се е разреждала цяла нощ и навлиза в режим на зареждане. До към 8 часа по-голямата част от разряд се е компенсирал и АБ преминава в режим на дозареждане. Зарядният контролер от АФЗС изключва зареждането при достигане на определена степен на заряд. Той не допуска презареждане на АБ чрез настройки в алгоритъма на заряд/разряд.

3. Определяне капацитета на генераторите от АФЗС при непрекъсваемо захранване на камера за видеоаблюдение.

Капацитет на акумулаторната батерия (АБ) от АФЗС:

▪ **Заряден капацитет** на АБ-номинален при отчитане к.п.д.:

$C_{АБ\ зар.} = C_{АБ} \bullet 0{,}85 = 48.0{,}85 = 40{,}8$, А.

▪ **Капацитет на разреждане** на АБ през нощта за 10 часа, когато не работят соларните панели СП:

$$C_{АБ\ разр.нощ} = \frac{I_{разр.средно} .t_{нощ}}{\eta_{UАБ\ разр.}} = \frac{0{,}1.10}{0{,}85} = 1{,}8,\ \text{Ah}.$$

Капацитет на СП при номинални условия на осветяване и температура през деня, когато генерират енергия:

$C_{СП\ STC} = I_{СП\ STC} \bullet \eta_{СП STC} \bullet t_{ден} = 23{,}24$, Ah.

Изчислени средни стойности за периода на пиково зареждане на АБ през сутрешните часове от 6ч до 8,30. Измерваните и изчислени данни са показани в табл. 4.

Таблица 4. измерени и изчислени средни стойности.

време	$U_{СП}$	$I_{СП}$	$U_{АБ}$	$I_{АБ}$	U_{IP}	I_{IP}	$P_{IP\ кон-сум.}$	$P_{АБ\ заряд}$	$P_{АБ\ разр.}$	$P_{СП\ генер}$
часове	**V**	**A**	**V**	**A**	**V**	**A**	**W**	**W**	**W**	**W**
6-8,30	15,41	0,709	13,82	0,581	13,82	0,1	1,381	8,07	3,6	10,91

▪ Баланс на мощностите

Използват се изчислените средни стойности от табл. 4.

$P_{генер.} \geq P_{консум.}$;

$P_{генер.} = P_{СП\ генер.} + P_{АБ\ разр.}$;

$P_{консум.} = P_{IP} + P_{АБ\ заряд}$;

$P_{СП\ генер.} + P_{АБ\ разр.} = P_{IP} + P_{АБ\ заряд}$;

$P_{генер.}$ = 10,91 + 3,6 = 14,51 W

$P_{консум.}$ = 1,381 + 8,07 = 9, 45 W

Условието за баланс на мощностите е изпълнено за АФЗС:

$P_{генер.} \geq P_{консум.}$, т.е. 14,41 ≥ 9,45.

▪ Изчислени средни стойност на необходимия минимален инсталационен капацитет **$\Sigma\ C_{консум.}$** на АФЗС за захранване на IP камерата по данните от табл.4:

$C_{консум.\ ден\ камера} = 0,1 \bullet 14 = 1,4$, Ah;

$C_{АБ\ зар.\ ден} = 8,2$, Ah;

$C_{АБ\ разр.\ нощ} = 1,4$, Ah;

$C_{консум.\ ден} = 10$, Ah;

$C_{консум.\ нощ} = 1$, Ah;

$\Sigma\ C_{консум.} = C_{консум.\ нощ} + C_{консум.\ ден} = 11$, Ah.

Баланс на необходимата енергия W за нормалната работа на АФЗС.

$W_{генер.} \geq W_{консум.}$;

$W_{генер.} = W_{СП\ генер.} + W_{АБ\ разр.}$;

$W_{консум.} = W_{IP} + W_{АБ\ заряд}$.

