

Для зменшення часових затрат щодо виявлення рухомих об'єктів на зображенні запропоновано використання методів, які ґрунтуються на використанні часових різниць, а для уникнення можливих спотворень – використання морфологічних перетворень.

Література

1. Національна доповідь про стан техногенної та природної безпеки в Україні у 2014 році. – 2015. [Електронний ресурс]. – Доступний з http://mns.gov.ua/files/prognoz/report/2014/ND_2014.pdf.
2. Бровко Н. Анализ методов обработки последовательностей видеоизображений в приложении к задаче раннего обнаружения пожаров / Н. Бровко, Р. Богуш // Вестник Полоцкого государственного ун-та : сб. науч. тр. – Сер.: С. – 2011. – С. 42-50.
3. Arjun S. Lab Color Space Model with Optical Flow Estimation for Fire Detection in Videos / S. Arjun, E. Vinoth // IOSR Journal of Computer Engineering, 2014. – Pp. 22-38.
4. Toulouse T. Automatic fire pixel detection using image processing: A comparative analysis of Rule-based and Machine Learning-based methods / T. Toulouse, L. Rossi, T. Celik, M. Akhloufi // Signal, Image and Video Processing manuscript, 2015. – Pp. 1-8.
5. Chino D. BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis / D. Chino, L. Avalhais, R. Jose, T. Agma // Proceedings of the 28th SIBGRAPI Conference on Graphics, Patterns and Images, 2015. – Pp. 95-102.
6. Celik T. Automatic Fire Detection in Video Sequences / T. Celik, H. Demirel // European Signal Processing Conference. – 2006. – № 2. – Pp. 1-5.
7. Celik T. Fast and efficient method for fire detection using image processing / T. Celik // ETRI journal. – 2010. – № 6. – Pp. 881-890.
8. Celik T. Fire and smoke detection without sensors: image processing based approach / T. Celik, H. Ozkaramanli, H. Demirel // 15th European Signal processing conference, 2007. – Pp. 1794-1798.
9. Celik T. Fire detection in video sequences using a generic color model / T. Celik, H. Demirel // Fire Safety Journal. – 2009. – № 2. – С. 147-158.
10. Celik T. Fire pixel classification using fuzzy logic and statistical color model / T. Celik, H. Ozkaramanli, H. Demirel // Acoustics, Speech and Signal Processing, 2007. – Pp. 1205-1208.
11. Che-Bin L. Vision based fire detection / L. Che-Bin, N. Ahuja // Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. – 2004. – № 4. – Pp. 134-137.
12. Chen T. An early fire-detection method based on image processing / T. Chen, P. Wu, Y. Chiou // Image Processing, 2004. – Pp. 1707-1710.
13. Gomes P. A Vision-based Approach to Fire detection / P. Gomes, P. Santana, J. Barata // International Journal of Advanced Robotic Systems, 2014. – Pp. 107-114.
14. Jin H. A fire and flame detecting method based on video / H. Jin, R. Zhang // Machine Learning and Cybernetics, 2009. – Pp. 2347-2352.
15. Ko B. Fire detection based on vision sensor and support vector machines / B. Ko, K. Cheong, J. Nam // Fire Safety Journal. – 2009. – № 44. – Pp. 322-329.
16. Kurup A.R. Vision based fire flame detection system using optical flow features and artificial neural network / Anila R. Kurup // Int J Sci Res, 2012. – Pp. 2161-2168.
17. Philips W. Flame recognition in video / W. Philips, M. Shah, N. Lobo // Pattern Recognition Letters. – 2002. – № 23. – Pp. 319-327.
18. Qi X. A Computer Vision-Based Method for Fire Detection in Color Videos / X. Qi, J. Ebert // International journal of imaging. – 2009. – № 9. – Pp. 22-34.
19. Rossi L. On the use of stereovision to develop a novel instrumentation system to extract geometric fire fronts characteristics / L. Rossi, M. Akhloufi, Y. Tison // Fire Safety Journal. – 2011. – № 46. – Pp. 9-20.
20. Soe A. Fire detection by static image characteristics of the light blue flame using roi-based color intensity composition detection algorithm / A. Soe, X. Zhang // ICIC Express Letters. – 2011. – № 12. – Pp. 4479-4486.
21. Bedo M. Techniques for effective and efficient fire detection from social media images / M. Bedo, G. Blanco, W. Oliveira et al. // International Conference on Enterprise Information Systems, 2015. – Pp. 34-45.
22. Bo J. Towards a Solid Solution of Real-time Fire and Flame Detection / J. Bo, L. Yongyi, L. Xiyang, L. Liang // Multimedia Tools and Applications. – 2015. – № 3. – Pp. 689-705.

