

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова  
праця на правах рукопису

**Гладкова Ольга Миколаївна**

УДК 004.896: 621.3.049.779

**ДИСЕРТАЦІЯ**

**АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ  
ЗАСОБАМИ ВІДДАЛЕНОЇ ІНЖЕНЕРІЇ**

05.13.12 – Системи автоматизації проектувальних робіт

05 «Технічні науки»

Подається на здобуття наукового ступеня  
кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне джерело

\_\_\_\_\_ О.М. Гладкова

(підпис, ініціали та прізвище здобувача)

Науковий керівник –  
Пархоменко Анжеліка Володимирівна,  
к.т.н., доцент

***Ідентичність всіх примірників дисертації***

***ЗАСВІДЧУЮ:***

*Вчений секретар спеціалізованої  
вченої ради*

***/Р.А. Бунь/***

Запоріжжя – 2018

## АНОТАЦІЯ

Гладкова О.М. Автоматизація проектування вбудованих систем засобами віддаленої інженерії. – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.12 – системи автоматизації проектувальних робіт. – Національний університет «Львівська політехніка» Міністерства освіти і науки України, Львів, 2018.

У дисертаційній роботі розв’язано наукове завдання розробки моделей, методів та інструментальних засобів віддаленої інженерії для підвищення ефективності автоматизованого проектування вбудованих систем. Актуальність дослідження обумовлена необхідністю розвитку сучасних методів автоматизованого проектування на тлі зростаючих вимог щодо скорочення термінів проектування вбудованих систем за умови забезпечення їх якості.

Тому, в роботі отримав подальший розвиток метод паралельного проектування вбудованих систем, що відрізняється спільним застосуванням паралельного та платформно-орієнтованого підходів, методології повторного використання, а також засобів віддаленої інженерії, що дає змогу зменшити час переходу між системним та функціонально-логічним рівнями при проектуванні вбудованих систем та підвищити рівень автоматизації проектувальних робіт шляхом автоматизованого вибору апаратно-програмної платформи.

Вперше розроблено метод автоматизованого проектування вбудованих систем з використанням технологій віддаленої інженерії, що дає змогу організувати спільне використання обладнання та програмного забезпечення (без необхідності їх придбання та налаштування) для прототипування проектованої вбудованої системи та, таким чином, знизити собівартість проектованої продукції.

Удосконалено модель формування вимог до вбудованих систем та метод роботи з ними, які на відміну від існуючих, враховують структурні особливості вбудованих систем, що містять апаратну та програмну частини, дозволяють розподіляти вимоги між програмними та апаратними компонентами вбудованої

системи при автоматизованому проектуванні та, таким чином, більш повно враховувати вимоги до вбудованих систем та виконувати необхідні процеси роботи з вимогами при автоматизованому проектуванні.

Отримав подальший розвиток метод формування рекомендацій на основі знань, який, на відміну від існуючого, поєднує в собі методи обмежень та подібних об'єктів, використовує метод багатокритеріального аналізу, що дає змогу надавати рекомендації навіть при суперечливості вимог користувача.

На основі розроблених методів, моделей та алгоритмів реалізовано програмне, технічне, інформаційне та методичне забезпечення для автоматизації проектування вбудованих систем.

Створений програмно-технічний комплекс автоматизації проектування вбудованих систем засобами віддаленої інженерії використовувався при автоматизованому проектуванні вбудованої системи управління рухомими об'єктами. Це дозволило зменшити витрати часу на розробку на 10-13% та, таким чином, підвищити продуктивність праці проектувальників. Отже, розв'язано завдання підвищення ефективності автоматизованого проектування вбудованих систем за рахунок скорочення термінів проектування, зниження витрат на проектування та підвищення рівня автоматизації проектувальних робіт. Це було досягнуто завдяки реалізації технологій віддаленої інженерії на основі:

- віддаленої лабораторії для організації спільного використання обладнання та програмного забезпечення, а також швидкого прототипування проектованої системи;
- рекомендаційної системи для автоматизованого вибору апаратно-програмної платформи при переході між системним та функціонально-логічним рівнями при автоматизованому проектуванні вбудованих систем.

В першому розділі виконано аналіз сучасного процесу автоматизованого проектування вбудованих систем (ВС) з урахуванням їх структурних та функціональних особливостей, а також досліджено шляхи підвищення рівня автоматизації проектувальних робіт. Показано, що автоматизоване

проектування вбудованих систем дистанційного управління рухомими об'єктами, які надають доступ людині до небезпечних та важкодоступних місць, є актуальним та економічно обґрунтованим. Але, розробка ВС «з нуля» з використанням традиційного підходу до проектування сьогодні не є ефективною. Саме тому, доцільним є розвиток паралельного та платформно-орієнтованого підходів до проектування, а також застосовування методології повторного використання, що дозволить прискорити розробку проекту та скоротити час виходу продукції на ринок. Показано, що використання готових апаратно-програмних платформ на основі мікроконтролерів сприяє скороченню часу проектування ВС, надає можливості повторного використання в інших проектах, але вимагає значних часових витрат на пошук відповідної завданням проекту платформи. Тому, автоматизація даного виду проектних робіт є актуальною. Виконано огляд програмних засобів, що використовуються для автоматизації проектування ВС та показано переваги застосування віддаленого інструментарію для розширення функціональних можливостей існуючих десктопних програмних систем.

У другому розділі запропоновано удосконалену модель формування вимог до вбудованих систем та метод роботи з вимогами, які дозволяють більш повно враховувати комплекс вимог до ВС та підвищити ефективність виконання необхідних процесів роботи з вимогами, а саме, визначення, аналіз і формування вимог до програмних і апаратних компонентів ВС при автоматизованому проектуванні. Удосконалено метод проектування ВС на основі паралельного та платформно-орієнтованого підходів до проектування, методології повторного використання апаратних та програмних компонентів, а також засобів віддаленої інженерії. Розроблено метод автоматизованого проектування вбудованих систем на основі технологій віддаленої інженерії, що дозволяє організувати спільне використання обладнання та програмного забезпечення для прототипування проектованої ВС та, таким чином, знизити собівартість проектованої продукції.

В третьому розділі обґрунтовано вибір рекомендаційного методу, на

основі якого будуть формуватися рекомендації щодо апаратно-програмних платформ при автоматизованому проектуванні ВС. З метою надання рекомендацій проектувальнику щодо апаратно-програмних платформ при автоматизованому проектуванні ВС було обрано метод формування рекомендацій на основі знань, оскільки він працює з нагальними вимогами користувача, якими в даному випадку виступають вимоги до проектованої вбудованої системи. Розроблено модель представлення знань, що побудована як семантична мережа, відображає інформацію про апаратно-програмні платформи та може бути ефективно використана при розробці бази знань апаратно-програмних платформ рекомендаційної системи. Запропоновано метод формування рекомендацій на основі знань, в основу якого покладено міру подібності об'єктів, яка базується на знаходженні близьких об'єктів за числовими значеннями, а також метод обмежень, який знаходить точну відповідність між об'єктами. Крім того, для надання рекомендацій у ситуаціях, коли одна вимога виключає іншу, застосовується метод багатокритеріального аналізу для вибору найкращого рішення за кожною з цих вимог.

В четвертому розділі, на основі запропонованих методів та моделей розроблено архітектуру та структурну схему програмно-технічного комплексу (ПТК) для автоматизації проектування ВС, що дозволяють реалізувати необхідні функціональні можливості ПТК та організувати ефективне використання засобів віддаленої інженерії. Описано особливості реалізації програмних модулів і бази знань ПТК. Реалізовано програмне та технічне забезпечення віддаленої лабораторії, що надає користувачам можливість спільного використання обладнання та програмного забезпечення для створення та дослідження прототипу проектованої ВС. Таким чином можуть бути зменшені матеріальні та часові витрати на розробку та, відповідно, собівартість проектованої продукції. Розроблено програмне та інформаційне забезпечення рекомендаційної системи, яка надає підтримку проектувальнику у вигляді рекомендації щодо використання апаратно-програмних платформ

при автоматизованому проектуванні ВС, виходячи з вимог до проектованої вбудованої системи.

Для ефективного впровадження ПТК в процес автоматизованого проектування ВС розроблено методику, яка охоплює наступні проектні етапи: розробка специфікації; розробка архітектури системи; вибір апаратно-програмних платформ на основі рекомендаційних методів; розробка програмного забезпечення; створення та дослідження прототипу системи на основі віддаленої лабораторії. Ця методика може використовуватись в практиці інженерного проектування для ефективного впровадження засобів віддаленої інженерії. Наведено приклад практичного застосування ПТК при автоматизованому проектуванні ВС управління рухомими об'єктами (ВС УРО). Показано, що використання розробленого ПТК дозволило зменшити витрати часу на розробку прототипу ВС УРО на 10-13% за рахунок скорочення часу пошуку відповідної вимогам проекту апаратно-програмної платформи за допомогою рекомендаційної системи, а також швидкого прототипування та дослідження прототипу системи з використанням віддаленої лабораторії. Це сприяло підвищенню продуктивності праці проектувальників та, відповідно, зменшенню собівартості проекту.

*Ключові слова:* вбудована система, віддалена інженерія, паралельне проектування, платформно-орієнтоване проектування, апаратно-програмна платформа, віддалена лабораторія, рекомендаційна система, база знань.

Список публікацій здобувача:

1. Parkhomenko A., Gladkova O. Complex requirements analysis for the high-level design of Embedded Systems. *Вісник НУ «Львівська політехніка». Серія «Комп'ютерні системи проектування. Теорія і практика»*. 2014. № 808. С. 3–9.
2. Гладкова О.М., Пархоменко А.В. Дослідження та практична реалізація рекомендаційної системи для вибору апаратно-програмних платформ при автоматизованому проектуванні вбудованих систем. *Наукові праці*

ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка». 2017. № 2(25). С. 22–31.

3. Дослідження та розробка автоматизованої системи віддаленого керування групою рухомих об'єктів / А.В. Пархоменко, О.М. Гладкова, О.П. Кравченко, Д.П. Кравченко. *Вісник СНУ ім. В. Даля*. 2017. № 8(238). С. 67–74.
4. Implementation of reusable solutions for remote laboratory development / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Y. Zalyubovskiy. *International Journal of Online Engineering*. 2016. Vol. 12(7). P. 24–29. (SCOPUS, Web of Science)
5. Development and application of remote laboratory for embedded systems design / A. Parkhomenko, O. Gladkova, E. Ivanov, A. Sokolyanskii, S. Kurson. *International Journal of Online Engineering*. 2015. Vol.11(3). P.27–31. (SCOPUS, Web of Science)
6. Internet-based technologies for design of embedded systems / A. Parkhomenko, O. Gladkova, S. Kurson, A. Sokolyanskii, E. Ivanov. *Journal of Control Science and Engineering*. 2015. Vol. 3(2). P. 55–63. (ProQuest, CrossRef)
7. Система автоматизованого керування групою рухомих об'єктів / А.В. Пархоменко, О.М. Гладкова, О.П. Кравченко : пат.123942 Україна : МПК G05B 19/05. № u2017110387 ; заявл. 27.10.2017 ; опубл. 12.03.2018, Бюл. №5.
8. Свідоцтво № 64253 Україна. Комп'ютерна програма “Віддалена лабораторія проектування вбудованих систем” / А.В. Пархоменко, О.М. Гладкова, О.В. Соколянський; Власник ЗНТУ; дата реєстрації в ДСІВ України 26.02.2016 р.
9. Свідоцтво № 67404 Україна. Комп'ютерна програма “Автоматизована система управління рухомим об'єктом” / А.В. Пархоменко, О.М. Гладкова; власник ЗНТУ; дата реєстрації в ДСІВ України 22.08.2016 р.
10. Challenges and solutions for mobile object control system / D. Kravchenko, O. Kravchenko, A. Parkhomenko, O. Gladkova. *Intelligent data acquisition and*

- advanced computing systems: technology and applications: proceedings of 10th IEEE international conference, 21-23 September, 2017. Bucharest (Romania), 2017. P.988–993.*
11. Path finding algorithm for moving robots and obstacles avoidance / S. Boeckx, P. Pelgrims, A. Parkhomenko, O. Gladkova, D.V. Merode. *Ambient intelligence and embedded systems: proceedings of the international symposium, 14-16 September, 2017. Vaasa (Finland). URL: <http://amies-2017.international-symposium.org/proceedings.html>*
  12. Modernization of mobile object control system based on Raspberry Pi and Arduino platforms / A. Parkhomenko, O. Kravchenko, D. Kravchenko, O. Gladkova. *Embedded systems and trends in teaching engineering: proceedings of the international symposium, 12-15 September, 2016. Nitra (Slovakia): Constantine the Philosopher University, 2016. P. 249–253.*
  13. Reusable solutions for embedded systems' design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Ya. Zalyubovskiy. *Remote engineering and virtual instrumentation: proceedings of 13th international conference, 24-26 February, 2016. Madrid (Spain), 2016. P. 313–317.*
  14. Investigation of reuse concepts for embedded systems design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, Ya. Zalyubovskiy. *Perspective technologies and methods in MEMS design: proceedings of XII international conference, 20-24 April, 2016. Lviv: NU "Lviv Polytechnic", 2016. P. 78–80.*
  15. Гладкова О.М. Підходи та особливості проектування RESTful API. *Тиждень науки – 2016 : тези доп. щоріч. наук.-практ. конф. викладачів, науковців, молодих учених, аспірантів, студентів ЗНТУ, 18–22 квітня 2016 р. Запоріжжя : ЗНТУ, 2016. С. 562–564.*
  16. Investigation of remote lab design technologies / A. Parkhomenko, A. Sokolyanskii, O. Gladkova, S. Kurson. *Perspective technologies and methods in MEMS design: proceedings of XI international conference, September 2-6, 2015. Lviv: NU "Lviv Polytechnic", 2015. P. 92–95.*
  17. Пархоменко А., Гладкова О. Інтерактивна віддалена лабораторія



- дослідження апаратно-програмних платформ. *Інтернет-Освіта-Наука-2014* : тези доповіді міжнародної науково-практичної конференції, 14-17 жовтня 2014 р. Вінниця : ВНТУ, 2014. С. 111–113.
18. Пархоменко А.В., Гладкова О.Н., Иванов Е.В. Исследование особенностей создания и применения лаборатории удаленного доступа при решении задач проектирования встроенных систем. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : тези доповіді VII міжнародної науково-практичної конференції, 17-19 вересня 2014 р. Запоріжжя : ЗНТУ, 2014. С. 205–206.
19. Parkhomenko A.V., Gladkova O.N. Virtual tools and collaborative working environment in embedded system design. *Remote engineering and virtual instrumentation: proceedings of XI international conference*, 26-28 February, 2014. Porto (Portugal), 2014. P. 91–93.
20. Parkhomenko A., Gladkova O. Analysis and application of existent approaches in microcontroller system designing. *Perspective technologies and methods in MEMS design: proceedings of IX international conference*, 16-20 April, 2013. Lviv: NU “Lviv Polytechnic”, 2013. P. 59–61.
21. Parkhomenko A., Gladkova O. Investigation of peculiarities of analysis of system and software requirements for designing automated system. *The experience of designing and application of CAD systems in microelectronics: proceedings of XII international conference*, 19-23 February, 2013. Lviv: NU “Lviv Polytechnic”, 2013. P.268–270.
22. Исследование современных технологий автоматизированного проектирования электронных устройств на микроконтроллерах / О.А. Поздняков, О.Н. Гладкова, А.В. Пархоменко, О.В. Цветков. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій* : тези доповіді VI міжнародної науково-практичної конференції, 19-21 вересня 2012 р. Запоріжжя : ЗНТУ, 2012. С. 248–250.

## ABSTRACT

Gladkova O.M. The computer aided design of embedded systems by the means of remote engineering tools. – On the rights of manuscript.

Dissertation for scientific degree of Candidate of Technical Sciences. Specialty 05.13.12 – computer aided design systems. – Lviv Polytechnic National University of Ministry of Education and Science of Ukraine, Lviv, 2018.

The scientific task of models, methods and remote engineering tools development for the increasing of efficiency of embedded systems computer aided design was solved in the dissertation work. The study is relevant due to the necessity of modern methods of computer aided design development against the backdrop of growing requirements for reducing the design time of embedded systems while ensuring their quality.

Therefore, the method of embedded systems parallel design was developed, based on the joint application of parallel and platform-oriented approaches, reuse methodology and remote engineering tools. It allows to reduce the time of transition between the system and the functional-logical levels of embedded systems design and to increase the level of design works automation at the expense of hardware-software platform automated selection.

For the first time, the method of computer aided design of embedded systems based on the remote engineering technologies was proposed. It allows the sharing of the equipment and software (without the necessity of their purchase and configuration) for embedded system prototyping and thus to reduce the cost of projected products.

The model of requirements for embedded systems forming and the method of working with them were improved, which, in contrast to the existing ones, take into account the structural features of the embedded systems that contain hardware and software parts. They allow to distribute requirements between the software and hardware components of the embedded system and, thus, to take into account the requirements to the ES more fully and to carry out the necessary processes of working with the requirements in computer aided designing.

The method of recommendations forming on the basis of knowledge was

developed, which, in contrast to the existing one, combines the methods of restrictions and similar objects, uses the method of multicriteria analysis, which makes it possible to give recommendations even in case of user requirements conflict.

The software, hardware tools as well as the informational and methodical support for embedded systems computer aided design on the basis of the developed methods, models and algorithms were implemented.

Created software and hardware complex for computer aided design of embedded systems by means of remote engineering tools was used for computer aided design of the ES of moving objects control. This allowed a drop of the development time by 10-13% and thus to rise the productivity of designers' work. So, the task of increasing of the efficiency of computer aided design of embedded systems by reducing the design time and the cost of design, as well as the increasing of the level of design work automation was solved. This was achieved through the implementation of remote engineering technologies based on:

- the remote laboratory for the sharing of equipment and software, as well as rapid prototyping of the projected system;
- the recommendation system for the automated selection of the hardware-software platform in the transition between the system and the functional-logical levels in the computer aided designing of the embedded systems.

In the first section the analysis of the modern process of computer aided design of the embedded systems (ES) was performed taking into account their structural and functional features, as well as the ways of increasing the level of automation of designing works. It was shown that the computer aided design of embedded systems of mobile objects' remote control, that provide access to dangerous and hard-to-reach places for people, is relevant and economically feasible. However, the development of the ES "from scratch" using the traditional approach in designing is not effective today. That is why, it is expedient to develop parallel and platform-based approaches in designing, as well as to apply the reuse methodology, which will accelerate the development of the project and reduce the time of products' release to the market. It is shown, that the usage of ready-made hardware-software platforms based on

microcontrollers contributes to shortening the design time of the ES, provides opportunities for reuse in other projects, but it requires considerable time for searching the corresponding platform. Therefore, the automation of this type of design work is relevant. An overview of the software tools used in computer aided designing of the ES was performed and the advantages of remote tools usage for expanding the functionality of existing desktop software systems were shown.

In the second section the improved model for requirements' forming for embedded systems and the method of work with requirements were proposed. They allow to take into account the complex of requirements of the ES more fully and to improve the efficiency of the implementation of the required processes of work with requirements, as follows, definition, analysis and formation of requirements for ES software and hardware components in computer aided designing. The method of ES designing was improved on the basis of parallel and platform-oriented approaches in design, as well as the methodology of reuse of hardware and software components and remote engineering tools. The method of computer aided design of embedded systems on the basis of remote engineering technologies was developed. It allows sharing of equipment and software for prototyping of projected ES, and thus to drop the cost of projected products.