Средните стойности на номиналните енергийни показатели за източниците са:

$W_{СП\ STC} = C_{СП\ STC} \bullet U_{СП\ STC.} = 217,3$ [W.h];

$W_{АБ\ зар.} = C_{АБ\ зар.} \bullet U_{АБ\ зар} = 530$ [W.h];

Отдадената енергия от АБ към камерата през нощта е:

$W_{АБ\ разр.\ нощ} = C_{АБ\ разр.\ нощ} \cdot U_{АБ} = 19,35$,Wh

Отдадената енергия от СП –средна стойност за ден е

$W_{СП\ генер.}$ = 0,79.15,41.14 = 170,4 ,Wh

Консумираната енергия при зареждане на АБ е:

$W_{АБ\ зар.\ ден} = C_{АБ\ зар.\ ден} \cdot U_{АБ} = 113,3$,Wh.

Консумирана енергия от камерата за 24часа:

$W_{IP} = P_{IP} \cdot t_{24ч.} = 1,38 \cdot 24 = 33,12$,Wh;

$W_{генер.} = W_{СП\ генер.} + W_{АБ\ разр.} = 189,7$,Wh;

$W_{консум.} = W_{IP} + W_{АБ\ заряд} = 146,42$, Wh;

Коефицент на полезно действие Η за работа на АФЗС в непрекъсваем режим.

За едно денонощие (24часа):

$\Eta_{ден} = W_{консум.} / W_{генер.} = 0,77 = 77\%$

ЗАКЛЮЧЕНИЕ

1.Капацитетът на АФЗС ще бъде балансиран при $W_{генер..} \geq W_{консум..}$ за да може да се изпълни условието за работа на камерата в непрекъсваем режим.

2.Високата стойност на к.п.д. се определя от степента на заряд разряд на АБ и енергийния капацитет на генераторите при продължителен автономен режим на работа АФЗС

3.Проектирана и реализирана в учебна лаборатория по Електротехника на катедра ЕЕМ към Университет „Проф. д-р Асен Златаров“, гр. Бургас АФВС може да намери приложение за практическо изработване на лабораторни упражнения в часовете по дисциплините: Електротехника и електроника, Електрически измервания, ВЕИ, Нетрадиционни и ВЕИ др.

4.Автономната фотоволтаична система може да се приложи за захранване на консуматори не зависещи от електрическата мрежа като осветителни тела, консуматори за постоянен ток с изходна мощност не по-голяма от реализираната.

5.Приложение на АФВС като независим източник за маломощни консуматори, които изискват непрекъсваемо захранване като медицинско оборудване, пожаро и звуково известяване, при взривоопасни обекти и материали и др.

ЛИТЕРАТУРА

[1] Д, Воденичаров, дипломна работа „Изследване капацитета на автономна фотоволтаична система, захранваща IP

видеокамера в непрекъсваем работен режим'', Бургас, 2019 г.
[2] http://www.udomi.de/downloads/luxor-sololine.pdf
[3] https://www.victronenergy.com/solar-charge-controllers/bluesolar-pwm
[4] http://tebix.bg/bg/
[5] Стоянов, И. Изследване работата на акумулаторна батерия на фотоволтаична система в програмната среда на LabVIEW, Научни трудове на Русенския университет 2010, Том 49, серия 3.1 Електротехника, електроника, автоматика, 2010, стр. 51-56.
[6] Стоянов, И., Н. Михайлов. Изследване възможностите за използване на слънчевата енергия в района на град Русе, Научни трудове на Русенския университет "Ангел Кънчев" 2003, том 40, серия 1.1., Електротехника, електроника, автоматика, физика, 2003, Печатна база на РУ, стр.137-141.
[7] Стоянов, И. Обследване характеристиките на фотоволтаичен елемент при директно преобразуване на слънчевата енергия в електрическа, Научни трудове на Русенския университет "Ангел Кънчев" 2006, том 40, серия 1.1, Електротехника, електроника, автоматика, физика, Печатна база на РУ, 2006, стр.141-146, ISBN:1311-3321.

Адреси за контакти:
гл. ас д-р Васил Борисов Иванов
Университет „Проф. д-р Асен Златаров", 8000 Бургас
e-mail: vasil_bi@abv.bg

инж. Христо Борисов Иванов
Областна администрация- Бургас

ОБЗОРЕН ПРЕГЛЕД НА АРХИТЕКТУРАТА НА СИСТЕМИТЕ ЗА ЗАДВИЖВАНЕ НА ЕЛЕКТРИЧЕСКИ ПРЕВОЗНИ СРЕДСТВА

Росен Даскалов, Иван Белоев
Русенски университет „Ангел Кънчев”

AN OVERVIEW OF THE POWERTRAIN TOPOLOGIES FOR ELECTRIC VEHICLES

Rosen Daskalov, Ivan Beloev
University of Rousse

Abstract: This article provides an overview of the powertrain topologies that are used in the electric vehicles. The series, the parallel and the series-parallel hybrid powertrain topologies for electric vehicles were discussed and presented.