Надійшла до редакції 21.06.2016 р.

Максимиw А.П., Рак Т.Е., Пелешко Д.Д. Аналитический обзор методов детектирования огня с помощью средств компьютерного зрения

Осуществлен аналитический обзор современных методов детектирования огня с помощью средств компьютерного зрения. В результате проведенной работы выявлено, что на сегодняшний день проблема разработки системы детектирования пламени актуальна и требует проведения дальнейших исследований. Представлен сравнительный анализ эффективности работы отдельно взятых методов и отмечены проблематичные моменты, которые вызывают снижение качества детектирования. Отмечены перспективные направления развития алгоритмов, которые, по нашему мнению, позволят в значительной степени повысить эффективность системы детектирования пламени.

Ключевые слова: детектирование пламени, компьютерное зрение, видеонаблюдение, автоматизация.

Maksymiv O.P., Rak T.E., Peleshko D.D. The Analytical Review of Fire Detecting Method by Using Computer Vision

The paper made an analytical review of modern fire detection methods by means of computer vision. The results of this work revealed that despite a number of publications in this field, today the problem of developing a flame detection system is urgent and requires further research. This paper presents a comparative analysis of the individual methods efficiency and marked problematic moments that cause reduced quality of detection. Noted the perspective directions of algorithms, which we believe will greatly improve the efficiency of detecting flames.

Keywords: flame detection, computer vision, video surveillance, automation.

УДК 004.[4+896]

ПРОГРАМНА МОДЕЛЬ ПІДСИСТЕМИ КЕРУВАННЯ МОБІЛЬНОЮ РОБОТОТЕХНІЧНОЮ СИСТЕМОЮ

К.В. Матвійчук¹, В.М. Теслюк², А.Я. Зелінський³

Описано особливості розроблення програмного забезпечення підсистеми віддаленого керування робототехнічною системою (РТС). Побудовано UML діаграму прецедентів, за допомогою якої відображено основні завдання, які виконуватиме РТС у процесі функціонування. Для реалізації ПЗ обрано клієнт-серверну архітектуру. Побудовано UML діаграми діяльностей для клієнтської та серверної частин ПЗ, описано принципи їх взаємодії та представлено блок-схеми алгоритмів їх роботи. Розроблено та описано інтерфейс користувача, для клієнтської частини ПЗ, обґрунтовано вибір мови програмування для реалізації функціоналу системи.

Ключові слова: робототехнічна система, програмне забезпечення, UML-діаграма, клієнт-серверна архітектура.