In the third section the choice of the recommendation method was justified, on the basis of which recommendations for hardware and software platforms in the ES computer aided designing will be formed. In order to provide recommendations to the designer on hardware and software platforms in the computer aided designing of the ES, the knowledge-based method of recommendations forming was chosen as it works with the urgent requirements of the user. In this case they are the requirements to the projected embedded system. A model of knowledge representation, that is built as a semantic network, was developed. It displays the information about hardware and software platforms and it can be effectively used in the development of the knowledge base of hardware and software platforms of the recommendation system. The method of formation of recommendations on the basis of knowledge was proposed. It is based on the measure of similarity of objects, which is based on the finding of close objects

by numerical values, as well as on the method of constraints that finds an exact match between objects. In addition, to provide guidance in situations where one requirement excludes another, a multicriteria analysis method was used to select the best solution for each of these requirements.

In the fourth section, on the basis of the proposed methods and models, an architecture and a structural scheme of the software and hardware complex for the computer aided design of the ES have been developed. They enable the implementation of the necessary functional capabilities of the software and hardware complex and organize the efficient usage of remote engineering tools. The peculiarities of software modules and knowledge base of software and hardware complex implementation were described. The software and hardware of the remote laboratory were implemented, which provide users with the sharing equipment and software for creation and studying the prototype of projected ES. In such way, the material and time costs for the development and, accordingly, the cost of projected products may be reduced. The software and information support of the recommendation system were developed. They provide support to the designer in the form of recommendations on the usage of hardware and software platforms in the ES computer aided designing, based on the requirements for the projected embedded system.

For the effective implementation of the software-hardware complex in the process of ES computer aided design, a methodology has been developed. It covers the following project stages: the development of the specification; the development of system architecture; the selection of hardware and software platforms based on recommendation methods; software development; creation and studying of a system prototype based on the remote laboratory usage. This methodology can be used in the practice of engineering design for the effective implementation of the remote engineering tools. An example of the practical application of software-hardware complex in computer aided designing of ES for moving object control was presented. It was shown, that the usage of the developed software and hardware complex allowed ta drop of the development time of the prototype of the ES by 10-13%. It was achieved with reducing the search time of the corresponding to the project requirements

hardware and software platform with the help of the recommendation system, as well as rapid prototyping and studying of the system's prototype using remote lab. Therefore, the productivity of designers' work was increased and, accordingly, the cost of the project was reduced.

*Key words:* embedded system, remote engineering, co-design, platform-oriented design, hardware-software platform, remote laboratory, recommendation system, and knowledge-database.

The list of author's publications:

1. Parkhomenko A., Gladkova O. Complex requirements analysis for the high-level design of Embedded Systems. *Visnyk of Lviv Polytechnic National University. Series «Computer systems design. Theory and practice»*. 2014. № 808. P. 3–9.
2. Gladkova O.M., Parkhomenko A.V. Research and practical implementation of the recommendation system for the choice of hardware-software platforms in the automated design of embedded systems. *Scientific works of DonNTU. Series "Informatics, Cybernetics and Computing"*. 2017. № 2(25). P. 22–31.
3. Investigation and development of automated remote control system of a group of mobile objects / A.V. Parkhomenko, O.M. Gladkova, O.P. Kravchenko, D.P. Kravchenko. *Visnyk of V. Dahl EUNU*. 2017. № 8(238). P. 67–74.
4. Implementation of reusable solutions for remote laboratory development / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Y. Zalyubovskiy. *International Journal of Online Engineering*. 2016. Vol. 12(7). P. 24–29. (SCOPUS, Web of Science)
5. Development and application of remote laboratory for embedded systems design / A. Parkhomenko, O. Gladkova, E. Ivanov, A. Sokolyanskii, S. Kurson. *International Journal of Online Engineering*. 2015. Vol.11(3). P.27–31. (SCOPUS, Web of Science)
6. Internet-based technologies for design of embedded systems / A. Parkhomenko, O. Gladkova, S. Kurson, A. Sokolyanskii, E. Ivanov. *Journal of Control Science and Engineering*. 2015. Vol. 3(2). P. 55–63. (ProQuest, CrossRef)

7. Automated control system of a group of mobile objects / A.V. Parkhomenko, O.M. Gladkova, O.P. Kravchenko: patent.123942 Ukraine: IPC G05B 19/05. № u201710387; declared 27.10.2017; publ. 12.03.2018, Bull. №5.
8. Certificate № 64253 Ukraine. Computer program «Remote laboratory of embedded systems design» / A.V. Parkhomenko, O.M. Gladkova, O.V. Sokolyanskii; owner is ZNTU; date of registration in SSIP of Ukraine 26.02.2016.
9. Certificate № 67404 Ukraine. Computer program «Automated control system for mobile object» / A.V. Parkhomenko, O.M. Gladkova; owner is ZNTU; date of registration in SSIP of Ukraine 22.08.2016.
10. Challenges and solutions for mobile object control system / D. Kravchenko, O. Kravchenko, A. Parkhomenko, O. Gladkova. *Intelligent data acquisition and advanced computing systems: technology and applications*: proceedings of 10th IEEE international conference, 21-23 September, 2017. Bucharest (Romania), 2017. P.988–993.
11. Path finding algorithm for moving robots and obstacles avoidance / S. Boeckx, P. Pelgrims, A. Parkhomenko, O. Gladkova, D.V. Merode. *Ambient intelligence and embedded systems*: proceedings of the international symposium, 14-16 September, 2017. Vaasa (Finland). URL: <http://amies-2017.international-symposium.org/proceedings.html>
12. Modernization of mobile object control system based on Raspberry Pi and Arduino platforms / A. Parkhomenko, O. Kravchenko, D. Kravchenko, O. Gladkova. *Embedded systems and trends in teaching engineering*: proceedings of the international symposium, 12-15 September, 2016. Nitra (Slovakia): Constantine the Philosopher University, 2016. P. 249–253.
13. Reusable solutions for embedded systems' design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Ya. Zalyubovskiy. *Remote engineering and virtual instrumentation*: proceedings of 13th international conference, 24-26 February, 2016. Madrid (Spain), 2016. P. 313–317.
14. Investigation of reuse concepts for embedded systems design / A. Parkhomenko,

- O. Gladkova, A. Sokolyanskii, Ya. Zalyubovskiy. *Perspective technologies and methods in MEMS design*: proceedings of XII international conference, 20-24 April, 2016. Lviv: NU “Lviv Polytechnic”, 2016. P. 78–80.
15. Gladkova O.M. Approaches and design features RESTful API. *The Science Week - 2016*: thesis of add. annual science-practice conf. for the teachers, scientists, young scientists, post-graduate students, students of ZNTU, 18–22 April 2016. Zaporizhzhia: ZNTU, 2016. P. 562–564.
  16. Investigation of remote lab design technologies / A. Parkhomenko, A. Sokolyanskii, O. Gladkova, S. Kurson. *Perspective technologies and methods in MEMS design*: proceedings of XI international conference, September 2-6, 2015. Lviv: NU “Lviv Polytechnic”, 2015. P. 92–95.
  17. Parkhomenko A., Gladkova O. Interactive remote laboratory for hardware-software platforms research. *Internet-Education-Science-2014*: proceedings of the international scientific and practical conference, 14-17 October 2014. Vinnytsya: VNTU, 2014. P. 111–113.
  18. Parkhomenko A.V., Gladkova O.N., Ivanov E.V. Investigation the features of creating and using a remote access lab to solve the problems of embedded systems design. *Modern problems and achievements in the field of radio engineering, telecommunications and information technologies*: proceedings of the VII International Scientific and Practical Conference, 17-19 September 2014. Zaporizhzhia: ZNTU, 2014. P. 205–206.
  19. Parkhomenko A.V., Gladkova O.N. Virtual tools and collaborative working environment in embedded system design. *Remote engineering and virtual instrumentation*: proceedings of XI international conference, 26-28 February, 2014. Porto (Portugal), 2014. P. 91–93.
  20. Parkhomenko A., Gladkova O. Analysis and application of existent approaches in microcontroller system designing. *Perspective technologies and methods in MEMS design*: proceedings of IX international conference, 16-20 April, 2013. Lviv: NU “Lviv Polytechnic”, 2013. P. 59–61.
  21. Parkhomenko A., Gladkova O. Investigation of peculiarities of analysis of system



and software requirements for designing automated system. *The experience of designing and application of CAD systems in microelectronics: proceedings of XII international conference*, 19-23 February, 2013. Lviv: NU “Lviv Polytechnic”, 2013. P.268–270.

22. Investigation of modern technologies of computer-aided design of electronic devices on microcontrollers / O.A. Pozdnyakov, O.N. Gladkova, A.V. Parkhomenko, O.V. Tsvetkov. *Modern problems and achievements in the field of radio engineering, telecommunications and information technologies: proceedings of the VI International Scientific and Practical Conference*, 19-21 September 2012. Zaporizhzhia: ZNTU, 2012. P. 248–250.

## ЗМІСТ

ЗМІСТ .....	18
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	21
ВСТУП .....	22
<b>РОЗДІЛ 1. ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ПРОЦЕСУ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ...</b>	<b>31</b>
1.1 Місце вбудованих систем управління рухомими об'єктами в загальній класифікації вбудованих систем .....	31
1.2 Аналіз особливостей вбудованих систем та технологій їх реалізації .	38
1.3 Дослідження існуючих підходів до проектування вбудованих систем .....	45
1.4 Дослідження процесів роботи з вимогами при автоматизованому проектуванні вбудованих систем .....	48
1.5 Програмні засоби для автоматизації проектування вбудованих систем .....	53
1.6 Висновки до розділу 1 .....	59
<b>РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ ЗАСОБАМИ ВІДДАЛЕНОЇ ІНЖЕНЕРІЇ.....</b>	<b>61</b>
2.1 Модель формування вимог до вбудованої системи .....	61
2.2 Метод роботи з вимогами .....	64
2.3 Методи автоматизованого проектування вбудованих систем на основі технологій віддаленої інженерії .....	67
2.4 Висновки до розділу 2 .....	73
<b>РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ПРИ АВТОМАТИЗОВАНОМУ ПРОЕКТУВАННІ ВБУДОВАНИХ СИСТЕМ .....</b>	<b>75</b>
3.1 Обґрунтування вибору рекомендаційного методу .....	75

3.2	Модель представлення знань щодо апаратно-програмних платформ для рекомендаційної системи .....	78
3.3	Математичний метод визначення подібності елементів .....	81
3.4	Метод формування рекомендацій щодо апаратно-програмних платформ .....	84
3.5	Висновки до розділу 3 .....	89
<b>РОЗДІЛ 4. РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНО-ТЕХНІЧНОГО КОМПЛЕКСУ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ .....</b>		
<b>91</b>		
4.1	Функціональні можливості та архітектура програмно-технічного комплексу.....	91
4.2	Опис особливостей розробки програмно-технічного комплексу .....	95
4.2.1	Основні рішення щодо реалізації віддаленої лабораторії .....	95
4.2.2	Основні рішення щодо реалізації рекомендаційної системи .....	101
4.3	Практичне застосування розробленого комплексу програмних та технічних засобів для автоматизованого проектування вбудованої системи управління рухомими об'єктами .....	109
4.3.1	Методика застосування програмно-технічного комплексу при автоматизованому проектуванні вбудованих систем управління рухомими об'єктами.....	109
4.3.2	Розробка специфікації вимог .....	112
4.3.3	Розробка архітектури системи.....	115
4.3.4	Вибір апаратно-програмної платформи для реалізації прототипу центрального блоку керування.....	119
4.3.5	Дослідження варіантів реалізації прототипу вбудованої системи управління рухомими об'єктами.....	120
4.4	Висновки до розділу 4 .....	126
<b>ЗАГАЛЬНІ ВИСНОВКИ .....</b>		
<b>128</b>		
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>		
<b>130</b>		

Додаток А. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації.....	142
Додаток Б. Акти впровадження результатів дисертаційної роботи.....	147
Додаток В. Свідоцтва про реєстрацію авторського права на твір (комп'ютерну програму) та патент на корисну модель .....	151

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- АЗ – апаратне забезпечення
- АП – автоматизоване проектування
- БЗ – база знань
- ВІ – віддалена інженерія
- ВЛ – віддалена лабораторія
- ВС – вбудована система
- ВС УРО – вбудована система управління рухомими об'єктами
- ЕКА – елементи конфігурації апаратури
- ЕКПЗ – елементи конфігурації програмного забезпечення
- КлФ – колаборативна фільтрація
- МК – мікроконтролер
- НВІС – надвелика інтегральна схема
- ПЗ – програмне забезпечення
- ПЛІС – програмована логічна інтегральна схема
- ПЛК – програмований логічний контролер
- ПТК – програмно-технічний комплекс
- РС – рекомендаційна система
- САПР – система автоматизованого проектування
- ЦБК – центральний блок керування
- API – Application program interface
- ECAD – electronic computer-aided design (САПР електронних пристроїв)
- MCAD – mechanical computer aided design (САПР механічних пристроїв)
- MVC – Model–View–Controller (Модель-Представлення-Контролер)
- RELDES – Remote Laboratory for Design of Embedded Systems

## ВСТУП

**Актуальність теми.** Світовий ринок вбудованих систем (ВС) постійно зростає, про що свідчать дослідження, проведені різними консалтинговими компаніями в останні роки [1-4]. З кожним роком ВС стають дедалі складнішими, а вимоги до забезпечення якості та скорочення термінів виходу на ринок – більш жорсткими. В цій роботі досліджується одна з категорій вбудованих систем управління рухомими об'єктами (ВС УРО), а саме система управління наземними колісними рухомими об'єктами. Наразі такі системи використовуються у різних галузях економіки (промисловість, транспорт, оборонна галузь, сільське господарство та сфера розваг), тому автоматизація їх проектування є актуальною.

При автоматизованому проектуванні (АП) ВС УРО необхідно враховувати наявність апаратної та програмної частин системи, а також велику кількість обмежень і вимог щодо розмірів, споживання енергії, завадостійкості та ін. Визначення, аналіз і формування вимог до програмних і апаратних компонентів ВС на етапі високорівневого (системний рівень) проектування є важливим фактором успішної реалізації ВС. На сьогодні існують стандарти (ГОСТ 34.602-89, ГОСТ Р 41904-2002, ISO 15288, ISO 12207, IEEE 830, IEEE1233), в яких наведено узагальнений перелік вимог до створюваної системи в цілому та, зокрема, до програмного забезпечення, призначення вимог та види діяльності по роботі з ними. Вагомий внесок у розвиток теорії та практики інженерії вимог до програмного забезпечення зробили: Wiegers K. [5] – запропонував схему класифікації вимог, ітеративний процес та прийоми формування вимог, розробив модель умовних ваг; Leffingwell D. [6] – розробив трирівневу модель подання вимог та підхід до виявлення системних обмежень; Cockburn A. [7] – запропонував концепцію створення варіантів використання проектованої системи. Результати цих досліджень можуть використовуватись в якості основи для розробки моделі формування вимог для ВС з урахуванням їх специфічних особливостей. При

цьому вимоги до апаратного забезпечення, як важливої складової ВС, також мають бути враховані.

Існуючі на сьогодні стандарти, що описують життєвий цикл проєктованих систем, не враховують особливостей автоматизованого проєктування ВС, які містять взаємопов'язані програмні та апаратні компоненти. Традиційний підхід [8] при проєктуванні ВС спричиняє ризик виявлення стратегічних помилок на кінцевих етапах проєктування, що може призвести до затримок у планах виконання проєкту. Сучасні підходи до проєктування ВС, такі як паралельне проєктування, об'єктно-орієнтоване проєктування, платформно-орієнтоване проєктування та методологія повторного використання покликані розв'язати деякі проблеми традиційного підходу до проєктування. Існуючі в цій галузі проблеми вимагають розвитку існуючих та впровадження нових методів і засобів АП.

Вагомий внесок у розвиток теорії та практики розробки ВС зробили такі вчені: Smith C.U. [9] – розробив підхід до паралельного проєктування; Wolf W. [10] – відокремив у проєктуванні ВС етап розподілу та запропонував схему розподілу складових ВС; Lee E.A. [11] – розробив систему Ptolemy для паралельної симуляції та швидкого прототипування; Sangiovanni-Vincentelli A. [12] – узагальнив та формалізував платформно-орієнтований підхід до проєктування ВС; Teich J. [13] – розробив модель подвійного «даху» паралельного проєктування; Лобур М.В. [14] – запропонував алгоритм проєктування та виготовлення ВС, який враховує специфіку цих пристроїв; Биковський С.В., Платунов А.Е. [15, 16] – узагальнили існуючі підходи та особливості проєктування вбудованих систем, та інші.

Сучасні ВС УРО можуть бути побудовані на основі використання різноманітних технологій [8,17]. Все частіше розробники ВС УРО надають перевагу використанню мікроконтролерів, оскільки вони дають змогу з мінімальними витратами реалізувати ефективну систему керування [18-23].

З метою реалізації платформно-орієнтованого підходу до проєктування виробники (Intel, Arduino, Raspberry, Texas Instrument, STMicrocontroller та

ін.), пропонують величезну кількість різноманітних апаратно-програмних платформ та компонентів, що відрізняються функціональністю та вартістю, і дають змогу прискорити створення проекту [24-26].

Перетворення он-лайн ринку у кілька мільярдну промисловість змінило позицію розробників ВС, надаючи їм доступ до великого розмаїття готових рішень, але спричинило проблему надлишковості інформації. Час, що витрачається на пошук та вибір апаратно-програмних платформ, які найкращим чином відповідають вимогам проекту, постійно збільшується через необхідність аналізу величезної кількості пропонованих варіантів.

Одним з рішень є використання рекомендаційних систем (РС), які дадуть змогу розробнику правильно обрати готову апаратно-програмну платформу в залежності від вимог до проекту.

Питанням дослідження та розвитку методів побудови рекомендаційних систем присвячені роботи F. Ricci та B. Shapira [27], R. Burke [28], J.K. Tarus [29], Ю.В. Стеха [30], О.Г. Дьяконов [31], Е.Е. П'ятикопа [32] та ін..

Існуючі на теперішній час РС використовуються в різних галузях для надання рекомендацій щодо вибору музики, фільмів, книг, електронних товарів, новин та web-сторінок. Українські Інтернет ресурси з продажу електронних компонентів [33,34] пропонують вже відфільтровану за назвами або за призначенням інформацію, наприклад: плати для розробок, плати розширення, сенсори і т.і. Це лише упорядковує інформацію, що безумовно спрощує її пошук, але не надає рекомендацій щодо вибору. Виробники електронних компонентів, такі як Texas Instrument [35] та STMicrocontrollers [36] пропонують деякі рекомендації, але, при цьому, концентруються на власній продукції (з метою її просування) та допомагають обрати додаткові можливості для розширення функціоналу лише своєї базової платформи.

У залежності від даних, з якими працює РС, використовуються різні рекомендаційні методи: колаборативна фільтрація, контентна фільтрація, фільтрація на основі знань, гібридна. Контентна та колаборативна фільтрація дають змогу користувачеві отримати рекомендацію, виходячи з його



попереднього досвіду. Це не є зручним для систем, які не зберігають дані про користувача та його поведінку, працюють з поточними даними та допомагають визначитись зі специфічними вимогами кожного користувача. Дослідження свідчать, що з нагальною потребою користувача працюють методи, які відносяться до фільтрації на основі знань. Тому, розробка РС формування рекомендацій щодо апаратно-програмних платформ на основі бази знань є актуальним завданням.

Крім того, для остаточного вибору апаратно-програмних платформ, окрім технічної специфікації, що надається виробником на сайті, потрібні експериментальні дослідження та досвід роботи з певним рішенням. Саме тому, активно використовуються різноманітні технології та засоби віддаленої інженерії (он-лайн симулятори, віддалені лабораторії, хмарні сервіси та ін.), які надають додаткові функціональні можливості користувачам десктопних систем автоматизованого проектування.

Віддалені лабораторії (ВЛ) є перспективною технологією, що активно розвивається. Вони дають можливість спільно використовувати обладнання та програмне забезпечення без необхідності його придбання, налагодження та встановлення безпосередньо на робочому місці, що надає переваги в умовах фінансово-економічних обмежень. Найбільш відомими ВЛ в галузі електроніки та автоматики є: GOLDI [37], WebLab-Deusto [38], Labshare [39], e-Laboratory Project[40], iLabs[41]. В основному, ВЛ сьогодні використовуються як суто освітні ресурси, але розробка та використання віддаленого інструментарію для дослідницьких та виробничих цілей наразі є актуальними.