Key words: Electric vehicle, Hybrid electric vehicle, Energy sources, Powertrain architectures.

ВЪВЕДЕНИЕ

В момента секторът на сухопътния транспорт е изправен пред важни предизвикателства по отношение на енергийния транспортен модел, базиран на петролни продукти. Използването на тези горива в превозните средства с двигатели с вътрешно горене (ДВГ) е основната причина за замърсяване в градовете, което води до силно намаляване на качеството на живот на жителите. Замърсяването на въздуха не само причинява дихателни проблеми на възрастни и деца, но също така е в пряка зависимост на причините за зачестяване случаите на инфаркт, инсулт, разстройства на метаболитните процеси и има дори връзка с диабета. Поради тази причина в световен мащаб се води политика и се придава голямо значение на този въпрос, като са поети редица инициативи, които насърчават развитието на проекти за разработване и интегриране на автомобили с ниски вредни емисии [5].

Освен екологичните аспекти, трябва да се вземат предвид икономическите и политическите аспекти. В днешно време производственият капацитет на петролни продукти се е увеличил поради използването на високотехнологични производствени мощности и като следствие цените на петрола са паднали. Въпреки това развитите страни продължават да популяризират алтернативните енергии за автомобилен транспорт, за да намалят колкото е възможно повече зависимостта от петрола. Освен това зависимостта и прекомерната употреба на петрол водят до геополитически проблеми поради нестабилността на някои страни производителки и използването на петрола като икономическо оръжие. В този контекст обществото отдавна е разбрало, че превозните средства с ДВГ не трябва да бъде бъдещето на мобилността. По тази причина се разработват алтернативни на конвенционалните превозни средства през последните десетилетия. Предприети са редица действия за намаляване на зависимостта от петролни продукти. От една страна е развит обществения транспорт, особено електрическите влакове (високоскоростни влакове, трамваи и метро). От друга страна, ефективността в личния транспорт е подобрена чрез намаляване теглото на превозните средства и използване на по-ефективни ДВГ. Подобренията, направени на ДВГ, обаче не са достатъчни за драстично намаляване на разхода на петролни продукти. Предложените решения включват превозни средства, задвижвани от биодизел, етанол, сгъстен природен газ (CNG), втечнен природен газ (LNG), водород, сгъстен въздух, електрически превозни средства и др. През последното десетилетие изглежда, че електрическите превозни средства са водещи по отношение на ефективността и устойчивостта [3,5].

ОБЩА КЛАСИФИКАЦИЯ НА ЕПС

Наложената дефиниция за електрическо превозно средство (ЕПС) се определя като превозно средство, което използва поне един електрически двигател и батерия за нуждите на задвижването. По този начин plug-in електрическите превозни

средства (PEV) могат да се разглеждат като под категория на електрическите превозни средства [1]. Няма международно определение за електрическо превозно средство, но според литературата PEV може да бъде определено като леко превозно средство, което черпи енергия от батерия с капацитет най-малко 4kWh и може да се зарежда от външен източник. В рамките на тази дефиниция могат да се разграничат няколко типа PEV, главно plug-in хибридни електрически превозни средства (PHEV), електрически превозни средства с акумулаторни батерии (BEV) и хибридни електрически превозни средства с включени горивни клетки (FC-PHEV). За разлика от тях, превозни средства, като хибридни електрически превозни средства (HEV) не се считат за PEV. Батериите на тези типове превозни средства се зареждат с електричество, генерирано чрез двигател с вътрешно горене и не могат да се зареждат с електричество, пренесено от разпределителната електрическа мрежа [2].