Вступ. Автоматизація проектування та розроблення робототехнічних систем займає провідне місце серед сучасних напрямів розвитку світової науки. Адаже дає змогу не тільки автоматизувати складні технологічні процеси та операції, але й замінити людину під час виконання важких і небезпечних робіт. Тому важливим завданням є процес проектування систем управління механізмами, технічного забезпечення, взаємодії цих складників, та розроблення спеціалізованого програмного забезпечення. Адаже від цього етапу напрямку залежить фун-

¹ аспір. К.В. Матвійчук – НУ "Львівська політехніка";

² проф. В.М. Теслюк, д-р техн. наук – НУ "Львівська політехніка";

³ асист. А.Я. Зелінський, канд. техн. наук – НУ "Львівська політехніка"

кціонування та цілісність розроблюваної системи. Використання технічних засобів, які забезпечують збирання, реєстрацію, передачу, зберігання, опрацювання та тиражування інформації, а також процеси взаємодії людини і комп'ютера для розв'язання задач управління РТС. Аналіз існуючих систем керування показав їх велике різноманіття, залежно від виконуваних завдань та ступеня їх автоматизації. Такі системи можуть бути повністю керовані як людиною, так і програмою, або ж в поєднанні цих способів. Це РТС з адаптивною системою управління, оснащені сенсорною частиною, з інтерактивним типом управління – беруть участь у діалозі з людиною відносно вибору стратегії управління, системи з біотехнічним управлінням, з інтелектуальним управлінням – засновані на застосуванні методів штучного інтелекту, та автоматичні – функціонують самостійно [1-6].

Особливості розроблення ПЗ підсистеми керування РТС. На основі розроблених структурної та функціональної схем підсистеми віддаленого управління РТС, для реалізації ПЗ було використано клієнт-серверну архітектуру [7, 8]. Ця архітектура сьогодні є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона містить такі основні компоненти: набір серверів, які надають інформацію або опрацьовують дані, отримані від клієнтів, які звертаються до них; набір клієнтів, які використовують сервіси, що надаються серверами; мережа, яка забезпечує взаємодію між клієнтами та серверами.

Для відображення поставленого завдання використаємо діаграму прецедентів. Суть цієї діаграми полягає в такому: проєктована система видається у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою т. зв. варіантів використання. Варіант використання (use case) слугує для опису сервісів, що система надає актору. Для розроблення ПЗ використано таку use case діаграму (рис. 1).

Програмне забезпечення підсистеми віддаленого управління, відповідно до представленої use-cases діаграми, працює за таким алгоритмом:

1. Серверна програма має ініціалізувати створення мережі та очікувати підключення клієнтів.
2. Наступним кроком клієнт має підключитися до створеної мережі.
3. Після успішного встановлення з'єднання із сервером клієнтська програма повинна відіслати серверу команди управління танком.
4. Сервер, своєю чергою, після отримання команд ідентифікує їх і конфігурує управлінське рішення для однієї із плат управління танком.

Розроблення діаграми діяльності підсистеми керування РТС

Діаграма діяльності в UML – візуальне представлення графу діяльностей [9]. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Специфікація діяльності може давати змогу виконати кілька потоків операцій та застосовувати механізми синхронізації

для гарантування виконання дій у правильному порядку. Зокрема, діаграму діяльності серверної частини програми зображено на рис. 2, а діаграму діяльності клієнтської частини програми – на рис. 3.