Таким чином, розробка програмних та технічних засобів віддаленої інженерії (ВІ), які дадуть змогу проектувальнику, використовуючи рекомендаційну систему, обрати апаратно-програмну платформу згідно з вимогами до проектованої ВС УРО, а також виконати прототипування проектованої системи на основі віддаленого експерименту, є актуальним завданням та визначає напрям дисертаційної роботи.

### **Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційні дослідження виконувалися відповідно до наукового напрямку кафедри «Програмні засоби» Запорізького національного технічного університету: «Дослідження особливостей розробки та використання дистанційно керованих систем». Дисертація виконана в межах науково-дослідних робіт: «Інтелектуальні методи діагностування систем керування віддаленими технічними об'єктами» (номер державної реєстрації 0115U002242); «Інформаційна система діагностування розподілених мінікомп'ютерних систем в багатокомпонентному зовнішньому середовищі» (номер державної реєстрації 0117U000615).

**Мета і завдання дослідження.** Метою роботи є розробка моделей, методів та інструментальних засобів віддаленої інженерії для підвищення ефективності автоматизованого проектування вбудованих систем.

Для досягнення поставленої мети необхідно було розв'язати наступні завдання:

- проаналізувати методи, моделі, підходи та засоби автоматизованого проектування вбудованих систем з урахуванням їх структурних та функціональних особливостей, а також технологій реалізації;
- розробити метод автоматизованого проектування вбудованих систем на основі технологій віддаленої інженерії;
- розробити модель формування вимог до вбудованих систем та метод роботи з вимогами при автоматизованому проектуванні;
- розробити метод формування рекомендацій щодо апаратно-програмних платформ при автоматизованому проектуванні вбудованих систем;
- реалізувати програмне, технічне та інформаційне забезпечення для автоматизації проектування вбудованих систем на основі розроблених методів та моделей;
- розробити методику впровадження засобів віддаленої інженерії в процес автоматизованого проектування вбудованих систем.

*Об'єкт та предмет дослідження.* Об'єктом дослідження є автоматизоване проектування вбудованих систем на основі технологій віддаленої інженерії. Предмет дослідження – моделі, методи та інструментальні засоби віддаленої інженерії для автоматизації проектування вбудованих систем управління рухомими об'єктами.

**Методи дослідження.** У дисертаційній роботі для розв'язання поставлених завдань використано: при розробленні методів, моделей та алгоритмів – теорія системного аналізу, теорія математичного моделювання, теорія багатокритеріального аналізу, теорія баз знань; при розробленні програмних засобів – принципи об'єктно-орієнтованого програмування та веб-орієнтованої розробки.

**Наукова новизна одержаних результатів** полягає у тому, що:

1. Вперше розроблено метод автоматизованого проектування вбудованих систем з використанням технологій віддаленої інженерії, що дає змогу організувати спільне використання обладнання та програмного забезпечення (без необхідності їх придбання та налаштування) для прототипування проектованої ВС та, таким чином, знизити собівартість проектованої продукції.

2. Отримав подальший розвиток метод паралельного проектування вбудованих систем, який відрізняється від існуючого спільним застосуванням паралельного та платформно-орієнтованого підходів, а також засобів віддаленої інженерії, що дає змогу зменшити час переходу між системним та функціонально-логічним рівнями при проектуванні ВС та підвищити рівень автоматизації проектувальних робіт шляхом автоматизованого вибору апаратно-програмної платформи.

3. Удосконалено модель формування вимог до вбудованих систем та метод роботи з ними, які, на відміну від існуючих, враховують структурні особливості вбудованих систем, що містять апаратну та програмну складові, дають змогу розподіляти вимоги між програмними та апаратними компонентами вбудованої системи при автоматизованому проектуванні та,

таким чином, повніше враховувати вимоги до ВС та виконувати необхідні процеси роботи з вимогами при автоматизованому проектуванні.

4. Отримав подальший розвиток метод формування рекомендацій щодо апаратно-програмних платформ при автоматизованому проектуванні вбудованих систем на основі знань, який на відміну від існуючого, використовує методи обмежень та подібних об'єктів, а також методи багатокритеріального аналізу, що дає змогу надавати рекомендації навіть при суперечливості вимог користувача.

**Практичне значення одержаних результатів** полягає у тому що:

1. Розроблено структуру програмно-технічного комплексу, що включає інструментарій віддаленої інженерії, використання якого дає змогу підвищити ефективність процесу автоматизованого проектування ВС за рахунок зменшення часу проектування та собівартості проектованої продукції, а також підвищення рівня автоматизації проектувальних робіт.

2. Розроблено програмне та інформаційне забезпечення рекомендаційної системи, яка надає підтримку проектувальнику у вигляді рекомендацій щодо використання апаратно-програмних платформ при автоматизованому проектуванні ВС, виходячи з вимог до проектованої вбудованої системи.

3. Розроблено програмне та технічне забезпечення віддаленої лабораторії, яка дає змогу виконувати швидке прототипування проектованої ВС, перевірку на сумісність і працездатність апаратної та програмної частин, тим самим зменшувати час реалізації проекту.

4. Розроблено методику застосування створеного програмно-технічного комплексу при автоматизованому проектуванні вбудованих систем, яка може використовуватись у практиці інженерного проектування для ефективного впровадження засобів віддаленої інженерії.

Розроблений програмно-технічний комплекс автоматизованого проектування вбудованих систем на основі створених методів та моделей віддаленої інженерії впроваджено в процес автоматизованого проектування

систем керування об'єктами спецпризначення НВП ХАРТРОН-ЮКОМ (Запоріжжя), що підтверджено відповідним актом. Розроблену рекомендаційну систему вибору апаратно-програмних платформ на основі моделі формування вимог та методу роботи з ними впроваджено в практику інженерного проектування вбудованої системи керування комплексом електротехнічного обладнання ПКФ МОТОР (Запоріжжя). Також, результати дисертаційного дослідження використовуються у навчальному процесі Запорізького національного технічного університету для дисциплін «Інженерія вбудованих систем», «Сучасні CAD/CAM/CAE системи в проектуванні та виробництві наукоємної продукції», «Технології та системи віртуальної та віддаленої інженерії».

**Особистий внесок здобувача.** Всі наукові результати досліджень, які викладені в дисертації – одержані здобувачем особисто. У друкованих працях, опублікованих у співавторстві, автору дисертації належать: [42,43] – обґрунтовано спільне використання паралельного та платформно-орієнтованого проектування для розробки ВС; [44-46] – розроблено модель формування вимог до ВС та метод роботи з ними, з урахуванням апаратно-програмних особливостей ВС; [47-51] – обґрунтовано використання технологій ВІ для дослідження та прототипування ВС, а також методології повторного використання апаратних і програмних рішень; [52-54] – розроблено метод паралельного проектування ВС зі спільним використанням платформно-орієнтованого підходу, методології повторного використання та засобів віддаленої інженерії, а також метод автоматизованого проектування ВС з використанням технологій ВІ; [55-58] – обґрунтовано використання методології повторного використання при автоматизованому проектуванні ВС з використанням технологій віддаленої інженерії; [59] – розроблено метод формування рекомендацій щодо апаратно-програмних платформ при автоматизованому проектуванні ВС; [60] – розроблено віддалену лабораторію для швидкого прототипування ВС; [61-66] – розроблено ВС УРО на основі використання розроблених програмних та технічних засобів.

**Апробація результатів дисертації.** Основні положення та результати дисертації доповідались і обговорювались на міжнародних конференціях: «Data acquisition and advanced computing systems: technology and applications» (Бухарест, Румунія, 2017), «Remote engineering and virtual instrumentation» (Порто, Португалія, 2014; Мадрид, Іспанія, 2016); «Перспективні технології і методи проектування МЕМС» (Львів-Поляна, 2013, 2015, 2016); «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці» (Львів-Поляна, 2013); міжнародних науково-практичних конференціях: «Інтернет-Освіта-Наука» (Вінниця, 2014); «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій» (Запоріжжя, 2012, 2014); міжнародних симпозиумах: «Embedded systems and trends in teaching engineering» (Нітра, Словаччина, 2016), «Ambient intelligence and embedded systems» (Вааса, Фінляндія, 2017); науково-практичній конференції «Тиждень науки ЗНТУ» (Запоріжжя, 2016).

**Публікації.** Результати дисертації опубліковано у 22 наукових працях (з них 6 статей – 3 у наукових фахових виданнях України, включених до переліку МОН України; 3 – у наукових періодичних виданнях інших держав, що включені до наукометричних баз знань); 1 патент на корисну модель, 2 свідоцтва України про реєстрацію авторського права на твір (комп'ютерну програму); 13 праць у матеріалах міжнародних наукових конференцій та симпозиумів (з них 5 – за кордоном); 6 праць індексуються в науко-метричній базі Scopus (індекс цитування  $h=3$ ).

**Структура та обсяг роботи.** Дисертаційна робота складається з вступу, чотирьох розділів, висновків, списку використаних джерел із 114 найменувань та додатків. Робота викладена на 155 сторінках, з яких основного тексту - 124 сторінки.

# **РОЗДІЛ 1. ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ ПРОЦЕСУ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ**

У цьому розділі, відповідно до мети та завдань дисертаційної роботи, виконано аналіз сучасного процесу автоматизованого проектування вбудованих систем з урахуванням їх структурних та функціональних особливостей, а також досліджено шляхи підвищення рівня автоматизації проектувальних робіт. Показано, що застосування традиційного підходу до проектування ВС спричиняє певні проблеми та вимагає впровадження нових підходів до проектування, в тому числі на основі технологій віддаленої інженерії.

Проаналізовано існуючі стандарти, а також теоретичні та практичні розробки, що описують етапи життєвого циклу проєктованих систем, зокрема процеси роботи з вимогами до ВС. Це дало змогу утворити методологічну основу для удосконалення моделі формування вимог та методу роботи з ними при автоматизованому проектуванні ВС.

Виконано огляд програмних засобів, що використовуються для автоматизації проектувальних робіт при розробці ВС та показано переваги застосування віддаленого інструментарію для підвищення ефективності процесу автоматизованого проектування ВС.

## **1.1 Місце вбудованих систем управління рухомими об'єктами в загальній класифікації вбудованих систем**

Вбудована система (англ. Embedded System) визначається сьогодні як спеціалізована обчислювальна система, яка в силу розв'язуваного завдання безпосередньо взаємодіє з фізичними об'єктами і процесами [16].

Вбудовані системи є особливим класом систем, який можна віднести як

до автоматизованих, так і до інформаційно-керуючих систем, оскільки вбудована система підтримує (розв'язує) низку автоматизованих завдань і є вбудованою в пристрій, яким вона буде керувати.

В залежності від об'єкта, яким будуть керувати, а також мети створення системи, ринок ВС можна сегментувати наступним чином: автомобільний сегмент, телекомунікації, медицина, промисловість, робототехніка, військова і аерокосмічна техніка, побутова техніка, комп'ютерна периферія та інші [67]. Виходячи з цього, класифікація вбудованих систем за галузями застосування може бути представлена, як зображено на рис. 1.1.

Однією з найскладніших і трудомістких з точки зору проектування, категорій ВС є системи управління різними типами рухомих об'єктів, які працюють у режимі жорсткого реального часу та безпосередньо взаємодіють з оточенням. Ці системи розповсюджені у різних галузях економіки: промисловість, транспорт, оборонна галузь, сільське господарство, сфера розваг.

В залежності від середовища використання, рухомі об'єкти діляться на наступні групи: наземні апарати; надводні апарати; підводні апарати; літальні апарати.

На рис. 1.2 представлено загальну класифікацію рухомих об'єктів, яка є розширенням класифікації, запропонованої у [68]. Згідно наведеної класифікації, до класу наземних апаратів відносяться об'єкти (автомобілі, автобуси, трактори, потяги, людино або твариноподібні роботи), які для пересування використовують різноманітні механізми руху (колесо, гусеничний хід, нога, повзання/стрибки).

В даному дослідженні основна увага приділяється саме наземним колісним рухомих об'єктам. Розповсюдженими прикладами таких об'єктів є дистанційно керовані автомобілі, мобільні роботи, пізнавально-розважальні платформи, а також безпілотні таксі, автобуси та т.і., що використовуються у різних сферах економіки та у військовій сфері [69-71].



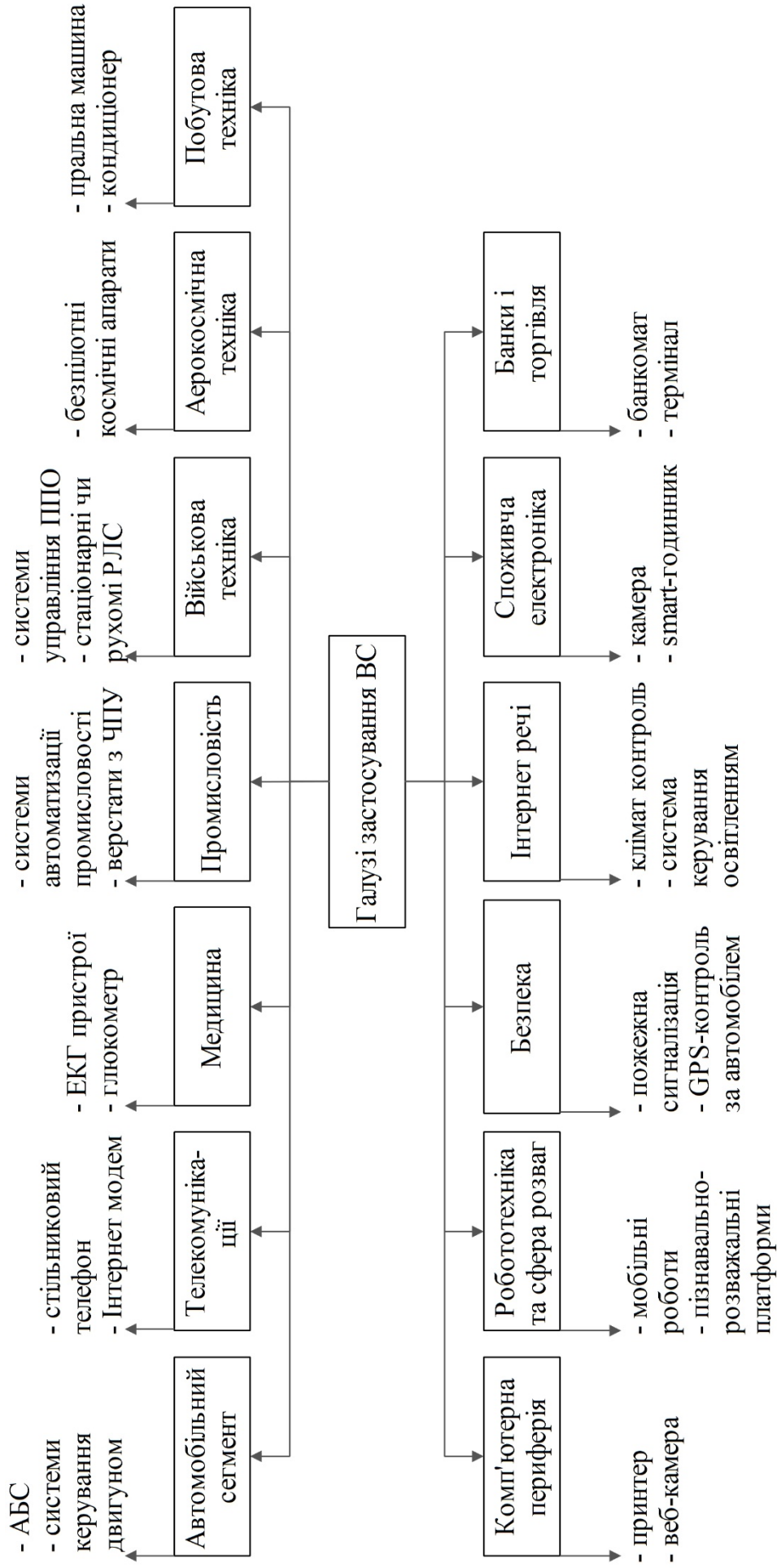


Рис. 1.1. Класифікація вбудованих систем за галузями застосування

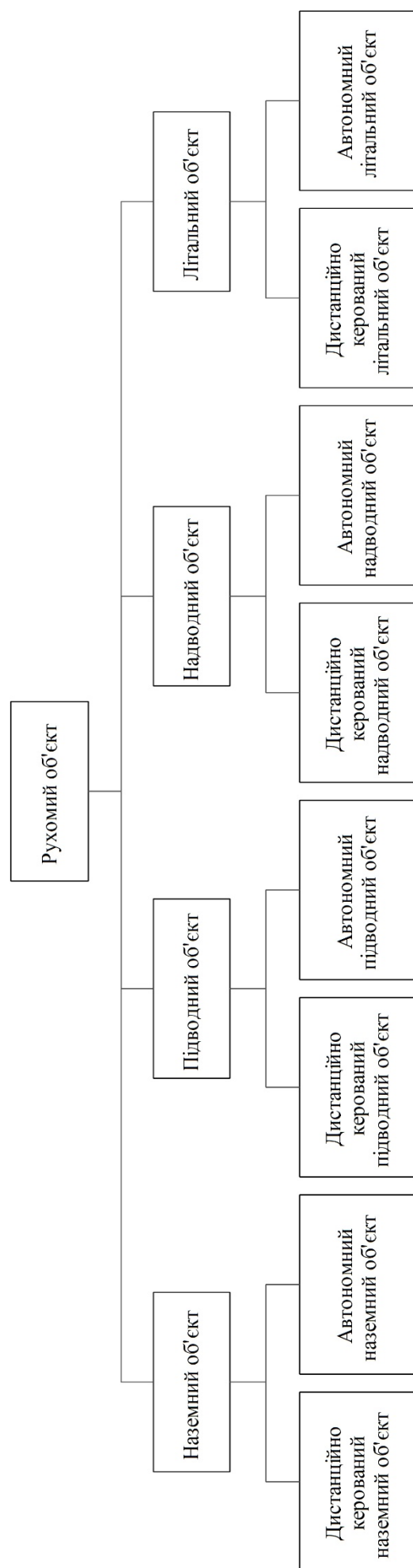


Рис. 1.2. Загальна класифікація рухомих об'єктів

Як показали проведені дослідження, системи управління рухомими об'єктами (УРО) можуть бути створені для об'єктів двох типів:

- автономні об'єкти, які будуються на основі різних сенсорів, актуаторів та “мозку” на основі штучного інтелекту;
- об'єкти дистанційного управління, що керуються людиною з використанням різних інтерфейсів взаємодії та каналів зв'язку.

Системи управління автономними об'єктами потребують реалізації складних алгоритмів розпізнавання перешкод та пошуку траєкторії руху, цілком залежать від керуючої програми, а також вимагають високоточного обладнання для взаємодії з оточенням. Деякі закордонні автовиробники (наприклад, Tesla, Mercedes та інші) вже запропонували свої моделі безпілотних автомобілів і провели випробування [72]. Але, розробники таких систем поки що стикаються з проблемами складності реалізації та забезпечення надійності функціонування автономних систем.

Робота системи дистанційного управління залежить здебільшого від дій оператора та якості інтерфейсу взаємодії (зв'язку між людиною та керованим об'єктом). Слід відзначити вітчизняну розробку компанії ТОВ СП «Інфоком» - безпілотна система спеціального призначення на базі армійського броньованого автомобіля КраЗ-СПАРТАН [73], керованого дистанційно.