Plug-in хибридните електрически превозни средства се задвижват от електродвигател и двигател с вътрешно горене и могат да бъдат включени към електрическата мрежа за зареждане на батериите. За осигуряване на енергия за задвижване на автомобила се използват ДВГ и батерия. По този начин се получават предимствата на двата вида технологии. Тоест ДВГ осигурява по-добра автономност и време за зареждане, докато електрическия двигател намалява вредните емисии и увеличава общата ефективност на системата. Освен това батериите на PHEV могат да бъдат частично заредени чрез регенеративна спирачна система, така че се постига допълнително подобрение на ефективността в сравнение с конвенционалните автомобили [3].

Двигателите с вътрешно горене имат ниско КПД при ниско или частично натоварване. Анализи показват, че градското шофиране е най-неефективния режим за автомобили с ДВГ. С използването на електрически двигател ефективността при частично натоварване значително се подобрява. Освен това електрическите двигатели почти не изискват енергия, когато не се използват (т.е. когато автомобилът е спрян на светофар). В този контекст има няколко разработки за спиране на ДВГ, когато автомобилът е спрян или работи на празен ход, за да се спести гориво. Тези разработки са известни като старт / стоп системи.

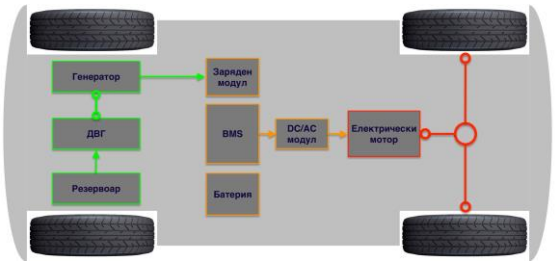
PHEV са по-сложни за реализация от автомобилите с ДВГ и тези задвижвани само от батерии, поради използването на два различни двигателя. Има три различни архитектури за интегриране на ДВГ и електрически двигател в превозно средство, които са класифицирани както следва: сериен хибрид, паралелен хибрид и последователно – паралелен хибрид.

ПОСЛЕДОВАТЕЛНИ ХИБРИДНИ СИСТЕМИ ЗА ЗАДВИЖВАНЕ

В последователната хибридна топология (S-PHEV) тягата се генерира само от електрически двигател, а електрическата енергия за захранването може да се черпи или от бордовата батерия, или от ДВГ чрез електрически генератор, както е показано на фиг.1. Освен батерии, могат да се използват и супер кондензатори като система за акумулиране на енергия. Поради големия динамичен диапазон на електрическия двигател, няма нужда от многостепенна скоростна кутия и в резултат на това теглото намалява, а КПД на автомобила се подобрява.

Освен това генерираната мощност от системата ДВГ - генератор е различна от мощността, изисквана за моментното задвижване. Благодарение на тази характеристика се получават две предимства. От една страна, ДВГ може да работи в точката си на максимална ефективност, дори ако автомобилът променя скоростта си, докато батерията осигурява останалата част от нужната мощност, действайки като енергиен буфер. От друга страна, устройството на PHEV е по-просто, защото системата ДВГ-генератор не е свързана с колелата. Тази топология обаче има важен недостатък, свързан с броя на необходимите преобразувания на енергия [6]. Механичната мощност на ДВГ трябва да се

преобразува в електрическа мощност и след това отново в механична мощност. Това води до съответна загуба на ефективност, особено при високи скорости. Поради тази причина тази топология е по-удобна за ниски скорости. Освен това задвижващата система трябва да бъде проектирана така, че да отговаря на максимално устойчивото потребление на енергия и в резултат на това електрическият мотор и батерията са доста големи, което повишава цената на тази конфигурация [6].

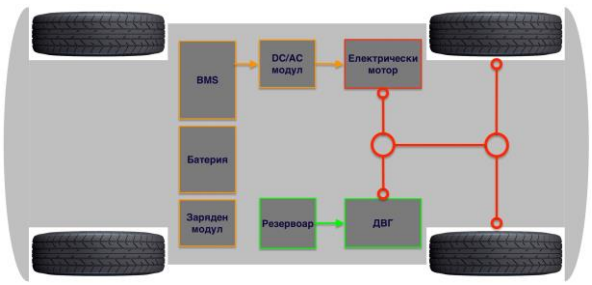


Фиг. 1 Топология на последователна хибридна система за задвижване

Обикновено този тип конфигурация се нарича още електрическо превозно средство с удължен обхват (EREV), където малък ДВГ е включен в електрическото превозно средство, за да се осигури резервно захранване в случай, че батериите са напълно изтощени.