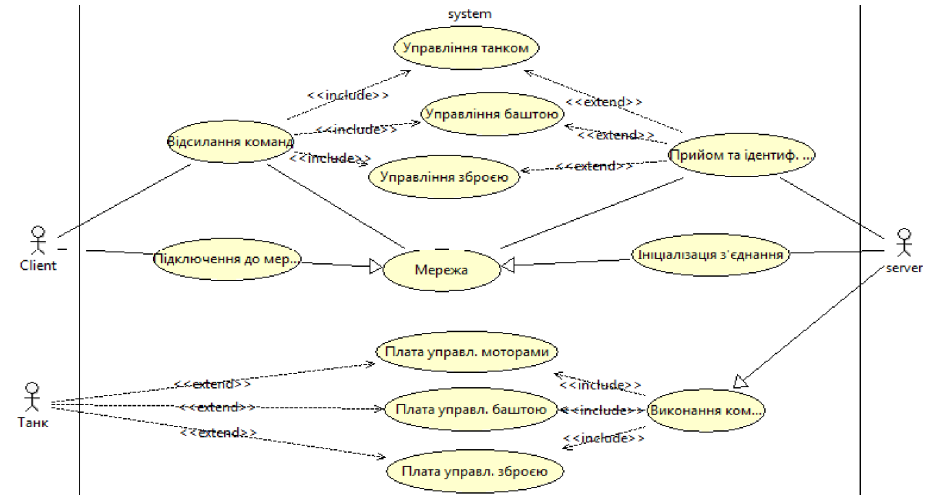


Рис. 1. Діаграма прецедентів програмного забезпечення

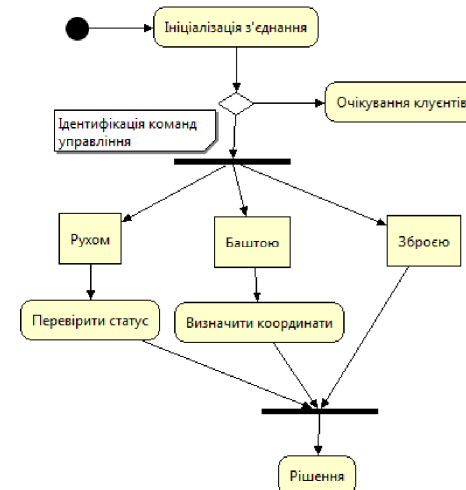


Рис. 2. Діаграма діяльності сервера

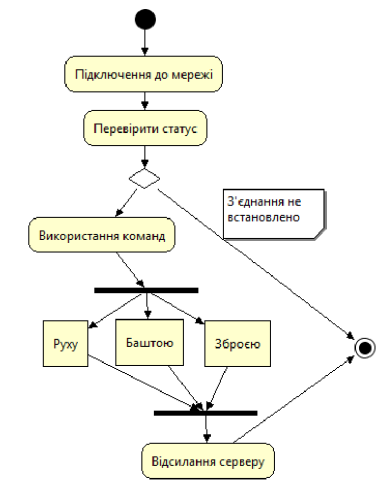


Рис. 3. Діаграма діяльності клієнта

Діаграма діяльності відображає, які дії повинен виконати клієнт і сервер у процесі функціонування підсистеми. Як випливає із рис. 2 і 3, сервер наділений більшою функціональністю, що цілком закономірно. Крім отримання команд від клієнта, він повинен перевірити стан РТС (чи РТС переміщується і в якому стані знаходяться складники РТС). Визначення координат, для прикладу башти, потрібне при використанні у програмі абсолютних чи відносних кутів.

У розроблювальній підсистемі клієнтом виступає клієнтський комп'ютер, телефон чи планшет. Відповідно, сервером може виступати мікрокомп'ютер Raspberry Pi чи якийсь інший [4, 8]. Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; у такій ситуації сервер часто називають локальним. У нашому випадку клієнтська програма розміщена на ПК, а серверна віддалена – розміщується на мікрокомп'ютері Raspberry Pi. Модель клієнт-серверної взаємодії визначається насамперед розподілом обов'язків між клієнтом та сервером. Логічно можна виокремити три рівні операцій:

- рівень представлення даних, який по суті є інтерфейсом користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд – клієнтська програма;
- прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідне опрацювання інформації – серверна програма;
- рівень управління даними, який забезпечує зберігання даних та доступ до них – серверна програма.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. Потрібно зауважити, що не обов'язково клієнтська і серверна програми мають бути розміщені в одній і тій самій платформі.

Особливості реалізації спеціалізованого ПЗ підсистеми управління РТС. Відповідно до використаної структури ПЗ розроблено дві програми, які розміщені на різних платформах. Як внаслідок, програма сервера створена на платформі Linux, а програма клієнта – на XP, Windows 7, Windows 8. На цей час створюється програма і для платформи Android (для використання планшетів, мобільних телефонів).