Системи дистанційного управління можна розрізнити за типом каналу зв'язку [74]:

- провідний канал (використовується, коли немає можливості застосувати бездротові канали, або з міркувань вартості і захищеності від перешкод);
- радіоканал (головним чином для радіокерування рухомими об'єктами, а також в ситуаціях, коли передавач і приймач не можуть перебувати в зоні прямої видимості);
- ультразвуковий канал (управління мобільними і стаціонарними об'єктами на порівняно невеликій відстані);

- інфрачервоний канал (зазвичай для керування побутовою електронікою).

Радіоканал є одним з найрозповсюдженіших типів каналу зв'язку для дистанційного керування рухомим об'єктом [75]. Він може використовуватись для модуляції FM, AM або SSB сигналів, діапазон частот також може бути різним. В залежності від обраного каналу зв'язку та частоти, відстань або радіус керування рухомим об'єктом також різняться. З розвитком технологій Інтернету речей з'явилась можливість дистанційного керування на основі технології Wi-Fi.

Як показали проведені дослідження існуючої системи дистанційного керування перегонами [76], вона складається з радіокерованої моделі автомобіля, що містить у своєму складі акселерометр та Bluetooth приймач, а також спеціального програмного забезпечення, встановленого на мобільному телефоні. У режимі реального часу система дистанційного керування передає інформацію про швидкість автомобіля на мобільний телефон. Малий радіус дії Bluetooth передавача (до 10 м), не дозволяє використовувати систему для керування рухомими об'єктами на більших відстанях. Крім того, в системі не передбачені можливості налаштування параметрів рухомих об'єктів, а також віддалене одночасне керування декількома об'єктами.

Інша досліджена автоматизована система [77] дозволяє керувати робочим органом транспортної мережі за допомогою радіоканалу. Вона складається з робочого органу транспортної мережі, в якому встановлений блок пам'яті, що забезпечений відповідними командами блока зв'язку, з'єднаного з пультом радіо-дистанційного керування, укомплектованим програмним мікропроцесором, і через блок зв'язку з'єднаного з пультом керування системою та блоком виводу інформації. Також, до складу системи входить набір навантажувально-розвантажувальних постів, що утворюють транспортний шлях. Вказана система працює на частоті радіохвиль 27 МГц. Недоліками цієї системи є відсутність прийому/передачі відео-зображення для контролю процесу руху. Використання діапазону СВ (Citizen`s Band, 27 МГц),

який має великий радіус дії, не гарантує стійкого і надійного зв'язку, оскільки є чутливим до впливу побутових і промислових перешкод. Крім того, він має слабку проникаючу здатність, а отже, не призначений для використання в умовах міської забудови. Для роботи на цій частоті практично не існує компактного обладнання, що не дозволяє змінювати швидко дислокацію системи, в свою чергу це знижує її мобільність.

Виходячи з аналізу особливостей ВС УРО, можна зробити висновок, що дистанційно керовані об'єкти, головним чином, реалізуються з використанням радіоканалу, оскільки дротове з'єднання перешкоджає гнучкості руху, а інші типи каналів значно поступаються потужністю сигналу.

Дистанційно керовані об'єкти мають автономне живлення. Живлення бортового обладнання виконується від акумулятора через стабілізатор напруги. Літій-полімерний акумулятор є найбільш популярним серед розробників дистанційно-керованих рухомих об'єктів, оскільки він має гарні показники вага/ємність, тобто є легким, але при цьому має достатньо велику ємність. Також, у дистанційно керованих рухомих об'єктах найбільш розповсюджені електродвигуни.

Дослідження подібних систем керування доцільно проводити на прототипах дистанційно керованих об'єктів. Моделі-прототипи є корисними з точки зору дослідження методів проектування вбудованих систем, а також розробки та удосконалення систем управління рухомими об'єктами. Тому, створення моделей-прототипів рухомих об'єктів є доцільним, економічно обґрунтованим та доступним.

Таким чином, ВС УРО є специфічним класом вбудованих систем, що використовуються у різних галузях людської діяльності. Вони є складними системами, які взаємодіють з оточенням та у режимі реального часу реагують на керуючі сигнали. Автоматизоване проектування вбудованих систем дистанційного керування рухомими об'єктами, які надають доступ людині до небезпечних та важкодоступних місць, є актуальним та економічно обґрунтованим.

Досліджені приклади реалізації автоматизованої системи віддаленого керування рухомими об'єктами не передбачають можливостей роботи з групою об'єктів, не забезпечують потрібного радіусу віддаленого керування (50 метрів та більше), а також мобільності, завадостійкості та надійності функціонування системи. Отже, актуальним є завдання створення автоматизованої системи віддаленого управління рухомими об'єктами, що надасть можливість керування рухом декількох об'єктів одночасно, налаштування їх параметрів, дозволить створити ефект присутності операторів у досліджуваній зоні для контролю траєкторії та умов руху, а також надасть актуальну інформацію про поточний стан рухомих об'єктів диспетчеру [65].

## **1.2 Аналіз особливостей вбудованих систем та технологій їх реалізації**

Для розробників обчислювальної техніки, ВС є одним з найбільш складних об'єктів проектування [15]. Аналіз типових вимог та обмежень, які необхідно враховувати при створенні ВС, підтверджує це. Виконавши дослідження робіт [15, 16, 18], було сформовано наступний список вимог та обмежень, що визначають якість проекрованої ВС:

- робота у реальному часі;
- міцність конструкції;
- мінімізація габаритів і ваги з урахуванням розміру і форми виробу;
- забезпечення теплових режимів (оскільки вимога мінімізації габаритів часто не дозволяє встановлювати системи примусового охолодження);
- мінімізація енергоспоживання (оскільки живлення у таких системах переважно автономне);
- радіаційна і електромагнітна стійкість (додатково - працездатність у вакуумі);

- гарантований час напрацювання на відмову;
- термін доступності рішення на ринку та т.і..

Власне дослідження існуючих систем автоматизованого керування рухомими об'єктами [65] дозволило виявити додаткові вимоги до ВС УРО:

- радіус віддаленого радіо керування 50 метрів та більше;
- мобільність;
- завадостійкість;
- надійність функціонування системи;
- можливість керування рухом декількох об'єктів одночасно;
- наявність функції налаштування параметрів керування;
- можливість прийому/передачі відео-зображення з рухомого об'єкта для контролю процесу руху та створення ефекту присутності операторів у досліджуваній зоні.

Системи дистанційного керування рухомими об'єктами мають свої особливості конструктивної реалізації. Ефективність таких систем здебільшого визначається інтерфейсом, характеристиками каналу зв'язку, а також діями оператора.

Аналіз конструктивних особливостей ВС УРО показав, що до складу типової ВС входять (рис. 1.3) [8, 14]:

- головний модуль керування;
- периферійна система (сенсори, актуатори і контролери введення-виведення для зв'язку з об'єктом контролю/управління, пристрої людино-машинного інтерфейсу за необхідністю);
- система електроживлення;
- конструктив, що об'єднує (шасі, корпус);
- програмне забезпечення (ПЗ), що керує.

Таким чином, завдання забезпечення вимог до проектованої ВС УРО має бути розв'язана комплексно з урахуванням взаємодії апаратної та програмної складових системи та особливостей їх реалізації.

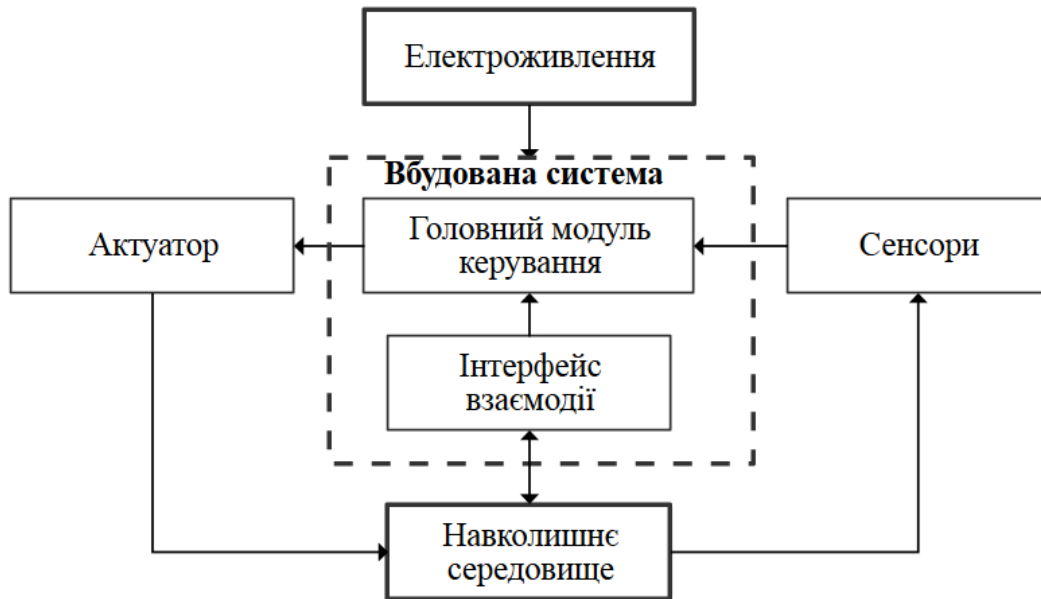


Рис. 1.3. Узагальнена структурна схема ВС УРО

У залежності від функціоналу, який необхідно реалізувати в системі, а також від мети створення такої системи, технології реалізації ВС УРО також різняться.

Апаратна частина головного модуля керування сучасних ВС УРО може бути реалізована з використанням різноманітних технологій (рис.1.4) [8]:

- промислові комп'ютери;
- програмовані логічні контролери (ПЛК);
- мобільні та Інтернет пристрої;
- контролерні і сенсорні мережі;
- мікроконтролери (МК);
- сигнальні процесори;
- програмовані логічні інтегральні схеми (ПЛІС);
- надвеликі інтегральні схеми (НВІС).

Всі технології мають як сильні, так і слабкі сторони. У більшості випадків, вибір залежить від складного поєднання чинників, і жодна з цих технологій не буде виступати в якості ідеального рішення [8].





Рис. 1.4. Технології реалізації вбудованих систем

При реалізації ВС УРО використання МК дає низку переваг: компактність, низька собівартість та мінімальне енергоспоживання, що є важливим для пристроїв автономного живлення, зокрема, якими і є рухомі об'єкти. До того ж, як вважають фахівці, системи виконані на базі МК, є більш гнучкими, оскільки керуюча програма може бути багаторазово змінена без змін апаратної частини пристрою [17, 20-23].

Як відомо, розробники МК з метою популяризації та просування своїх виробів, широко пропонують готові апаратно-програмні платформи, що надають базовий функціонал, який може бути розширений за необхідності. Сьогодні на ринку існує дуже багато варіантів апаратно-програмних платформ для розробки ВС: Arduino (Mega, Uno, Leonardo, Mini, Yun та інші), Raspberry Pi, TI MSP430 Launchpad, Beagleboard, Intel Galileo, pcDuino, STM32 Nucleo, Strela, Iskra (Mini, Neo), Netduino, Teensy 3.2, CMUcam та інші. Є ціла низка традиційних порівняльних факторів, які використовуються при виборі платформ. До них відносяться: параметри контролера (швидкість, енергоспоживання, периферійні пристрої на кристалі), доступні функції (підтримувані протоколи зв'язку, з урахуванням пам'яті), периферійні пристрої, що вбудовані на платформі, а також технічна підтримка (набір

інструментів, необхідний для створення ПЗ, приклади застосування) від постачальника [24-26]. У табл. 1.1 наведено результати дослідження та порівняння характеристик деяких найбільш популярних платформ.

Аналіз ринку апаратно-програмних платформ показав, що в загальному випадку, платформи можна поділити на два типи: платформи на основі МК та одноплатні комп'ютери (або мінікомп'ютери). При порівнянні специфікацій, платформи на основі МК значно поступаються одноплатним комп'ютерам. Але при виборі платформи для розробки проекту брати до уваги лише одну специфікацію є не зовсім правильним.

Таблиця 1.1.

## Результати порівняння популярних апаратно-програмних платформ

Назва платформи	Процесор	Сімейство	ПЗП	ОЗП	Частота	Аналогові	Цифрові
Arduino Uno	ATmega328	AVR	32KB	2KB	16MHz	6	14
Arduino Due	ATSAM3X8 E Cortex-M3	ARM	512 KB	96 KB	84MHz	16	54
MSP-EXP430G2 LaunchPad	MSP-EXP430G2	MSP430	16KB	512B	16MHz	8	16
Raspberry Pi 3	Broadcom BCM2837	ARM	-	1GB	1.2GHz	0	27
Intel Galileo	Quark SoC X1000	x86 Quark	8MB	512 KB	400 MHz	6	20

Наприклад, якщо необхідно побудувати просту систему управління рухомим об'єктом, яка включає в себе керування приводами, сенсорами та іншою периферією, то цілком достатньо буде взяти платформу на основі МК (наприклад Arduino Uno, MSP430 Launchpad). Якщо ж стоїть завдання обробки медіа (розпізнавання зображень, мови), вихід до мережі Інтернет, то простий МК не здатен розв'язувати такі завдання (платформи на МК можна

підключити до мережі, але для цього необхідні додаткові плати розширення). Тому, в цих випадках доцільно використовувати одноплатний комп'ютер (наприклад, RaspberryPi, pcDuino3, Beagleboard).

У випадках, коли необхідно розробити складну систему управління різними робочими механізмами (з обробкою відео та інформації з сенсорів), рекомендується розбивати завдання між МК та мінікомп'ютером. Це пов'язано з тим, що через багатозадачність, мінікомп'ютер може невчасно виконати критичне до часу завдання, що є неприйнятним при створенні керованих рухомих об'єктів реального часу. Таким чином, потрібно розподілити точну роботу з сенсорами та актуаторами на МК, а складну частину обробки даних на процесор. Тому, порівнювати ці два типи платформ не завжди є доцільним. У табл. 1.2 зведена виявлена інформація щодо особливостей застосування платформ на основі МК та одноплатних комп'ютерів [24-26].

Платформи на основі МК та одноплатні комп'ютери можна об'єднувати за допомогою інтерфейсів зв'язку (USB, локальна мережа, послідовне підключення портів вводу/виводу, радіоканал). Також, на сьогодні вже існують гібридні платформи, які об'єднують в собі і МК і мікропроцесор (Arduino Yun, Embedded Pi). Слід зауважити також, що на ринку існують спеціалізовані апаратно-програмні платформи, орієнтовані на виконання конкретних завдань (наприклад, для робототехнічних систем – Strela, Romeo BLE/ -mini). Відмінність існуючих платформ також полягає у форм-факторі, тому вибір платформи суттєво впливає на виконання вимог щодо масо-габаритних показників проектованої системи. Не менш важливим критерієм є наявність Інтернет спільки користувачів певної платформи, а також відкритість проекту, особливо для розробників-початківців.

Отже, як показав проведений аналіз, існує велика кількість технологій реалізації ВС УРО. Використання готових апаратно-програмних платформ на основі МК сприяє скороченню часу проектування ВС, надає можливості повторного використання в інших проектах, але, з іншого боку, пошук відповідної завданням проекту платформи зазвичай вимагає значних часових витрат.

## Особливості застосування різних апаратно-програмних платформ

	<b>Платформи на основі МК</b>	<b>Одноплатні комп'ютери</b>
<b>Продуктивність</b>	≈10-100МГц, ≈10КБ RAM, ≈10-100КБ ROM	≈100-1000МГц, ≈100КБ RAM, ≈1ГБ ROM
<b>Мережа</b>	Додаткові модулі та глибоке знання протоколів.	Підключення просте, модуль роботи з мережею є на платформі.
<b>Живлення</b>	Споживає одиниці-десятки мА. Тижні автономної роботи.	Споживає сотні-тисячі мА. Години автономної роботи та складності при її реалізації.
<b>Обробка відео</b>	Не вистачає потужності.	OpenCV, апаратні відеокодеки, HDMI-вихід
<b>Робота у реальному часі</b>	Повний контроль над часом.	Можуть виникати затримки роботи через багатозадачність або помилки ОС.
<b>Сенсори</b>	З легкістю працюють з будь-якими сенсорами.	У деяких є проблема підключення аналогових пристроїв. Для взаємодії необхідне ПЗ.
<b>Мова програмування</b>	Частіше мови C/C++ та їх різновиди.	Не обмежена мова C. В залежності від можливостей ОС.

Для обґрунтованого вибору апаратно-програмної платформи необхідно враховувати комплекс вимог до проектованої системи, знати особливості існуючих на ринку готових апаратно-програмних платформ, ретельно аналізувати їх специфікації, а також мати досвід роботи з обраним проектним рішенням.

### 1.3 Дослідження існуючих підходів до проектування вбудованих систем

Оскільки ВС є особливим класом систем, що містить апаратну та програмну складові, тому дуже часто використання традиційного підходу [8, 13] до їх розробки спричиняє наступні недоліки:

- неформальне розбиття на апаратну та програмну частини на початковому етапі;
- послідовне та відокремлене проектування апаратної і програмної складових;
- ручна інтеграція апаратної і програмної частин проекту;
- компенсація помилок при налагодженні за рахунок зміни програми [15].

Як наслідок, маємо високий ризик виявлення помилки вже на останніх етапах проектування, що призводить до критично довгих затримок виходу продукту на ринок [13].

Експерти [10,13,15,78] представляють та аналізують наступні, найбільш перспективні підходи до проектування вбудованих систем, що спрямовані на розв'язання проблем традиційного підходу:

- паралельне проектування (co-design);
- об'єктно-орієнтоване проектування (object-oriented design);
- платформно-орієнтоване проектування (platform-based design).

Процес паралельного проектування вбудованих систем у загальному вигляді наведено на рис. 1.5 [10]. Насправді він набагато складніший та ітераційний, як на самих етапах, так і між ними.

Процес проектування починається зі створення специфікації проекту, тобто оформлення вимог до системи. Далі створюється архітектура системи, в якій відокремлюються програмні та апаратні засоби. Проектування апаратного та програмного забезпечення розподіляється певним чином у часі [13]. Після реалізації апаратних і програмних засобів, вони повинні бути окремо

перевірені, інтегровані, і далі вся система тестується в цілому [79].

Паралельне проектування має як переваги, так і недоліки. З одного боку, спричиняє ускладнення процесу проектування ВС, а з іншого боку, дає значне поліпшення характеристик кінцевого продукту у порівнянні з альтернативними версіями проектних рішень.

Об'єктно-орієнтоване проектування ВС перейняло загальні методи, які використовуються при створенні програмного забезпечення. У цьому підході програмні методи застосовуються до мов опису апаратного забезпечення. Причиною виникнення цього підходу є намагання скористатися перевагами об'єктно-орієнтованої методології, а саме методологіями повторного використання (reuse) та управління складністю [80].

Платформно-орієнтований підхід базується на методології повторного використання [12, 81] та дозволяє значно скоротити час проектування завдяки повторному використанню попередньо реалізованих і протестованих програмних та апаратних компонентів системи. До повторно використовуваних програмних компонентів належать: операційні системи реального часу [82], прикладні програмні інтерфейси (API) [56-58], бібліотеки, драйвери та ін..



Рис. 1.5. Узагальнена схема методу паралельного проектування ВС

В якості повторно використовуваних компонентів можуть виступати, також апаратно-програмні платформи (Arduino, STM Discovery та ін.). Використання таких платформ спрощує етап проектування ВС, завдяки використанню готових конструктивних рішень. Крім того, спеціалізоване програмне забезпечення, яке постачається в комплекті з ними, високорівнева мова програмування, велика кількість стандартних функцій та бібліотек, спрощує написання та відлагодження програмного забезпечення ВС [42].

Необхідно зауважити, що виведення на широкий ринок готових апаратно-програмних платформ поставило перед їх розробниками завдання універсалізації доступу та роботи з цими пристроями. Ціною шаблонного принципу побудови стали проблеми застосування таких платформ. Так, для реалізації складного функціоналу системи, доводиться інколи об'єднувати обрану платформу з платами розширення (шилдами), що призводить до збільшення вартості, габаритів і маси, а це зазвичай є неприйнятним при проектуванні ВС. Тому, для правильного вибору апаратно-програмних платформ необхідно враховувати цілу низку факторів, основними з яких є вимоги до ВС.