ПАРАЛЕЛНИ ХИБРИДНИ СИСТЕМИ ЗА ЗАДВИЖВАНЕ

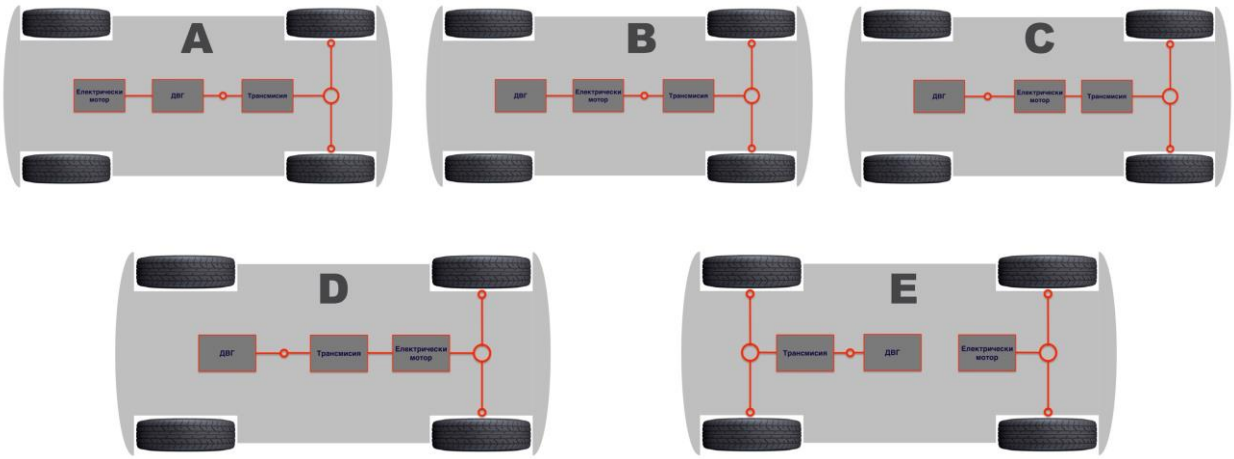
Паралелната хибридна топология (P-PHEV) е най-често срещаната от хибридните архитектури и съответно най-евтината. В тази топология (фиг. 2) тягата може да бъде осигурена от двата двигателя.



Фиг. 2 Топология на паралелна хибридна система за задвижване

Механичното предаване е по-сложно, отколкото при последователната хибридна система, но от своя страна електрическият генератор е премахнат. ДВГ трябва да бъде проектиран да осигурява максимално продължително потребление на енергия, тъй като електродвигателят работи само ако батерията не е изтощена. По този начин ДВГ в паралелна хибридна архитектура е по-голям, отколкото в последователната хибридна, но електрическият мотор и батериите са по-малки. В резултат на това, тази топология е по-подходяща за високи скорости. Въпреки това, той е по-малко ефективен при ниски скорости или в задръствания поради намаления капацитет на батерията и мощността на електродвигателя [2].

Според местоположението на модула за обединяване на въртящия момент от ДВГ и електродвигателя, могат да бъдат получени пет различни паралелни архитектури производни на основната (фиг. 3). Тип А паралелна архитектура е подходяща както за малки HEV, така и за среден тип HEV. В сравнение с други архитектури, тази е най-рентабилният паралелен хибрид, защото се внедрява с изключително ограничени промени в конвенционалната силова платформа. Архитектура тип В изисква електродвигателя да бъде монтиран в пространството между маховика на двигателя и трансмисионното устройство. Тази архитектура споделя общи енергийни възможности с архитектура тип А, но нейното внедряване е изключително скъпо. При архитектура тип С, електродвигателя може да поддържа превозното средство да се движи със скорост до няколко десетки км/час. Тъй като електродвигателя може да бъде изолиран от ДВГ чрез съединител, този тип се характеризира с работа в реален PEV режим, без загуба от триенето на подвижните елементи в ДВГ. В случаите когато се стигне до разреждане на батерията под предварително зададената прагова стойност или когато изискването за енергия надвишава капацитета на електрическата система, ДВГ автоматично се стартира и съединителят се включва постепенно. Тип С е много обещаваща архитектура благодарение на баланса в разходите за внедряване, размера на системата, потенциала за енергоспестяване