Приклад блок-схеми алгоритму роботи сервера зображено на рис. 4. Після запуску РТС відбувається ініціалізація підсистем сервера для роботи в мережі. Зі сторони клієнта надсилається команда створення з'єднання між РТС та оператором. Після встановлення з'єднання підсистема керування РТС отримує команди, після ідентифікації яких, виконує необхідні користувачеві дії. Для написання програми сервера доцільно використати мову python, оскільки тут дуже легко встановити і використовувати бібліотеку для роботи із портами RPI.GPIO та бібліотеки для роботи в мережі [10].

Для використання згаданої вище бібліотеки потрібно використати таку команду `import Rpi.GPIO as GPIO`, а для використання порту UART (com порту) Raspberry треба використати бібліотеку `serial` (`import serial`). Наприклад, для ініціалізації порту, використаємо об'єкт `GPIO.setmode(GPIO.BOARD)`.

Для керування пінами Raspberry потрібно використати набір команд: `GPIO.setup(f1, GPIO.OUT)`, `GPIO.output(f1, True)`. За допомогою цих команд порт `f1` ініціалізується як вихід, а остання команда відсилає на нього "логічну одиницю".

Для встановлення з'єднання з клієнтом використовують модуль

```
import socket
s.bind((HOST, PORT)),
```

де: *HOST* – адреса Raspberry Pi; *PORT* – порт, для обміну повідомленнями з клієнтом.

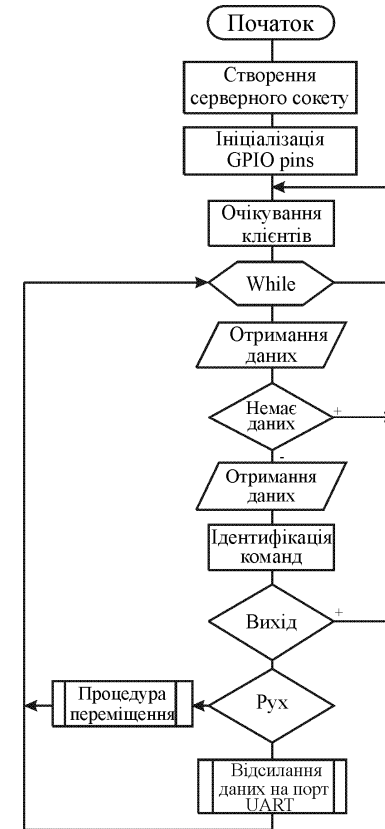


Рис 4. Блок-схема функціонування програми сервера

У разі підключення клієнта (`conn, addr = s.accept()`), можна отримувати дані з цього порту. `data = conn.recv(1024)` – отримання даних від клієнта. При отриманні команд від клієнта, які не відповідають за переміщення РТС, то серверна програма ідентифікує їх як такі, що потрібно виставити на порт UART, по якому вони відсилаються до плат управління баштою, зброєю чи світлом. Це можна зробити командою `ser.write(str)`.

При отриманні команд переміщення робототехнічної системи використовують процедури, які відповідають за ініціалізацію, відповідно, порту GPIO. А якщо команди не відповідають за переміщення РТС, то в програмі передбачено цикл, який приймає дані та формує з них керуючу стрічку для UART. Цей цикл представлено нижче.

```

mas=['frw','bck','lft','rgt','flt','blt','frr','brt','stp','ext']
...
elif not data in mas:
    new=int(data)
    data=chr(new)
    i=i+1
    str=str+data
    if i==6:
        print(new, str)
        print("Send to Uart")
        print(str)
        ser.write(str)
        i=0
        str=""
    
```

Після прийому команди перетворюємо її у цілий тип (отримали стрічку) і отримуємо по числу символ. Потрібно відсилати символи, щоб на процесорі виставити число у шістнадцятковій системі числення.