Збір та аналіз вимог, додавання обмежень (щодо швидкодії, габаритів, надійності, енергоспоживання та ін.) та аналіз відповідності готових апаратних та програмних рішень встановленим вимогам є важливим етапом високорівневого проектування ВС. Подальший правильний розподіл вимог між програмними і апаратними компонентами архітектури на етапі високорівневого проектування є одним з найважливіших факторів успіху при реалізації ВС.

Аналізуючи розподіл часу за стадіями проектування, представлений у [83], можна виявити, що при збільшенні часу, затраченого на стадію високорівневого проектування системи, зменшується час, витрачений на подальше прототипування та тестування. Однак, як зазначають фахівці, значний розрив, що існує між вимогами до ВС і ефективністю апаратно-програмного проектування, необхідними об'ємами верифікації та тестування

пристроїв найближчим часом буде зростати [15]. В основному, проблема полягає в неспроможності існуючих підходів до проектування ВС на тлі сучасних вимог до таких систем.

Таким чином, розробка ВС «з нуля» з використанням традиційного підходу до проектування сьогодні не є ефективною. Саме тому, в області проектування ВС активно розвиваються такі підходи до проектування як паралельне та платформно-орієнтоване проектування. Стає необхідним також застосування методології повторного використання, що дозволяє прискорити розробку проекту та, таким чином, скоротити час виходу продукції на ринок. Отже, подальший розвиток методів проектування ВС, а також накопичення існуючих проектних (програмних та апаратних) рішень з метою повторного використання є одними з ключових напрямків підвищення ефективності автоматизованого проектування ВС.

#### **1.4 Дослідження процесів роботи з вимогами при автоматизованому проектуванні вбудованих систем**

Як показано в стандарті [84], проектування будь-якої системи включає в себе наступні технічні процеси роботи з вимогами: процес визначення вимог правовласника та процес аналізу вимог. Результатом процесу визначення вимог правовласника є перелік всіх зацікавлених осіб, а також їх вимоги до системи, які є основою для подальшого аналізу і виявлення функціональних і нефункціональних вимог до системи. Результатом процесу аналізу вимог є повне перетворення вимог правовласника (зацікавлених осіб) в технічне бачення розроблюваної системи, тобто чітко визначену класифікацію вимог до системи. Процес аналізу вимог суттєво впливає на архітектурне проектування, а також на засоби його реалізації.

На сьогоднішній день, питання роботи з системними вимогами розглядаються в [5,85,86]. Аналіз стандарту [86] показав, що під поняттям



«Вимоги до системи» розуміється наступна сукупність:

- вимоги до системи в цілому (бізнес вимоги, зовнішній інтерфейс);
- вимоги до функцій (завдань), що виконуються системою (функціональні вимоги, атрибути якості);
- вимоги до видів забезпечення (обмеження).

До першого виду вимог відноситься також опис структури системи, куди входить перелік її компонентів. Для складних проектів, що включають велику кількість компонентів (підсистем), необхідно провести розподіл вимог високого рівня за цими компонентами [5].

Першим кроком до об'єданого комплексу стандартів, що описують процеси життєвого циклу систем і програмних засобів, є міжнародний стандарт [85]. Він встановлює строгий зв'язок між системою і використовуваними в ній програмними засобами, трактуючи програмний засіб, як складову частину системи, за допомогою виділення вимог до програмних засобів з вимог до системи. У стандарті наведено різницю між аналізом системних вимог і аналізом вимог до програмного продукту, оскільки в загальному випадку побудова системної архітектури визначає системні вимоги для різних складових частин системи, а аналіз вимог до програмних засобів зумовлює вимоги до них, виходячи із системних вимог, призначених кожній складовій програмній частині.

Крім перерахованих у стандарті [84] процесів роботи з вимогами до системи, у стандарті [85] виділяються також спеціальні процеси програмних засобів, призначені для створення конкретного програмного елемента системи, зокрема процес аналізу вимог до ПЗ. В ході цього процесу визначаються вимоги до програмних елементів системи та їх інтерфейсів, встановлюється вплив вимог до ПЗ на середовище функціонування, а також сумісність та зв'язок між вимогами до ПЗ і вимогами до системи.

В [87] підкреслюється, що вхідними даними для процесу визначення вимог до ПЗ є системні вимоги, опис апаратного інтерфейсу і архітектури системи (якщо вони не включені в системні вимоги). А мета процесу

проектування ПЗ полягає у тому, щоб розробити архітектуру ПЗ і вимоги нижнього рівня на основі вимог верхнього рівня до ПЗ. Звідси випливає, що розробка самої архітектури ПЗ відбувається після опису апаратної частини проекту та її архітектури.

У роботі Д. Леффінгвелла [6] детально розглядається процес управління вимогами. Модель формування вимог Леффінгвелла складається з 3-х рівнів:

- потреби – вимоги користувача/зацікавлених осіб;
- функції –обслуговування, що надається системою для задоволення вимог зацікавлених осіб;
- програмні вимоги – конкретизовані вимоги до програмного забезпечення.

Кластер «потреби» стосується області розуміння проблеми користувачів (замовників) для подальшої побудови системи, що задовольняє їх потребам. Наступні два кластери «функції» і «програмні вимоги» стосуються області розв'язку проблем користувача.

Як наголошується в [88], в різних методологіях розробки ПЗ, застосовується підхід, який базується на визначенні груп вимог до програмного продукту. Такий підхід зазвичай включає групи (типи, категорії) вимог, наприклад: системні, програмні, функціональні, нефункціональні (зокрема, атрибути якості) і т.і.

Класичний приклад високорівневого структурування груп вимог, як вимог до програмного продукту, представлений у роботі К. Вігерса [5]. Модель формування вимог, запропонована Вігерсом для проектування ПЗ складається з трьох рівнів вимог:

- бізнес – навіщо розробляється система;
- користувача – описують користувачів системи, а також їх роботу з нею;
- функціональні – визначають функціональність (поведінку) програмної системи.

Модель передбачає два типи вимог:

- функціональні – що система повинна робити;
- нефункціональні – які умови при роботі повинні виконуватися.

У розглянутій моделі формування вимог під зовнішнім інтерфейсом розуміють «інтерфейс користувача», тобто взаємодію користувача з ПЗ. Для розробки ВС в стандарті [87] надається більш розширене визначення інтерфейсу – це взаємозв'язок між двома або більше об'єктами, які спільно використовують і забезпечують дані або обмінюються ними. В якості таких об'єктів можуть виступати: елемент конфігурації ПЗ (ЕКПЗ) - сукупність компонентів ПЗ, а також елемент конфігурації апаратури (ЕКА) - сукупність компонентів апаратного забезпечення (АЗ). Відповідно, варіанти інтерфейсів можуть бути наступні: ЕКПЗ/ЕКПЗ, ЕКПЗ/ЕКА, ЕКПЗ/користувач, ЕКА/користувач або модуль ПЗ/модуль ПЗ.

У стандарті [86] дається таке визначення вбудованої системи – це набір апаратних і програмних компонентів, створений для виконання певної функції або низки функцій. Оскільки розробка ПЗ є важливою складовою проектування ВС, потрібно застосовувати комплексний підхід на основі процесів системної та програмної інженерії, і визначити як системні (високорівневі вимоги до всього проекту), так і програмні вимоги (тобто вимоги, розподілені між програмними компонентами даного проекту).

Одним з важливих етапів контролю створюваних вимог є трасування (взаємозв'язок). Трасування необхідне для розуміння того, як вимоги верхнього рівня перетворюються на вимоги низького рівня. Щоб забезпечити трасування, кожна вимога повинна бути унікальною і ідентифікованою, щоб мати можливість посилатися на неї. Найкраще керувати трасуванням, використовуючи методи управління вимогами [89] та відповідні інструменти. Можна виділити такі програмні інструменти управління вимогами як: IBM Rational RequisitePro, Borland CaliberRM, DOORS. Для Rational RequisitePro можна відзначити низку переваг:

- підтримка Word для створення проектної документації;

- наявність шаблонів для розробки проектної документації;
- наявність зручного навігатору між Microsoft Word і потужною інфраструктурою бази даних;
- можливість інтеграції із засобами об'єктно-орієнтованого моделювання;
- можливість простеження взаємозв'язку вимог різних рівнів за допомогою Матриці трасування.

Складання переліку вимог включає в себе структурування вимог, тобто розподіл вимог за групами, котрі формалізуються подібним чином [88]. Одним з критеріїв, за якими можна проводити розподіл, є компоненти або об'єкти (підсистеми) системи. Для ВС головними складовими є апаратні та програмні компоненти, за якими виконується розподіл процесу проектування у часі [19, 78]. У роботі [89] процес апаратного проектування також пропонується розподілити на процеси проектування електроніки та механіки.

Відповідно, перелік вимог до ВС можна розподілити за такими основними групами: вимоги до системи в цілому; вимоги до ПЗ; вимоги до АЗ (до електричної частини та до механічної частини); нефункціональні вимоги.

Таким чином, робота з вимогами є дуже важливою при створенні проекту ВС для досягнення поставлених цілей з його реалізації. Існуючі на сьогоднішній день стандарти, а також роботи в галузі створення і аналізу вимог описують лише окремі аспекти цього завдання і не враховують всі особливості автоматизованого проектування ВС.

Отже, моделі вимог до ПЗ, представлені в роботах [5,6], можуть розглядатися в якості основи для розробки моделі формування вимог до ВС УРО, яка врахує їх специфічні особливості як сукупності апаратних та програмних компонентів. Однак, вони мають бути модифіковані з метою урахування інших типів вимог, а також процесів роботи з вимогами, представлених у стандартах [85-87]. Подальше застосування моделі вимог та методу роботи з ними дозволить виконати обґрунтований вибір апаратно-програмної платформи при АП ВС.

## **1.5 Програмні засоби для автоматизації проектування вбудованих систем**

На сьогоднішній день на різних етапах АП ВС використовується різноманітне програмне забезпечення для рішення завдань системного, функціонально-логічного, схемотехнічного та конструкторського проектування [44, 46, 79], а також моделювання електричних, теплових, механічних та інших процесів [54].

Наприклад, відомі ECAD (Electronic Computer Aided Design) системи (Altium Designer, SPICE, EAGLE, ISIS PROTEUS, AutoCAD Electrical, KiCad, Circuit Maker та інші) дозволяють виконувати розробку схем електричних принципів, проектування друкованих плат, схемотехнічний аналіз, 3D візуалізацію друкованої плати та інші проектні процедури. Деякі з ECAD систем (наприклад, PROTEUS) надають можливість проводити моделювання роботи програмованих пристроїв та електронної схеми в цілому (симуляцію роботи систем). Більшість дослідженого програмного забезпечення є платним, вимагає значного часу на вивчення, потребує достатньо великих ресурсів комп'ютера, а застосування його повного функціоналу не завжди є потрібним.

Швидкий розвиток Інтернет-технологій та їх зростаюча популярність справили величезний вплив на галузь проектування, тому поряд з десктопними системами активно використовуються технології та інструментарій віддаленої інженерії (он-лайн симулятори, віддалені лабораторії, рекомендаційні системи, хмарні сервіси та ін..). Віддалена інженерія (online, remote) має на меті організацію спільного використання обладнання та ресурсів, а також спеціалізованого програмного забезпечення (наприклад, симулятори) [90,91].

Використання інструментальних засобів віддаленої інженерії є перспективним напрямом в області проектування ВС, що визначається наступними тенденціями:

- зростаючою складністю інженерних завдань, які потребують розв'язку в короткий термін реалізації проектів;

- використанням у процесі проектування все більшої кількості спеціалізованого і дорогого обладнання, а також програмних засобів;
- недоступністю високотехнологічного обладнання для малих і середніх підприємств;
- необхідністю висококваліфікованого персоналу для управління сучасним обладнанням;
- вимогами глобалізації ринку і поділу праці [90].

Серед он-лайн інструментарію для створення схем електричних принципів та друкованих плат можна виділити Circuits від Autodesk [92] та сервіс EasyEDA [93].

У зв'язку з швидким розвитком популярності використання готових апаратно-програмних платформ при проектуванні ВС, деякі компанії пропонують програми-симулятори для побудови віртуального прототипу ВС з наступним моделюванням її роботи: Virtualbreadbord, Virtronics, Simuino та ін. [94-96].

Компанія Autodesk пропонує он-лайн версію симулятора для Arduino – Tinkercad [97]. Microsoft створив он-лайн симулятор для платформи Raspberry Pi [98]. Вони дозволяють проектувати схему електричну, проводити налагодження програми та симуляцію роботи платформи з базовими сенсорами та актуаторами.

Сервіс TIDesigns від компанії Texas Instruments [35] використовує засоби WEBENCH, працює як Flash-додаток у браузері, та має назву WEBENCH Design Center. WEBENCH Design Center забезпечує розв'язування наступних завдань: розробку схеми налагодженого живлення або освітлення, з використанням власних продуктів компанії; симуляцію проходження струму та моделювання теплових процесів схеми. Запропоновані варіанти реалізації схем можна оптимізувати щодо вартості компонентів, компактності схеми та працездатності. Після завершення тестування можна експортувати схему електричну принципову та друковану плату у формати, які підтримуються системами Altium Design, OrCAD, EAGLE та ін..

Симулятори є важливим інструментарієм при паралельному проектуванні ВС. Але, як відомо, симуляція не замінює реальної роботи з апаратним і програмним забезпеченням ВС. Фактично, замість реального фізичного процесу, симулятор дозволяє вивчити лише його математичну модель, від точності якої залежать результати моделювання. Як показали дослідження, існуючі програми-симулятори зазвичай мають обмежені функціональні можливості та в деяких випадках неповну елементну базу.

Віртуальні та віддалені лабораторії є засобами віддаленої інженерії, що надають доступ до віртуальних і дистанційно керованих пристроїв, а також до дослідження різних технологій проектування. Таким чином, вони дозволяють ефективно організовувати спільне використання різноманітного програмного забезпечення та інженерного обладнання з урахуванням фінансово-економічних обмежень. Дослідження показали, що на сьогоднішній день в галузі електроніки та робототехніки представлено значну кількість віртуальних та віддалених лабораторій.

Наприклад, гібридна віддалена лабораторія GOLDI (Grid of Online Lab Devices Imenau) (рис.1.6), яку розроблено на кафедрі інтегрованих систем зв'язку Технологічного університету Ільменау (Німеччина) [37]. Вона надає набір інструментів, що підтримує всі етапи розв'язку складних завдань керування (наприклад, у галузях машинобудування, робототехніки, телекомунікацій).

Віддалена лабораторія WebLab-Deusto (рис.1.7) створена на інженерному факультеті Університету Деусто (Іспанія) [38]. Вона може бути завантажена і розгорнута для обслуговування різноманітних дистанційних експериментів в різних середовищах і операційні системах з метою вивчення робототехніки та електроніки.

Labshare – мережа лабораторій з віддаленим доступом п'яти технічних університетів (Австралія) [39]. Експериментальні стенди дозволяють розв'язувати різні завдання у галузі фізики, робототехніки, телекомунікацій.

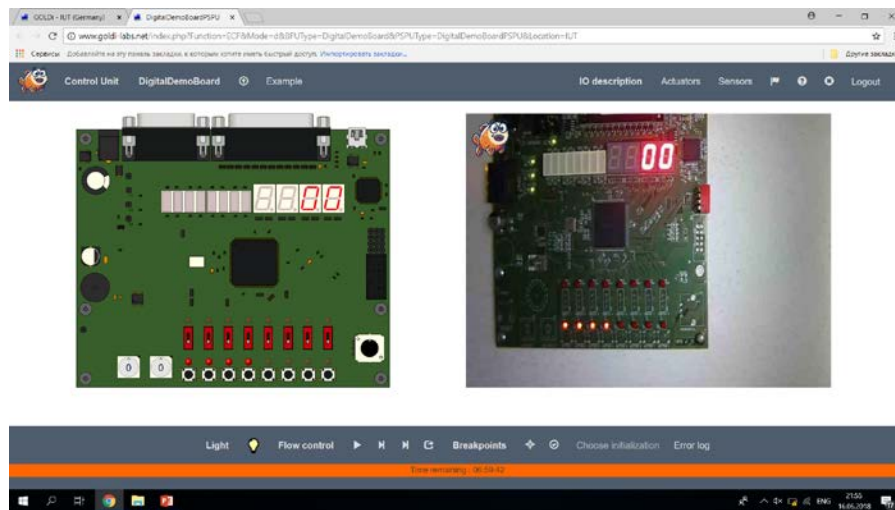


Рис.1.6. Сторінка інтерфейсу GOLDI

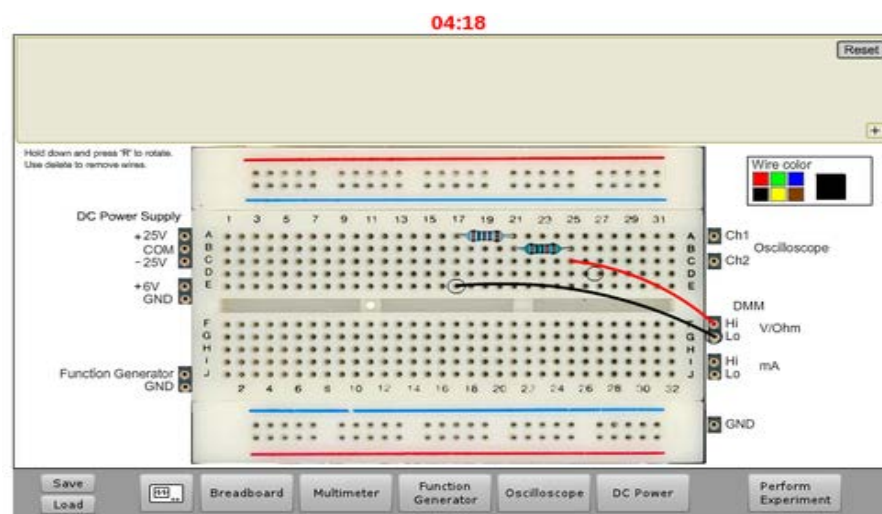


Рис.1.7. Сторінка інтерфейсу Web lab Deusto

e-Laboratory Project (Чехія та Словаччина) – віддалена лабораторія, що надає доступ до різноманітних експериментів в галузі електроніки та фізики [40].

iLabs – он-лайн лабораторія, створена в Массачусетському технологічному інституті, що пропонує експерименти у галузі мікроелектроніки, проектування конструкцій та обробки сигналів [41].

Як показали дослідження [47, 48, 50], проблема полягає в тому, що більшість існуючих віддалених лабораторій (ВЛ) є суто освітніми ресурсами. Для проектних і виробничих цілей ці лабораторії потребують подальшого розвитку. Тому, завдання створення та впровадження ВЛ в процес проектування ВС з метою зниження витрат на розробку виробів за рахунок



спільного використання обладнання та програмного забезпечення залишається актуальним.

На сьогоднішній день, серед існуючих інформаційно-пошукових систем можливо виділити такий популярний інструмент аналізу даних та надання рекомендацій, як рекомендаційні системи (РС).

Існуючі РС електронних компонентів можна умовно поділити на дві групи: сайти з продажу та сайти розробників. Як показали проведені дослідження, українські сервіси з продажу [33,34] використовують просте групування елементів за назвою виробника та призначенням. Сайти розробників електронних компонентів (Arduino [99], Raspberry Pi [100]) пропонують структуровану інформацію про свою продукцію. Наприклад, на сайті компанії Arduino виконується просте групування платформ, модулів, додаткових шилдів за допомогою поділу їх на класи або групи відповідно до галузей застосування. Виробник Raspberry Pi поділяє продукцію за такими категоріями: платформи, аксесуари (різні сенсори та актуатори), промислові модулі.