Фиг. 3 Топологични разновидности на паралелната хибридна система за задвижване

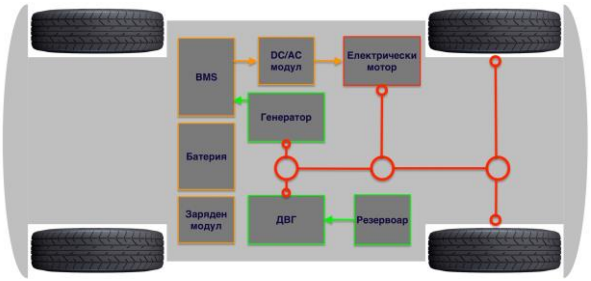
и оперативната гъвкавост. Архитектура тип D е подобна на предходната, но при нея електродвигателя предава въртящия момент към колелата с фиксирано предавателно отношение. Съществува и алтернативна паралелна хибридна система тип E, която се състои от ДВГ, осигуряващ въртящ момент на една от осите на превозното средство, докато електрически мотор прави същото с другата ос.

По този начин механичната система е опростена и може да се осигури сцепление на четирите колела. В тази конфигурация електрическият двигател не само осигурява мощност, но също така може да се използва за регенеративно спиране за зареждане на батерията, когато водачът спира или когато батерията е изтощена.

ПОСЛЕДОВАТЕЛНО - ПАРАЛЕЛНИ ХИБРИДНИ СИСТЕМИ ЗА ЗАДВИЖВАНЕ

В последователно паралелна конфигурация (SP-PHEV) ДВГ може да доставя механичен въртящ момент директно, както и електрическа мощност чрез генератор, както е показано на фиг.4. За целта той включва така наречената устройство за разделяне на мощността, което позволява предаването на въртящия момент от ДВГ на колелата или на генератора [2, 6].

В тази топология системата за управление определя най-добрия баланс между двете задвижващи системи за постигане на по-добра ефективност и производителност. Системата обаче е механично по-сложна и се нуждае от генератор, като например в последователната хибридна конфигурация.

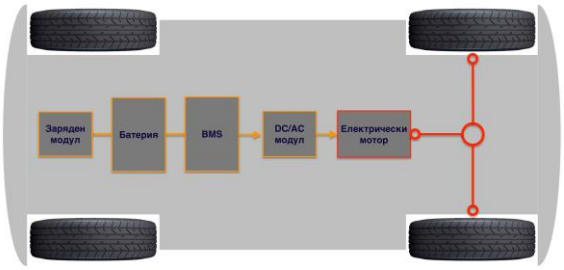


Фиг. 4. Топология на последователно – паралелна хибридна система за задвижване

В резултат на това цената е относително по-висока в сравнение с топологиите на последователни и паралелни задвижващи механизми.

ЕЛЕКТРИЧЕСКИ СИСТЕМИ ЗА ЗАДВИЖВАНЕ ЗАХРАНВАНИ ОТ БАТЕРИИ

Акумулаторните електрически превозни средства се захранват само от електричество, получено от батерия или от супер кондензатори. Батериите се зареждат от електрическата мрежа или от регенеративна спирачна система, когато автомобилът спира. Архитектурата на BEV е по-малко сложна, отколкото при PHEV и е показана на фиг.5. Този тип електрическо превозно средство се нуждае от тежки батерии, за да постигне достатъчно автономност за ежедневна употреба [4].

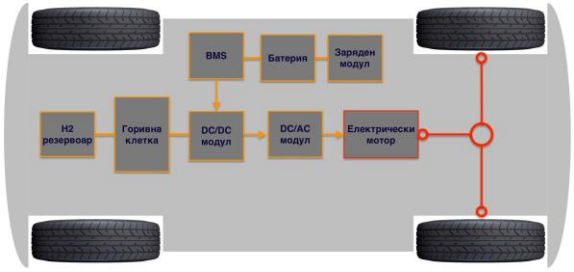


Фиг. 5. Топология на електрическо превозно средство захранвано от батерии

В резултат на това първоначалните разходи са по-високи, въпреки че се очаква цената за kWh на батерията да продължи да пада, което прави този тип PEV икономически конкурентен. Въпреки това, тези автомобили имат няколко недостатъка, свързани с технологията на батерията, например: бавно време за зареждане и ограничена автономност.