Після виконання циклу на UART порт сервер виставить стрічку з шести символів "ç\x04\x01\x05\x03\x00\r", яка відповідає 'E7 04 01 05 03 00 D' (використання спуску 1 при включеному денному світлі РТС). Програма сервера завантажується одразу після завантаження мікрокомп'ютера Raspberry Pi і виконується доти, поки включений мікрокомп'ютер з можливістю підключення/відключення клієнтів.

Особливості реалізації клієнтської частини ПЗ. Блок-схему алгоритму роботи програми клієнта зображено на рис. 5.



Рис 5. Блок-схема роботи клієнтської частини ПЗ

На основі алгоритму створено програму, задавши їй зручний інтерфейс для користувача (рис. 6) Інтерфейс програми поділено на частини:

- Управління платформою танка.
- Управління баштою РТС.

Використання зброї та налаштування параметрів світла (через меню програми).

В рядку стану програми відображається стан поточної команди, який змінюється при натисненні будь-якої кнопки.

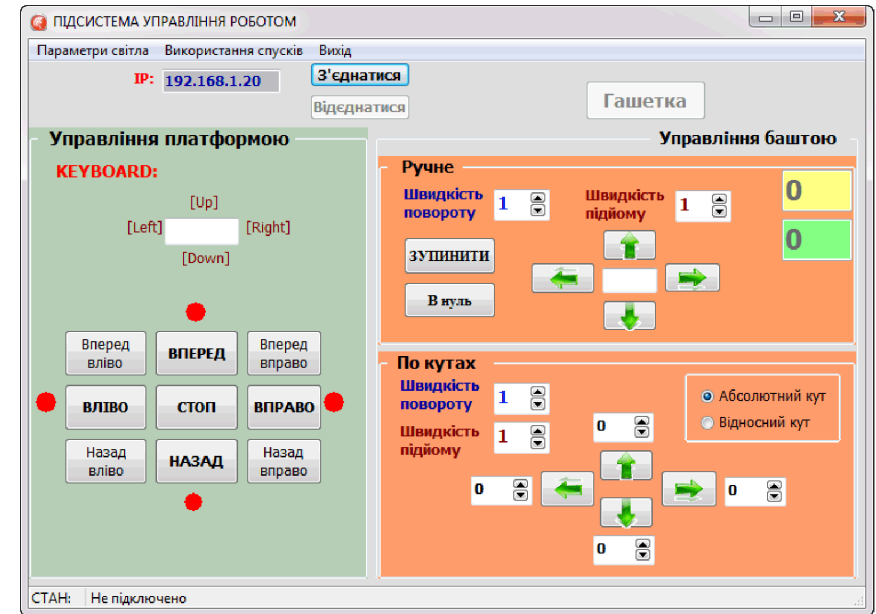


Рис. 6. Інтерфейс програми клієнта

При запуску програми, координати башти зчитуються з файлів, а при виході з програми координати перезаписуються у відповідні файли. Таким чином ми зберігаємо координати башти при виключенні програми. Потреба в координатах з'являється тоді, коли потрібно задати кут повороту башти.

Управління баштою представлено також двома варіантами: ручним і управлінням по координатах. Ручне управління баштою відбувається аналогічно до управління РТС з використанням клавіатури. При управлінні баштою по координатах у відповідних полях потрібно ввести кут повороту і натиснути напрям повороту башти.

Функцію управління РТС було реалізовано двома способами: за допомогою використання клавіш управління курсором (стрілки Up, Down, Left і Right), або ж використовуючи кнопки зліва внизу вікна програми. Інтерфейс програми для управління баштою також складається із панелі з кнопками, користувачеві пропонується здійснювати поворот башти вручну, аналогічно управлінню переміщення з використанням клавіатури.

На цей час представлена можливість перегляду відео з камер та відео реєстратора тільки за допомогою браузера Internet Explorer. У ньому достатньо прописати адресу 192.168.1.100 і відео буде відображатися в даному браузері. Як варіант, можна використати програму перегляду відео з відео реєстратора.