Сервіс TIDesigns від компанії Texas Instruments також є прикладом рекомендаційної системи в області електронних компонентів [35]. Він забезпечує фільтрацію схем та елементів компанії відповідно до вхідних параметрів користувача, пошук сумісної продукції компанії та ін. Крім того, для своєї платформи LaunchPad компанія Texas Instruments пропонує інструмент BoosterPack Checker (рис.1.8) для перевірки її сумісності з додатковими модулями компанії BoosterPack. Фільтрація компонентів виконується за сумісністю по кількості контактних пінів, встановлених на платформі LaunchPad та додатковому модулі BoosterPack.

Інший, не менш відомий виробник електроніки STMicrocontrollers [36], на основі вибраної налагоджувальної платформи пропонує перелік можливих сумісних платформ в рамках своєї продукції, а також всі супутні матеріали.

Microchip пропонує інструмент Parametric Search Tool [101] для пошуку Atmel та Microchip мікроконтролерів і мікропроцесорів за введеними

користувачем параметрами. Інтерфейс Parametric Search Tool представлений на рис. 1.9. Параметри, за якими виконується пошук, згруповані та представлені у вигляді списків (наприклад, є група параметрів «сімейство», «тип процесору» та інші). За кожним введеним пошуковим параметром система оновлює результат.

Базуючись на проведеному аналізі різних ресурсів з продажу електронних компонентів та сайтів розробників електроніки, було зроблено висновок, що сайти з продажу лише упорядковують інформацію, що безумовно спрощує її пошук, але не надають рекомендацій щодо вибору.

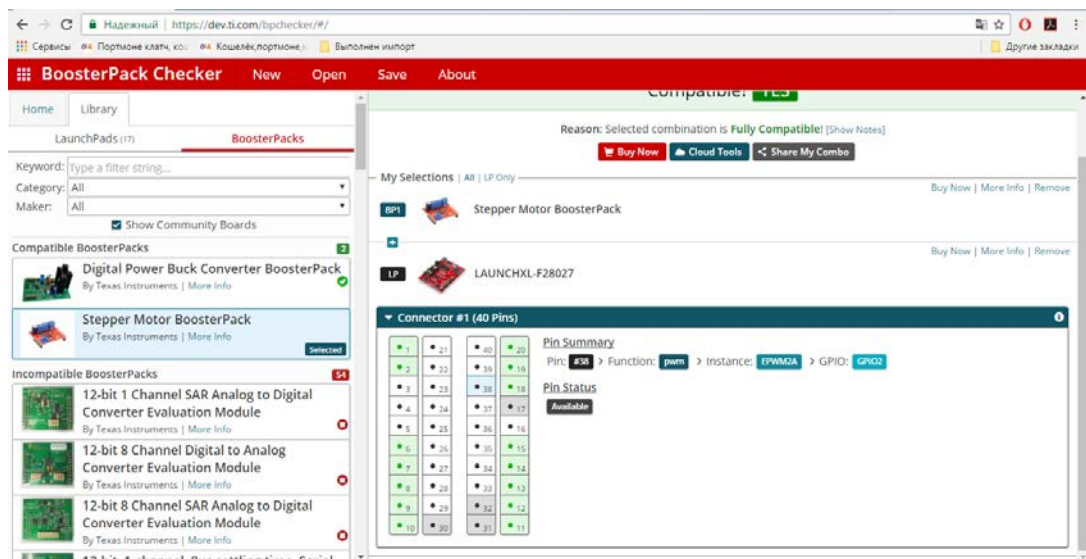


Рис. 1.8. Інструмент TI BoosterPack Checker

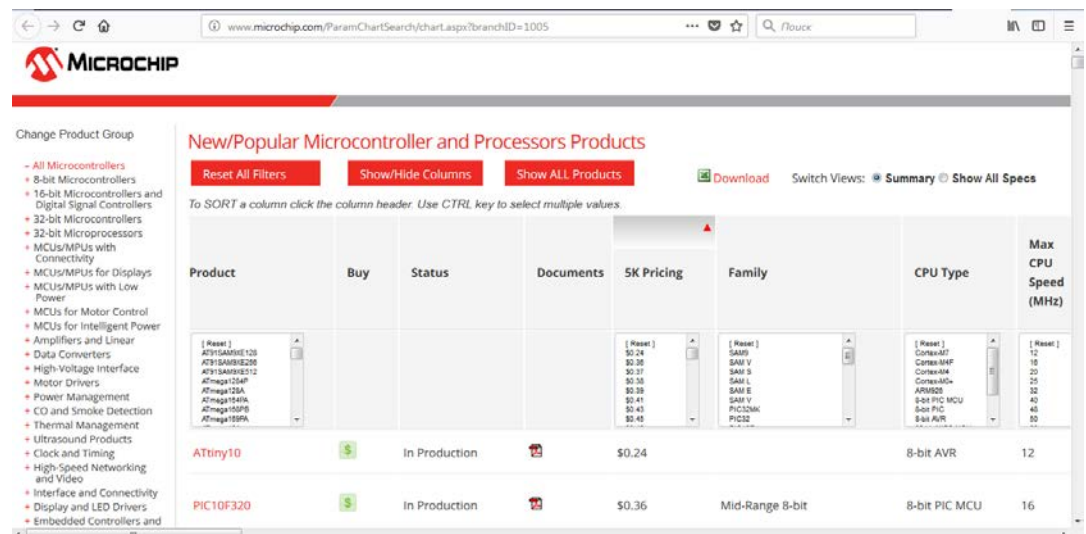


Рис. 1.9. Інтерфейс Parametric Search Tool

Виробники електроніки пропонують деякі рекомендації, але при цьому концентруються на власній продукції, з метою її просування, та допомагають обрати додаткові можливості для розширення функціоналу своєї базової платформи, що не стане у пригоді проектувальнику при виборі саме базової платформи. Отже, інформація, яка пропонується на сайтах виробників не завжди дозволяє правильно обрати проектне рішення та оптимально реалізувати проект.

Тому, проектувальникам, особливо початківцям, потрібен інструментарій для ефективного вибору апаратно-програмної платформи, виходячи з вимог до проектованої системи. Впровадження рекомендаційної системи, яка допоможе розробнику працювати з вимогами та виконувати вибір апаратно-програмної платформи на початкових етапах проектування ВС УРО, сприятиме скороченню часу пошуку проектного рішення.

Таким чином, розробка та впровадження засобів віддаленої інженерії (віддалених лабораторій, рекомендаційних систем) при АП ВС дозволить розширити функціональні можливості існуючих десктопних програмних систем, організувати спільне використання обладнання та програмного забезпечення, підвищити рівень автоматизації проектних робіт, та, таким чином, прискорити процес проектування ВС та знизити собівартість проектованої продукції.

## **1.6 Висновки до розділу 1**

1. Виконано аналіз сучасного стану розвитку ВС та класифікацію ВС за галузями застосування. Проведено дослідження існуючих вбудованих систем дистанційного управління наземними колісними рухомими об'єктами, яке дозволило виявити їх переваги та недоліки, а також особливості розробки та використання.

2. Проведено дослідження структурних та функціональних

особливостей ВС УРО, що дало змогу визначити комплекс вимог до ВС УРО, який має бути врахований при автоматизованому проектуванні.

3. Проведено детальний аналіз технологій реалізації ВС, який показав, що використання МК та апаратно-програмних платформ на базі МК при проектуванні ВС дає низку переваг. Крім того, в результаті аналізу було виявлено проблему швидкого вибору відповідної вимогам проекту апаратно-програмної платформи серед великої кількості пропонуваніх в мережі Інтернет рішень.

4. Виконано аналіз методів проектування ВС, в результаті якого виявлено недоліки традиційного підходу до проектування ВС та зроблено висновок, що сумісне використання паралельного та платформно-орієнтованого методів проектування, а також методології повторного використання сприятиме підвищенню ефективності процесу автоматизованого проектування.

5. Виконано аналіз існуючих на сьогоднішній день стандартів, а також робіт в галузі формування та аналізу вимог до ВС, який показав, що ці джерела містять інформацію щодо окремих аспектів розв'язуваного завдання, які не враховують всіх особливостей проектуваних вбудованих систем.

6. Проведено дослідження десктопних та он-лайн програмних систем для автоматизації проектування вбудованих систем, яке показало, що існують різноманітні програмні інструменти для розв'язку проектних завдань на різних рівнях проектування ВС. У той же час, виявлено відсутність засобів для програмної підтримки перехідного етапу між системним рівнем, на якому проводиться робота з вимогами до проектованої ВС та функціонально-логічним рівнем проектування.

7. Показано, що використання засобів віддаленої інженерії, а саме рекомендаційних систем та віддалених лабораторій, є перспективним напрямом з точки зору підвищення ефективності проектування ВС.

Результати розділу опубліковано у наукових роботах [42, 44, 46-48, 50, 54, 56-59, 65].

## **РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ ЗАСОБАМИ ВІДДАЛЕНОЇ ІНЖЕНЕРІЇ**

У цьому розділі, відповідно до мети та завдань дисертаційної роботи удосконалено модель формування вимог до вбудованих систем та метод роботи з вимогами, які дозволяють врахувати структурні особливості ВС, як сукупності апаратних та програмних компонентів та дають змогу розподіляти вимоги між ними при автоматизованому проектуванні. Удосконалено метод проектування ВС на основі паралельного та платформно-орієнтованого підходів до проектування ВС, методології повторного використання апаратних та програмних компонентів, а також засобів віддаленої інженерії. Розроблено метод автоматизованого проектування вбудованих систем з використанням технологій віддаленої інженерії, що дозволяє організувати спільне використання обладнання та програмного забезпечення для прототипування проекрованої ВС.

### **2.1 Модель формування вимог до вбудованої системи**

Грунтуючись на виконаному аналізі існуючих описових моделей вимог та стандартів, що надають різні типи вимог, була запропонована удосконалена модель формування вимог до ВС [45,46]. У формальному вигляді цю модель можна представити так:

$$S = \{R, L, C\}, \quad (2.1)$$

де  $R = \{r_1, r_2, \dots, r_m\}$  – множина вимог;  $L = \{l_1, l_2, \dots, l_n\}$  – множина рівнів вимог;  $C = \{c_1, c_2, \dots, c_k\}$  – множина зв'язків між вимогами.

В структурованому вигляді модель формування вимог до ВС наведена на рис. 2.1. Вона, як і моделі вимог Вігерса і Леффінгвелла [5,6], має 3 рівня, проте, на відміну від запропонованих, враховує структурні особливості ВС як сукупності апаратних та програмних компонентів та показує процес розподілу вимог на вимоги до програмного та апаратного забезпечення системи.

У запропонованій моделі формування вимог виділені наступні рівні:

- рівень користувача – вимоги всіх зацікавлених осіб проекту, (наприклад, таких як, клієнт, кінцевий користувач продукту), можуть також вказуватись розробники і всі особи, що задіяні у проекті. Можуть бути представлені: бізнес-вимоги, варіанти використання, атрибути якості.

- рівень системи – вимоги до системи, що включають в себе комбінацію взаємопов'язаних компонентів (АЗ, ПЗ, користувачі та ін.). Можуть бути представлені системні вимоги (до системи в цілому) та нефункціональні вимоги (обмеження, інтерфейси, конструктивні особливості, вимоги до умов експлуатації).

- функціональний рівень – функції (функціональні характеристики), розподілені між програмними і апаратними компонентами системи.

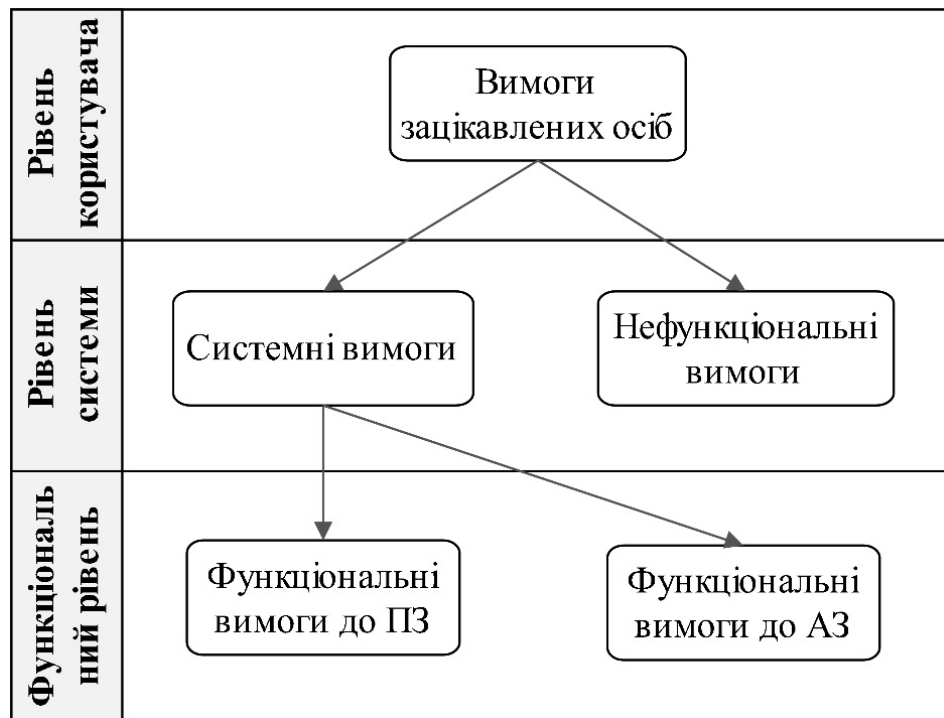


Рис. 2.1. Структура моделі формування вимог до ВС

Можна також підкреслити такі відмінності запропонованої моделі від моделі Вігерса:

- на верхньому рівні знаходяться лише вимоги зацікавлених осіб;
- на введеному другому рівні розташовуються системні вимоги до розроблюваної ВС, а також визначаються нефункціональні вимоги (вимоги до інтерфейсів та системні обмеження);
- на нижньому рівні, з урахуванням особливостей ВС, відбувається декомпозиція функціональних вимог на функціональні вимоги до програмного та апаратного забезпечення системи.

У табл. 2.1 представлено відмінні особливості запропонованої моделі формування вимог [46] від розглянутих вище моделей Вігерса і Леффінгвелла.

Таблиця 2.1.

Відмінні особливості моделей вимог

	<b>Запропонована модель</b>	<b>Модель Вігерса</b>	<b>Модель Леффінгвелла</b>
<b>Рівні</b>	- користувача - системи - функціональний	- бізнес-вимог - вимог користувачів - функціональний	- потреб - функцій - програмних вимог
<b>Типи</b>	- апаратні - програмні	- функціональні - нефункціональні	
<b>Інтерфейси</b>	Користувача, ПЗ/ПЗ, АЗ/ПЗ, АЗ/АЗ, ПЗ(АЗ)/користувач.	Користувача, ПЗ/ПЗ, ПЗ/користувач.	Інтерфейс користувача.

Таким чином, удосконалено модель формування вимог до ВС, що описує всі необхідні типи вимог до ВС, враховує специфічні особливості ВС, як сукупності апаратного і програмного забезпечення, та, відповідно, розподіляє вимоги між програмними та апаратними компонентами. Це дозволяє більш повно враховувати вимоги до ВС при АП.

## 2.2 Метод роботи з вимогами

Для розробленої моделі формування вимог пропонується метод роботи з вимогами до ВС [46], діаграма послідовності етапів якого наведена на рис. 2.2. Метод дозволяє розв'язати завдання визначення, аналізу і формування вимог до програмних і апаратних компонентів ВС на етапі високорівневого проектування.

Перш за все виконується визначення зацікавлених осіб, поділ їх на групи та фіксування вимог цих осіб у довільній формі.

Наступним етапом є розробка варіантів використання (сценаріїв) системи за допомогою опису робочих процесів користувачів.

Далі визначаються вимог до системи, з урахуванням оточення, та нефункціональні вимоги (атрибути якості, обмеження, вимоги до умов експлуатації, інтерфейси).

Визначення функціональних вимог до АЗ і ПЗ має на увазі більш детальне виявлення та перетворення вимог користувача в форму, корисну для розробників. Як для АЗ так і для ПЗ передбачається побудова архітектури та розподілення вимог за компонентами системи. Фіксування функціональних вимог до програмного забезпечення в специфікації вимог до ПЗ може виконуватися за допомогою подання очікуваної поведінки системи у формі «подія-реакція». Така форма є зручною для подальшої розробки варіантів тестування.

Після кожного етапу визначення вимог виконується перевірка розроблених специфікацій вимог на наявність помилок, повноту, недвозначність, несуперечливість та ін.

Робота з вимогами представляє собою ітеративний процес. Перший етап зазвичай виконується один раз (але при необхідності переглядається і змінюється), інші етапи повторюються для кожної нової версії системи.



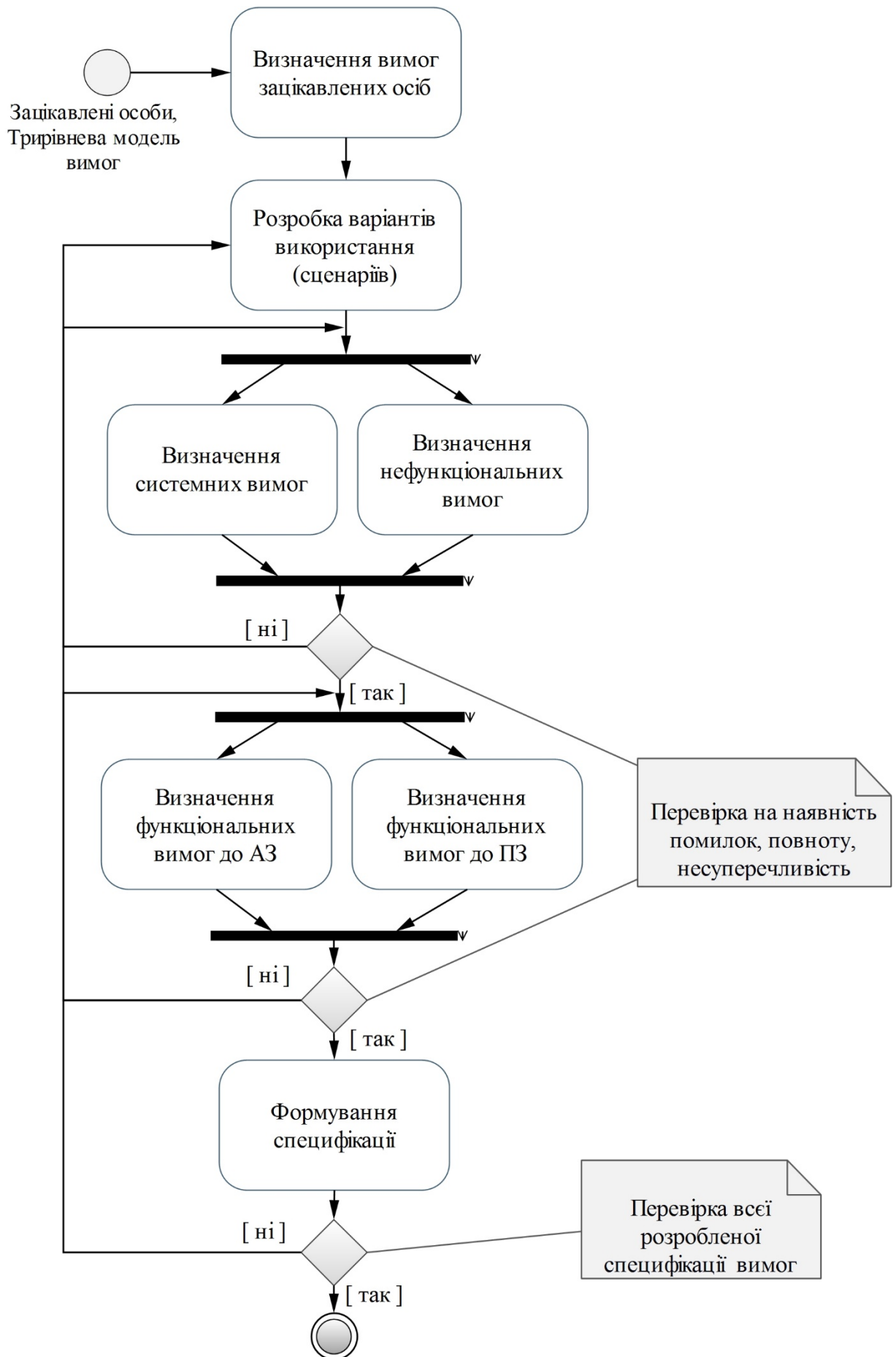


Рис. 2.2. UML діаграма послідовності етапів методу роботи з вимогами.