ХИБРИДНИ ЕЛЕКТРИЧЕСКИ СИСТЕМИ ЗА ЗАДВИЖВАНЕ С ВКЛЮЧЕНИ ГОРИВНИ КЛЕТКИ

Горивните клетки преобразува водород и кислород (получени от въздуха) в електричество, топлина и вода, без да отделя вредни емисии. Този тип електрическо превозно средство може да се класифицира като сериен хибриден електрически автомобил, но в този случай вместо ДВГ се използва горивна клетка, захранвана от водород фиг.6 [1].



Фиг. 6. Топология на електрическо превозно средство с горивна клетка

По този начин се подобрява общата ефективност, тъй като ефективността на горивните клетки може да достигне 40-50% в сравнение с 35% от дизеловия ДВГ. Освен това, за разлика от ДВГ, ефективността на горивните клетки остава висока при ниски натоварвания. Работата на системата е доста подобна, както при S-PHEV: горивната клетка работи често в точката си на максимална ефективност, докато батерията доставя или абсорбира липсата или излишъка от енергия [1]. Въпреки че водородът може да се получи чрез чист процес на електролиза, понастоящем водородът обикновено се генерира чрез преобразуване на метан или природен газ. След това водородът може да се съхранява в компресирано състояние в резервоари. Времето за презареждане на този тип превозни средства е много бързо (около 5 минути) в сравнение със зареждането на батерията от мрежата.

ЗАКЛЮЧЕНИЕ

Електрическите превозни средства имат голям потенциал да се превърнат в бъдещето на транспорта, като същевременно спасяват планетата от непосредствени бедствия, причинени от глобалното затопляне. Те са жизненоважна алтернатива на конвенционалните превозни средства, които зависят пряко от намаляващите запаси от изкопаеми горива. Целта на тази статия е да класифицира и покаже общата структурна схема на различните топологии на системите за задвижване на ЕПС.

ЛИТЕРАТУРА

[1] A.l.i. Emadi, Kaushik Rajashekara, Sheldon S. Williamson, et al., Topological overview of hybrid electric and fuel cell vehicular power system architectures and configurations, IEEE Trans. Veh. Technol. (2005) 763–770.

[2] C.C. Chan, Alain Bouscyrol, Electric, hybrid, and fuel-cell vehicles: architectures and modeling, IEEE Trans. Veh. Technol. 59 (2) (2010) 589–598.

[3] E.D. Tate, Michael Harpster and peter Savagian, The electrification of the automobile: from conventional hybrid, to plug-in hybrids, to extended-range electric vehicles, in: Proceedings of 2008 World Congress, Detroit, Michigan, 2008.

[4] Grunditz, E.A.; Thiringer, T. Performance Analysis of Current BEVs Based on a Comprehensive Review of Specifications. IEEE Trans. Transp. Electr. 2016, 2, 270–289.

[5] NHTSA, E.P.A., Joint Technical Support

Document: Final Rulemaking for 2017 – 2025 Light-Duty Vehicle Greenhouse Gas Emission Stadnards And Corporate Average Fuel Economy Standards, 2012.

[6] Yimin Gao, Mehrada Ehsani, Hybrid electric vehicles: architecture and motor drives, Proc. IEEE 95 (4) (2007) 719–728.

Адрес за контакти:
доц. д-р инж. Иван Х. Белоев
Катедра „Транспорт”
Русенски Университет „Ангел Кънчев”
ул. „Студентска” № 8
7017, Русе, България
тел. +359 82 888 605
E-mail: ibeloev@uni - ruse.bg

Assoc. Prof. Ivan H. Beloev
Department of Transport
University of Rousse „Angel Kanchev”
8 Studentska Str.
7017 Rousse, Bulgaria
Tel +359 82 888 605
E-mail: ibeloev@uni - ruse.bg