У розробленій програмі також реалізовано функції роботи зі світлом та спусковими механізмами (рис. 7. та 8).

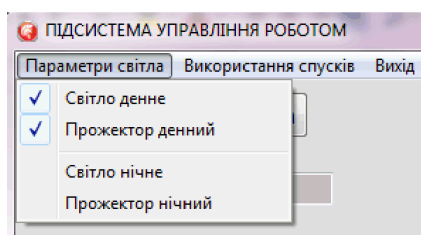


Рис. 7. Включення денного світла

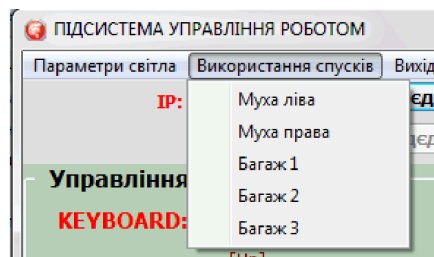


Рис. 8. Використання спускових механізмів

Висновок. Для реалізації моделі ПЗ використано клієнт-серверну архітектуру. Для кожної із програмних частин розроблено блок-схеми алгоритмів. Розроблено інтерфейс клієнтської частини програми.

Розроблене програмне забезпечення є багатофункціональною підсистемою віддаленого управління (управління переміщенням РТС, параметрами світла, обертанням башти та управління використанням зброї та спускових механізмів, відеоспостереження). Також це програмне забезпечення володіє принципом модульності, в подальшому його функціонал можна розширити.

Література

1. Tsmots I.G. Hardware and software tools for motion control of mobile robotic system / I.G. Tsmots, V.M. Teslyuk, I. Vavruk // In Proc. of the 12-th International Conference on The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM'2013, Lviv-Polyana, Ukraine, Feb. 19-23, 2013. – Pp. 368.
2. Баранів, Д.Н. Розроблення інтелектуальної системи управління мобільними роботами на основі стежачої системи технічного зору і нечіткої логіки : автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 12.06.08 "Менеджмент" / Д.Н. Баранів; [Ун-т "СТАНКИН"]. – М., 2008. – 222 с.
3. Афонин В.Л. Проектирование и компьютерное моделирование интеллектуальных технологических систем механической обработки / В.Л. Афонин, Т.Н. Зайченко // Доклады Томского государственного ун-та систем управления и радиоэлектроники. – 2011. – Вип. 2-3 (24). – С. 32-44.
4. Babiarz, Artur, et al. "Control System of Mobile Robot Group / Artur Babiarz, et al.// Innovative Control Systems for Tracked Vehicle Platforms. Springer International Publishing, 2014. – Pp. 3-18.
5. Цмоць І.Г. Моделювання інтелектуального управління рухом мобільної робототехнічної системи / І.Г. Цмоць, І.С. Ваврук, В.М. Теслюк // Науковий вісник НЛТУ України : зб. наук.-техн. праць. – Львів : РВВ НЛТУ України. – 2013. – Вип. 23.15. – С. 290-300.
6. Юревич Е.И. Основы робототехники / Е.И. Юревич. – Изд. 2-ое, [перераб. и доп.]. – СПб. : Изд-во БВХ-Петербург, 2005. – 416 с.
7. Комарницький М.В. Матвійчук К.В., Теслюк В.М. Структурна та фізична моделі для автоматизованого проектування підсистеми віддаленого управління мобільною робототехнічною системою / М.В. Комарницький, К.В. Матвійчук, В.М. Теслюк // Науковий вісник НЛТУ України : зб. наук.-техн. праць. – Львів : РВВ НЛТУ України. – 2014. – Вип. 24.9. – С. 156-161.
8. Chapter 3: Architectural Patterns and Styles. [Electronic resource]. – Mode of access <http://msdn.microsoft.com/en-us/library/ee658117.aspx>
9. Software architecture of mobile robotic systems: A case study Capucha Lubomir, Rohal-Ilkiv Boris / 2016 Cybernetics & Informatics, Feb. 2016. – Pp. 1-5.