Створювана матриця трасування в IBM Rational RequisitePro [44] дозволяє при модифікації одних вимог легко визначити, яких вимог торкається ця зміна і які необхідно перевірити. Трасування створюваних вимог відповідно до схеми розробленої моделі формування вимог в RequisitePro виглядає як на рис. 2.3.

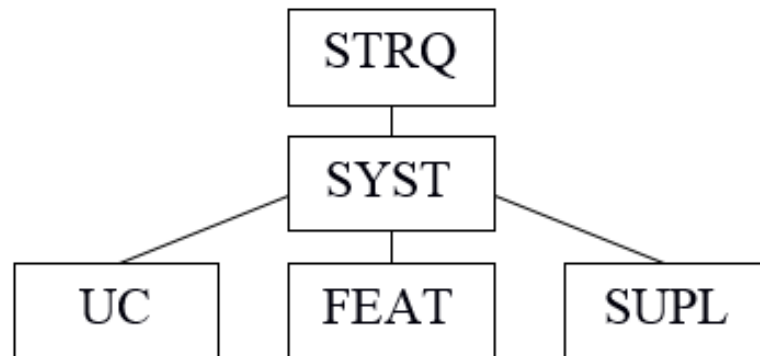


Рис. 2.3. Трасування вимог в RequisitePro

На верхньому рівні знаходяться Потреби зацікавлених осіб (ідентифікатор даного виду вимог STRQ), а також Системні вимоги (SYST), сформульовані з Потреб зацікавлених осіб. Сценарії використання (UC) представляють собою опис поведінки системи в термінах послідовності дій. Функціональні вимоги – це вимоги до функціональних можливостей системи, які зазвичай формуються бізнес-аналітиком; призначення цих вимог – задовольнити потреби замовника. Функціональні вимоги (FEAT) розподіляються за різними видами забезпечення проекту на основі Системних вимог: FEAT\_HW – функціональні вимоги до АЗ; FEAT\_SW – функціональні вимоги до ПЗ. У якості додаткових вимог (SUPL) виступають інші вимоги (зазвичай нефункціональні), які не можуть бути охоплені сценаріями використання.

Таким чином, застосування удосконаленого методу роботи з вимогами дозволить підвищити ефективність виконання необхідних процесів роботи з вимогами, а саме визначення, аналіз і формування вимог до програмних і апаратних компонентів ВС при АП.

### 2.3 Методи автоматизованого проектування вбудованих систем на основі технологій віддаленої інженерії

Базуючись на проведених в розділі 1 дослідженнях, було прийнято рішення щодо доцільності удосконалення існуючих методів проектування вбудованих систем (рис. 1.5). Запропонований метод паралельного проектування відрізняється спільним застосуванням паралельного та платформно-орієнтованого підходів [42, 43], а також методології повторного використання готових програмних і апаратних рішень [55, 58] та засобів віддаленої інженерії. Удосконалена схема методу паралельного проектування представлена на рис. 2.4.

Використання платформно-орієнтованого підходу на основі методології повторного використання дозволяє значно скоротити час проектування завдяки повторному використанню попередньо реалізованих і протестованих програмних та апаратних компонентів системи.



Рис. 2.4. Удосконалена схема методу паралельного проектування ВС

Застосування технологій віддаленої інженерії дозволяє підвищити рівень автоматизації проектувальних робіт шляхом:

- автоматизації процесу вибору апаратно-програмної платформи на основі рекомендаційних методів;
- ефективної розробки програмного забезпечення ВС на основі методології повторного використання;
- швидкого прототипування проектованої системи на основі віддаленого експерименту.

Реалізація технологій віддаленої інженерії передбачає створення віддаленого серверу, на якому встановлене все необхідне програмне забезпечення та підключене обладнання для виконання проектних процедур при АП. Доступ до серверу забезпечується через мережу Інтернет з використанням будь-якого браузеру та електронного пристрою (комп'ютер, планшет, смартфон). Узагальнену схему реалізації технологій віддаленої інженерії зображено на рис. 2.5. Впровадження технологій віддаленої інженерії дозволяє знизити матеріальні витрати на проектування, оскільки дозволяє спільно використовувати обладнання та програмне забезпечення без його придбання, налаштування та обслуговування [47]. Це сприяє зменшенню витрат на проектування та, відповідно, собівартості проектованої продукції.

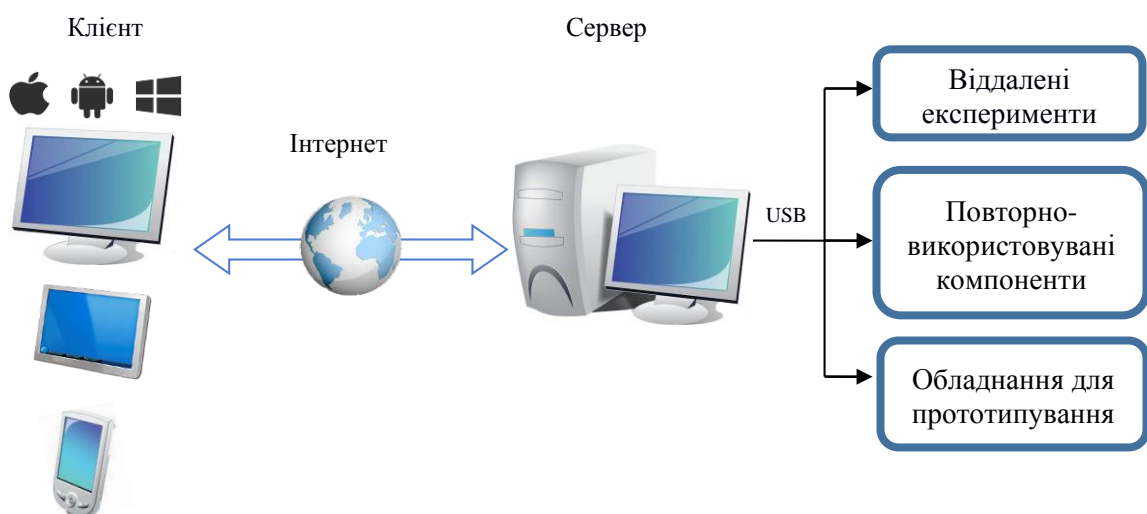


Рис. 2.5. Узагальнена схема організації технологій віддаленої інженерії

Як було встановлено в результаті досліджень, різноманітність та роз'єднаність описів вже існуючих рішень в галузі проектування ВС (апаратних та програмних компонентів, алгоритмів, бібліотек, API, апаратно-програмних платформ і т.і.) ускладнює їх повторне застосування. Використання технологій віддаленої інженерії дозволить проектувальнику швидко оцінити можливості готових апаратно-програмних платформ, прийняти рішення щодо їх інтеграції в проект та, таким чином, знизити ризики та часові витрати на проектування [50, 55].

Таким чином, за допомогою технологій віддаленої інженерії проектувальник має можливість:

- отримувати та аналізувати інформацію про специфікації готових апаратно-програмних платформ;
- виконувати вибір готових апаратно-програмних рішень;
- виконувати розробку та верифікацію програмного забезпечення;
- виконувати інтеграцію апаратного та програмного забезпечення ВС;
- виконувати дослідження прототипу проектованої системи;
- спостерігати проведення віддаленого експерименту на реальному обладнанні за допомогою веб-камери.

Розроблений метод автоматизованого проектування ВС на основі технологій віддаленої інженерії [52-54], відрізняється використанням інструментарію рекомендацій та віддаленого експерименту, та може бути представлений у вигляді UML діаграми послідовності етапів методу (рис. 2.6).

Запропонована трирівнева модель формування вимог до ВС лежить в основі методу АП ВС. Вона має бути редуційована з метою формування вимог для роботи рекомендаційної системи. Модель включає в себе перш за все відомості про зацікавлену особу (STRQ). Оскільки функціональність ВС розподіляється між апаратними і програмними складовими, то далі формуються функціональні апаратні та функціональні програмні вимоги (Feat\_HW, Feat\_SW). Останнім етапом є формування нефункціональних (SUPL) вимог та специфікації використання системи (UC).

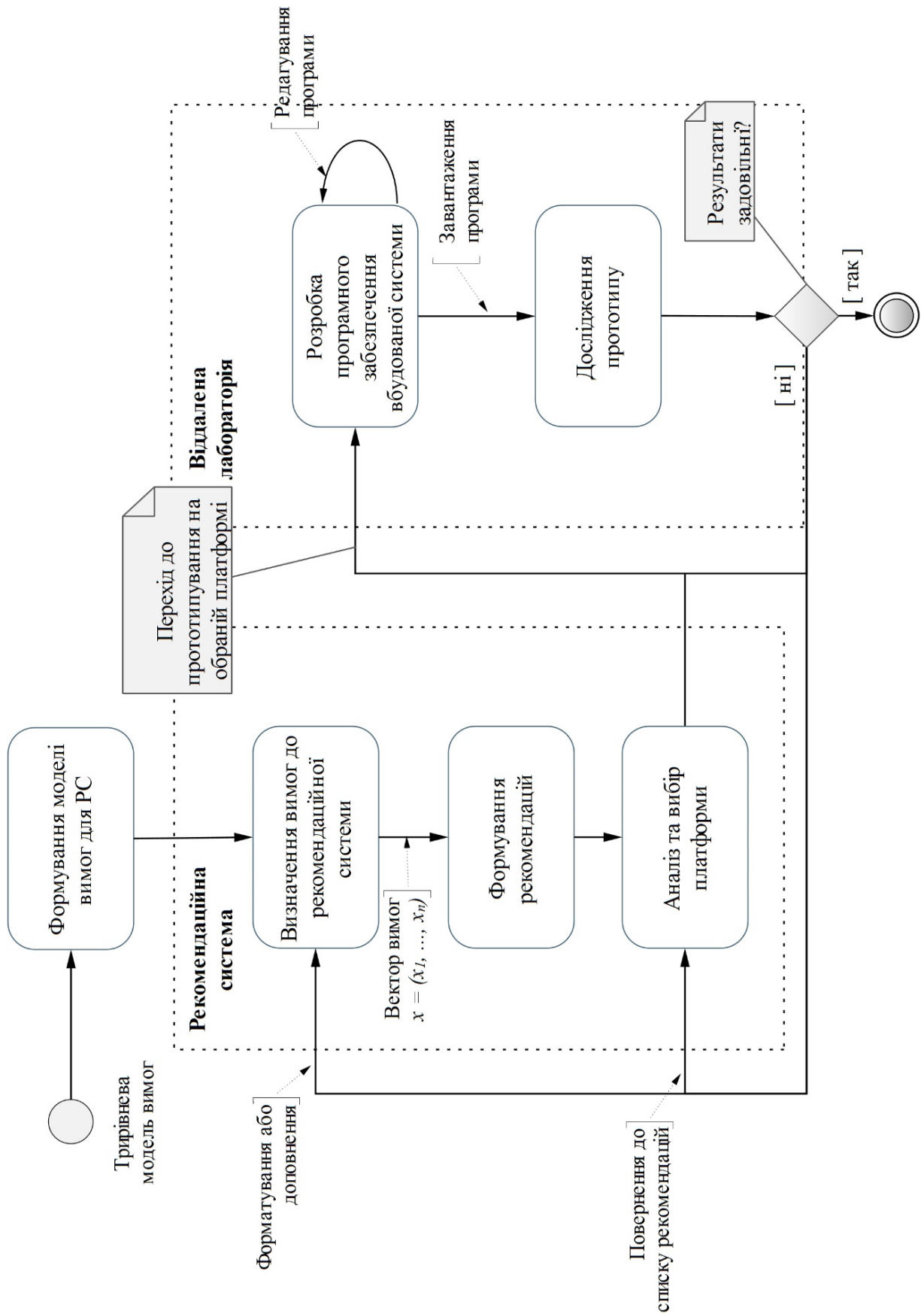


Рис. 2.6. UML діаграма послідовності етапів методу автоматизованого проектування ВС на основі технологій віддаленої інженерії

Для рекомендаційної системи сценарії УС не враховуються, оскільки ці специфікації представляють собою послідовність дій користувача відносно проєктованої системи, або послідовність поведінки самої системи. Для формування вектору вимог з окремо взятих критеріїв цей вид вимог не є зручним. Але УС відіграють значну роль саме на етапах верифікації системи для складання протоколів тестування. Тому, вони можуть бути враховані на інших етапах проєктування ВС, а у сформованій моделі формування вимог для РС можуть не фігурувати.

Наступним етапом методу є видобуток вимог для РС шляхом визначення потреб розробника (функціональних та не функціональних), з яких у подальшому отримуємо вектор вимог  $x = (x_1, x_2, \dots, x_n)$ , при  $x_n \in X$ , де  $X$  – множина усіх можливих вимог.

Для того, щоб визначитись з вимогами розробника, створено список запитів, на основі яких будуть сформульовані певні ознаки шуканої апаратно-програмної платформи для проєктування ВС:

- рівень знань розробника ВС;
- планована кількість контактів для підключення периферії;
- переваги щодо використання окремих сімейств процесорів;
- знання мови програмування;
- обмеження цінового діапазону та ін.

Відповідь на всі запити не є обов'язковою, але дозволяє більш детально врахувати вимоги розробника та виконати пошук відповідної апаратно-програмної платформи.

На основі цих відповідей формується вектор вимог до платформи. Якщо задані вимоги  $x_1, \dots, x_n$ , при  $x: X \rightarrow D_x$ , де  $D_x$  – множина допустимих значень вимог, то вектор опису вимог платформи можна представити у вигляді:

$$x = (x_1, \dots, x_n). \quad (2.2)$$

Матриця вимог розмірністю  $l \times n$  усіх апаратно-програмних платформ має вигляд:

$$F = \| y_i \|_{l \times n} = \begin{bmatrix} y_1 & \dots & y_1 \\ \dots & \dots & \dots \\ y_l & \dots & y_l \end{bmatrix}. \quad (2.3)$$

Формування рекомендацій передбачає визначення тотожності між заданим розробником вектором вимог  $x = (x_1, \dots, x_n)$  та вектором властивостей апаратно-програмних платформ  $y = (y_1, \dots, y_n)$ . На цьому етапі реалізується метод формування рекомендацій щодо апаратно-програмної платформи [59], який наведено в підрозділі 3.3. В результаті реалізації методу, користувач отримує перелік платформ, які відповідають його вимогам. Це може бути одна, або декілька платформ, в залежності від повноти введених вимог.

Після того, як розробник отримав рекомендації, що надані системою, він аналізує їх та обирає апаратно-програмну платформу для подальшої роботи з нею у віддаленій лабораторії.

Інформаційна підсистема віддаленої лабораторії надає доступ до типових рішень (сценарії проведення віддалених експериментів, схеми підключення сенсорів та актуаторів, фрагменти програмного коду) для обраної апаратно-програмної платформи. За допомогою засобів віддаленої лабораторії проектувальник може також розробити власний програмний код та перевірити його на наявність помилок.

Швидке прототипування та дослідження прототипу ВС на основі обраної платформи за допомогою віддаленої лабораторії передбачає завантаження відлагодженої програми до контролеру обраної платформи та проведення віддаленого експерименту у відповідності з обраним сценарієм.

Якщо результати дослідження прототипу задовільні, виконується завершення роботи з віддаленою лабораторією. У випадку отримання незадовільних результатів, в залежності від ступеня невідповідності результатів прототипування вимогам до проектованої ВС, розробник може повернутися на етап визначення вимог, де він може корегувати їх, або до списку рекомендацій, з метою обрання іншої рекомендованої платформи, або ж до корегування програми та повторного дослідження прототипу ВС.



Таким чином, впровадження методу паралельного проектування вбудованих систем на основі спільного застосування паралельного та платформно-орієнтованого підходів, повторного використання готових програмних і апаратних рішень, а також засобів віддаленої інженерії, дає змогу зменшити час переходу між системним та функціонально-логічним рівнями при проектуванні ВС та підвищити рівень автоматизації проектувальних робіт шляхом автоматизованого вибору апаратно-програмної платформи.

Запропонований метод автоматизованого проектування вбудованих систем на основі використання технологій віддаленої інженерії дозволяє організувати спільне використання обладнання та програмного забезпечення (без необхідності їх придбання та налаштування) для прототипування проектованої ВС та, таким чином, знизити собівартість проектованої продукції.

## **2.4 Висновки до розділу 2**

1. Удосконалено модель формування вимог до ВС на основі існуючих моделей вимог Вігера і Леффінгвелла, а також стандартів, що описують різні типи вимог. В даній моделі реалізовані три рівні та два типи вимог для системи в цілому, а також для її програмного та апаратного забезпечення. Це дозволяє найбільш повно врахувати специфічні особливості ВС та вимоги до проектованих ВС при автоматизованому проектуванні.

2. Представлено удосконалений метод роботи з вимогами, що дозволяє організувати поетапне визначення, аналіз і формування вимог з урахуванням особливостей ВС та, таким чином, підвищити ефективність виконання необхідних процесів роботи з вимогами при автоматизованому проектуванні.

3. Запропоновано метод автоматизованого проектування вбудованих систем, що базується на спільному застосуванні паралельного та платформно-орієнтованого підходів до проектування, методології повторного

використання програмних та апаратних компонентів, а також технологій віддаленої інженерії, а саме інструментарію рекомендацій та віддаленого експерименту. Використання зазначеного інструментарію дозволить користувачам: отримувати і аналізувати інформацію про специфікації готових апаратних і програмних компонентів для їх повторного використання, виконувати вибір апаратно-програмних платформ та розробку програмного коду, проводити дослідження прототипу системи на сумісність та працездатність апаратної і програмної частин на базі обладнання віддаленої лабораторії та програмного забезпечення рекомендаційної системи. Таким чином, підвищується ефективність автоматизованого проектування за рахунок зниження витрат на проектування, скорочення термінів виконання проектних робіт та підвищення рівня автоматизації проектувальних робіт.

Результати розділу опубліковано у наукових роботах [42-47, 50, 52-55, 58, 59].

## **РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ПРИ АВТОМАТИЗОВАНОМУ ПРОЕКТУВАННІ ВБУДОВАНИХ СИСТЕМ**

В цьому розділі, в рамках рішення завдань дисертаційної роботи, обґрунтовано вибір рекомендаційного методу, на основі якого будуть формуватися рекомендації щодо апаратно-програмних платформ при автоматизованому проектуванні ВС. Отримав подальший розвиток метод формування рекомендацій, на основі якого буде створено рекомендаційну систему. Розроблено математичне забезпечення формування рекомендацій при АП ВС на основі математичних методів визначення подібності, жорстких обмежень та багатокритеріального аналізу.

### **3.1 Обґрунтування вибору рекомендаційного методу**

Для спрощення пошуку та підвищення ефективності вибору готових рішень в мережі Інтернет існують різноманітні рекомендаційні технології. Їх сутність полягає у передбаченні того, що може бути корисним для того чи іншого користувача, базуючись на оцінках інших користувачів зі схожими інтересами, а також з урахуванням контентної інформації користувача, історії його попередньої поведінки та т.і.