10. [Electronic resource]. – Mode of access <http://openmicros.org/index.php/articles / 94 - ciseco - product - documentation/raspberry - pi/217 - getting - started - with - raspberry - pi - gpio - and - python>.

11. Denysyuk Pavlo. UML Models of Determining the Direction of Motion of Mobile Robotic System / Pavlo Denysyuk, Kateryna Matviichuk, Andriy Medvid // Комп'ютерні науки та інформаційні технології : матер. восьмої Міжнар. конф. CSIT'2013, 11-16 листоп. 2013, Львів, Україна, Нац. ун-т "Львів. політехніка". – Львів : Вид-во "Вежа і Ко", 2013. – С. 135-136.

Надійшла до редакції 31.08.2016 р.

Матвійчук Э.В., Теслюк В.Н., Зелинский А.Я. Программная модель подсистемы управления мобильной робототехнической системой

Описаны особенности разработки программного обеспечения подсистемы удаленного управления робототехнической системой (РТС). Построена UML диаграмма прецедентов, с помощью которой отражены основные задачи, которые будет выполнять РТС в процессе функционирования. Для реализации ПО выбрана клиент-серверная архитектура. Построены UML диаграммы деятельности для клиентской и серверной частей ПО, описаны принципы их взаимодействия и представлены блок-схемы алгоритмов их работы. Разработан и описан интерфейс, для клиентской части ПО, обоснован выбор языка программирования для реализации функционала системы.

Ключевые слова: робототехническая система, программное обеспечение, UML диаграмма, клиент-серверная архитектура.

Matviichuk K.V., Teslyuk V.M., Zelinsky A.Ya. Programming Model of Control Subsystem for Mobile Robotic Technical System

The article describes the features of the development of software for the subsystem of the remote control of robotic technical system (RTS). Use case UML diagram that reflects the main tasks that perform the RTS in the operation process was constructed. To implement the software, client-server architecture was chosen. UML activity diagrams for client and server parts of the software were built, the principles of their cooperation were described and flowcharts of algorithms of their work were presented. The user interface for the client part of software was developed and described; the choice of programming language for implementation of a functional system was proved.

Keywords: robotic technical system, software, UML-diagram, client-server architecture.

UDK 697.1:699.865

WPLYW SPOSOBU ZASILANIA BUDYNKU W ENERGIE NA WSKAZNIK ROCZNEGO ZAPOTRZEBOWANIA NA NIEODNAWIALNA ENERGIĘ PIERWOTNA

Myroslav Malovanyy¹, Anna Życzyńska², Grzegorz Dys³

Omówiono wpływ wykorzystania odnawialnych źródeł energii na wartość wskaźnika rocznego zapotrzebowania na energię końcową i nieodnawialną energię pierwotną na przykładzie budynków użyteczności publicznej. Przeanalizowano kilka wariantów rozwiązań, technicznie możliwych do realizacji, wykorzystujących nieodnawialne i odnawialne źródła energii dla zasilania budynków w energię na cele ogrzewania, przygotowania ciepłej wody i oświetlenia. Przedstawiono strukturę zapotrzebowania na energię na poszczególne cele oraz relacje pomiędzy wskaźnikami zapotrzebowania na energię użytkową, końcową i nieodnawialną pierwotną. Uzyskane wyniki porównano do standardu energetycznego obowiązującego w

¹ Kierownik Katedry EZWZN, Myroslav Malovanyy, dr.hab. – Politechnika Lwowska;

² Prodziejkan WBiA, Anna Życzyńska, dr inż. – Politechnika Lubelska;

³ Właściciel, Grzegorz Dys, mgr inż – Energetyczna Pracownia Inżynierska ERG S.C.