В залежності від принципів роботи з даними, існуючі рекомендаційні методи можна розділити на три групи [27,29,31,102-104] (рис.3.1): колаборативна фільтрація (collaborative), контентна фільтрація (content-based), фільтрація на основі знань (knowledge-based). Варто також відокремити гібридну фільтрацію (hybrid), яка поєднує в собі методи різних груп, з метою усунення недоліків існуючих. Існують вузько направлені рекомендаційні

методи [29], що не підходять для рішення завдання надання рекомендацій щодо апаратно-програмних платформ, оскільки застосовуються для розв'язку більш конкретних специфічних завдань.

Проведені дослідження свідчать, що колаборативна фільтрація (КлФ) є найбільш популярною [27] і широко використовуваною для комерційних рекомендаційних систем [30,105]. КлФ виконується довкола реакції користувача щодо об'єкту (оцінки, лайки, коментарі та т.і.). Існують два підходи до побудови колаборативної РС [32]. У першому підході для побудови рекомендацій використовується інформація інших користувачів зі схожими інтересами (User-Based Collaborative Filtering). Альтернативним є підхід, коли замість користувачів порівнюються самі об'єкти рекомендацій на основі оцінок користувача (Item-Based Collaborative Filtering). КлФ передбачає, що користувачі, які мали однаковий інтерес до одних об'єктів, матимуть схожий інтерес до інших. Подібність користувачів (їх оцінок) розраховується за допомогою наступних рівнянь: косинус подібність, евклідова відстань, коефіцієнт кореляції Пірсона. Найбільш широко використовуваним для КлФ є алгоритм k-найближчих сусідів.

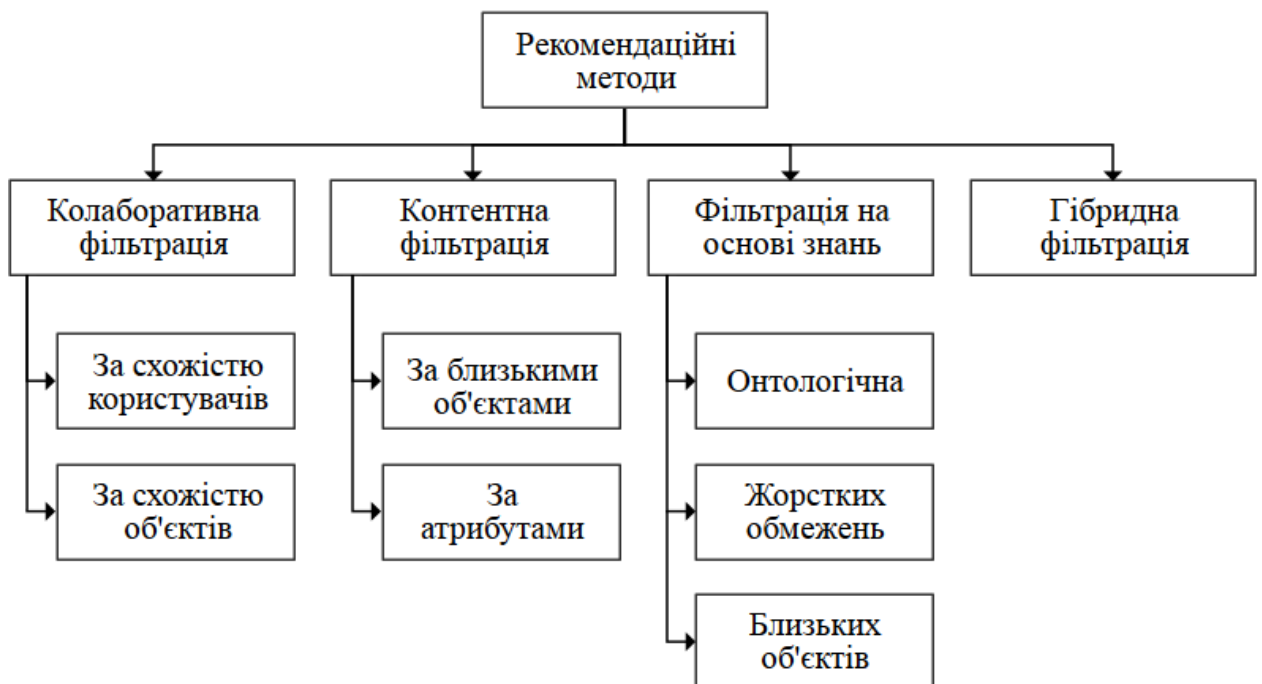


Рис. 3.1. Класифікація рекомендаційних методів

Контентна фільтрація, або фільтрація на основі змісту, на відміну від КлФ, базується на властивостях рекомендованих об'єктів та характеристиках користувача (його вподобаннях у минулому), та ігнорує вподобання інших користувачів [29, 103, 104]. Цей метод працює для надання рекомендацій щодо книг, документів, веб-сторінок та інших об'єктів з текстовою інформацією. Системи рекомендацій на основі контенту використовують наступні засоби: опису елементів, які можуть бути рекомендовані; створення профілю користувача, в якому описані його інтереси; порівняння елементів і профілю користувача для надання рекомендацій. Контентна фільтрація може здійснюватися за близькими об'єктами (рекомендує об'єкти, які близькі до тих, що сподобались користувачу у минулому) та за атрибутами (рекомендує атрибути об'єкта відповідно до профілю користувача). В основі методу в основному використовуються відповідні ключові слова або модель Vector Space. Алгоритм TF-IDF використовується для виділення ключових слів у документі [106].

Крім того, варто виділити системи, що базуються на знаннях (knowledge-based), які орієнтуються на нагальні потреби користувача [29, 103, 104]. Це той випадок, коли користувач сам описує свої потреби, а система, використовуючи базу знань, шукає об'єкти, які відповідають потребі користувача. До систем на основі знань можна віднести системи на основі обмежень (constraint-based systems) та системи на основі вибору подібних об'єктів (case-based) [104]. Також, одним із різновидів таких систем є онтологічні систем (ontology-based) [32]. Причиною виникнення таких систем є різниця представлення знань різними людьми. Такі системи намагаються встановити уніфіковану мову представлення знань для подальшої можливості роботи з ними, та, таким чином, звести до мінімуму концептуальну та термінологічну плутанини.

Гібридна фільтрація (hybrid filtering) [28] поєднує у собі декілька підходів для усунення недоліків попередніх та отримання найкращого результату.

Таким чином, колаборативна та контентна фільтрації для надання рекомендацій будуються довкола рейтингів та оцінок елементів, виставлених користувачами у минулому. Для цього необхідно організувати опитування користувачів щодо уподобання обраного рішення, а також зберігання інформації у системі. Це не є зручним для систем, які не зберігають дані про користувача та його поведінку, працюють з поточними даними та допомагають визначитись зі специфічними вимогами кожного користувача.

Отже, фільтрація на основі знань зазвичай застосовується у системах так званого одноразового використання. Для рекомендаційних систем на основі методу знань не передбачено зберігання поведінки користувача та його оцінок щодо об'єктів, а система працює з нагальною потребою користувача, що відповідає поставленому завданню вибору апаратно-програмної платформи при АП ВС.

### **3.2 Модель представлення знань щодо апаратно-програмних платформ для рекомендаційної системи**

Завдання розробки бази знань РС потребує реалізації процедур видобутку, структурування та представлення знань.

Можна виділити три основні групи методів видобутку знань: комунікативні методи, текстові методи та вилучення знань з даних (з нейронних мереж, дерев рішень, тощо) [107].

Елементами розроблюваної рекомендаційної системи є апаратно-програмні платформи, інформація про які знаходиться у специфікаціях. Для видобутку необхідних знань про апаратно-програмні платформи для розробки ВС, крім аналізу специфікацій платформ (текстовий метод), актуальним є опитування експертів в цій галузі, наприклад, на електронних форумах (комунікативний метод).

Після видобутку знання необхідно структурувати. Елементи

рекомендаційної системи найчастіше зберігаються у таблицях, де рядки – це елементи, а стовпчики – властивості (атрибути, характеристики і т.і.) елементів [106]. Кожен атрибут елемента має відповідне задалегідь відоме значення. Це приклад того, як можна структурувати дані для подальшої роботи з ними. Інший приклад збереження даних елемента - це простий текст у довільному форматі. Але, у цьому випадку, робота з такими даними ускладнюється, оскільки у контексті одне і те саме слово може мати різні значення, або два різних слова мати одне і те саме значення. Таким чином, коли представлення елемента включає у себе велику кількість різної інформації, найкраще зберігати такий елемент у формі таблиці, де одні атрибути представлені з обмеженою кількістю значень, а інші, як текст у довільному форматі. Довільний текст, в свою чергу, також потім може розглядатися, як структурований, де кожне слово може буде розглянуте як атрибут з булевим значенням.

Наступним етапом роботи є представлення знань на основі обраної моделі: семантична мережа, асоціативні правила, дерева рішень та інші. Можна підкреслити такі переваги семантичних мереж: наочність знань; великі можливості мережних моделей; невелика та добре формалізована сукупність взаємин між поняттями та подіями [107].

Тому, розв'язок завдання представлення знань для рекомендаційної системи пропонується на основі використання семантичної мережі, яку формально можна представити у вигляді [107]:

$$H = \langle I, C_1, C_2, \dots, C_n, G \rangle, \quad (3.1)$$

де  $I$  – множина інформаційних одиниць (або параметри апаратно-програмних платформ, що беруться зі словника предметної області);  $\{C_1, C_2, \dots, C_n\}$  – множина типів зв'язків між інформаційними одиницями;  $G$  – відображення, що задає взаємодії між інформаційними одиницями.

Після аналізу специфікацій платформ та експертних знань, були виділені головні інформаційні одиниці апаратно-програмних платформ, які використовувались для побудови словника предметної області, а в подальшому – для рекомендаційного пошуку. Словник включає наступні поняття: апаратно-програмна платформа; атрибути; сімейство процесорів; форм-фактор; живлення; аналогові входи/виходи; цифрові входи/виходи; ціна.

Як результат, отримуємо схему семантичної мережі знань, що відображає інформацію щодо апаратно-програмних платформ, фрагмент якої наведено на рис. 3.2.

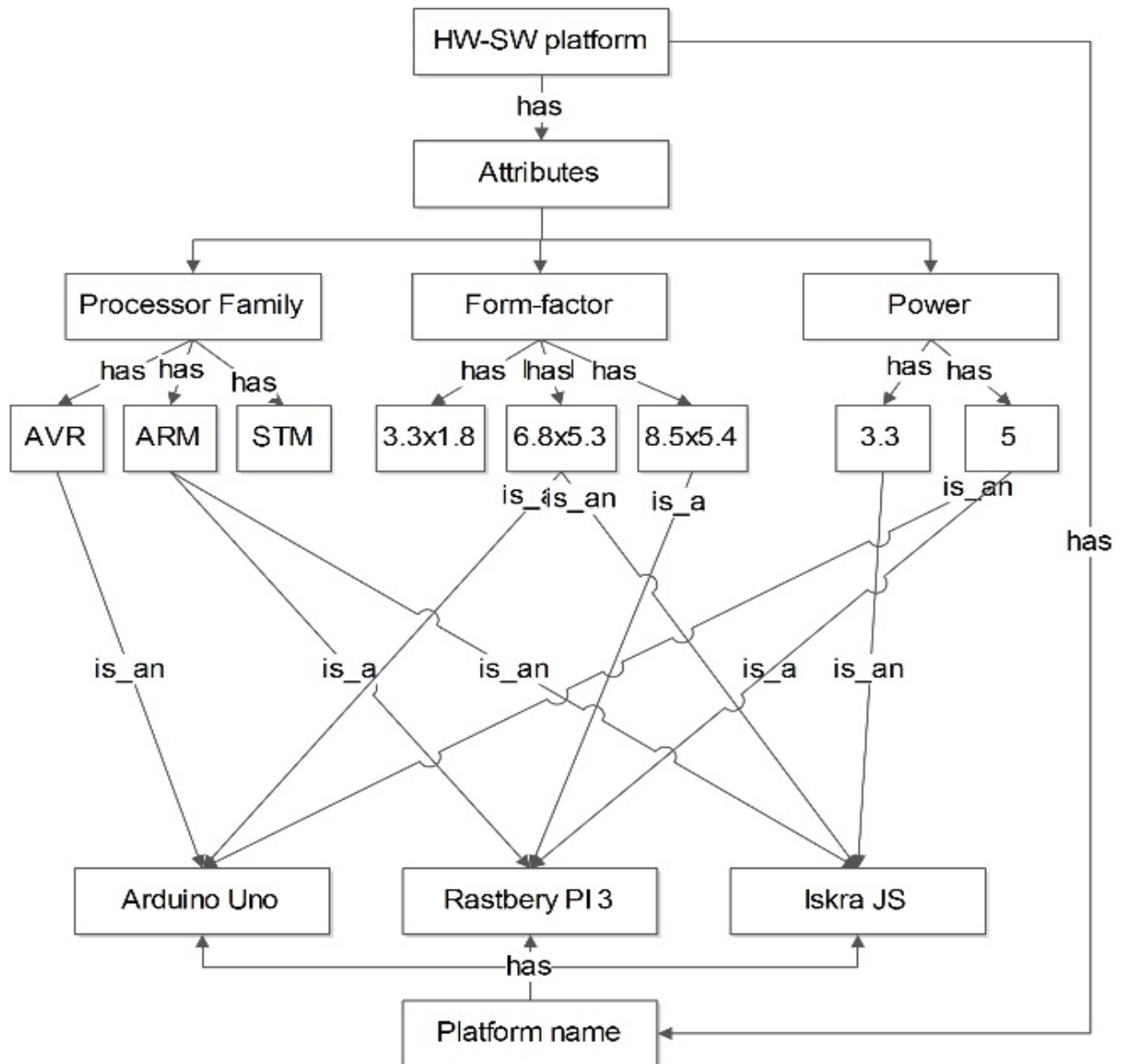


Рис. 3.2. Схема фрагменту побудованої семантичної мережі



Базуючись на трирівневій моделі формування вимог [46], яка запропонована у підрозділі 2.1, виходячи з аналізу специфікацій та експертних знань щодо особливостей апаратно-програмних платформ [42,43], представлених у підрозділі 1.2, до бази знань (БЗ) були додані додаткові інформаційні одиниці. Повний перелік видобутих інформаційних одиниць (надалі вимог) для роботи рекомендаційної системи, які згруповані за рівнями, представлено у табл. 3.1.

Таблиця 3.1.

#### Вимоги для рекомендаційної системи

Назва рівня	Вимоги
Зацікавлена сторона (Stakeholder)	Рівень користувача.
Апаратний (Hardware)	Кількість аналогових входів; кількість цифрових входів; живлення; сімейство процесорів.
Програмний (Software)	Мова програмування.
Додатковий (Supplementary)	Ціна платформи; форм-фактор.

Таким чином, розроблено модель знань, що представлена як семантична мережа та відображає інформацію про апаратно-програмні платформи. Семантична мережа дає змогу визначити та структурувати основні поняття (атрибути) бази знань. Таким чином, вона може бути ефективно використана при розробці БЗ апаратно-програмних платформ рекомендаційної системи.

### 3.3 Математичний метод визначення подібності елементів

Для реалізації інструментарію надання рекомендацій використовуються різні математичні методи [29,59,105]. Всі вони зводяться до розв'язку завдання знаходження подібності. Метод відстаней — найбільш простий в реалізації. Сутність його полягає в тому, що за допомогою обраної

метрики (евклідова відстань, косинус подібності та інші) розраховується відстань між двома об'єктами.

Найпоширенішою мірою подібності між двома об'єктами є евклідова відстань [27]:

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}, \quad (3.2)$$

де  $n$  – кількість атрибутів;  $x_k, y_k$  –  $k$ -тий атрибут вектора об'єкта.

Косинус подібності є найбільш часто використовуваною мірою в колаборативній фільтрації [108] для знаходження подібності між двома об'єктами, представленими у вигляді векторів [29]:

$$\cos(x, y) = \frac{\sum_{k=1}^n x_k y_k}{\sqrt{\sum_{k=1}^n x_k^2} \sqrt{\sum_{k=1}^n y_k^2}}, \quad (3.3)$$

де  $n$  – кількість атрибутів;  $x_k, y_k$  –  $k$ -ий атрибут вектора об'єкта.

Отриманий результат сортується у порядку зростання та видається у формі рекомендації: найбільш близькі – найбільш бажані.

У [104] для методу фільтрації на основі знань розглядається міра подібності між об'єктами та вимогами користувача:

$$\text{similarity}(X, Y) = \frac{\sum_{y_n \in Y} w_{y_n} \cdot \text{sim}(x_n, y_n)}{\sum_{y_n \in Y} w_{y_n}}, \quad (3.4)$$

де  $X$  – множина атрибутів об'єкта;  $Y$  – множина вимог користувача;  $y_n$  – вимога;  $w_{y_n}$  – вага даної вимоги;  $\text{sim}(x_n, y_n)$  – міра подібності атрибуту об'єкта  $x_n$  до вимоги  $y_n$ .

Існує декілька варіантів знаходження міри подібності  $\text{sim}(x_n, y_n)$  чисельних атрибутів [104]:

– «більше – краще» (наприклад, чим більше аналогових та цифрових входів у платформи, тим краще),

$$sim(x, y) = \frac{\varphi_y(x) - min(y)}{max(y) - min(y)}, \quad (3.5)$$

– «менше – краще» (класичний приклад – ціна, чим вона нижче, тим краще),

$$sim(x, y) = \frac{max(y) - \varphi_y(x)}{max(y) - min(y)}, \quad (3.6)$$

– «ближче – краще» (класичний підхід у вигляді евклідового простору, коли для користувача не має різниці, більше чи менше; своєрідна модифікація жорстких правил до нечіткої форми),

$$sim(x, y) = 1 - \frac{|\varphi_y(x) - y|}{max(y) - min(y)}, \quad (3.7)$$

де  $sim(x, y)$  – міра подібності об'єкта  $x$  до вимоги  $y$ ,  $min(y)$  та  $max(y)$  – відповідно мінімальне та максимальне значення критеріїв,  $y$  – точне значення вимоги користувача,  $\varphi_y(x)$  – значення відповідного критерію у об'єкта  $x$ .

На основі аналізу ефективності функціонування існуючих методів визначення подібності [109-111], можна зробити висновок, що результати їх роботи значно різняться та залежать від особливостей поставленого завдання. N. Lathia та ін. [112] провели експериментальні дослідження, згідно з якими точність прогнозування РС не залежить від обраної метрики подібності. Однак, у випадку недостатньої інформації на вході РС або її суперечливості, можуть виникнути непередбачені проблеми при використанні вказаних вище методів визначення подібності. Тому, їх можна модифікувати шляхом обробки семантично виняткових випадків та/або шляхом використання елементів інших методів (наприклад, методу багатокритеріального аналізу [113]).

Таким чином, для розв'язку поставленого завдання визначення подібності елементів для реалізації рекомендаційного методу, заснованого на знаннях, доцільно використовувати метод відстаней та комбінацію варіантів знаходження міри подібності для чисельних атрибутів. Але, для урахування особливостей розв'язуваного завдання, ці міри подібності було модифіковано.

### **3.4 Метод формування рекомендацій щодо апаратно-програмних платформ**

Як показали проведені дослідження, колаборативна і контентна фільтрація не можуть використовуватись при розробці системи надання рекомендації щодо апаратно-програмних платформ, оскільки ця РС розв'язує специфічні завдання кожного окремого розробника і зазвичай не зберігає поведінки розробників та їх оцінок. Саме тому, вирішено використовувати метод на основі знань для формування рекомендацій та пошуку рішення нагальної вимоги розробника.

Метод формування рекомендацій щодо апаратно-програмних платформ використовує підхід на основі знань, який включає в себе методи жорстких обмежень та вибору близьких об'єктів, з метою створення більш гнучкої рекомендаційної системи. Ідея обох методів наступна: розробник формулює свої вимоги до об'єкту, система намагається знайти потрібний об'єкт, виходячи з цих вимог. У першому випадку рекомендуються тільки ті об'єкти, які точно відповідають всім вимогам розробника. У другому випадку можуть бути рекомендовані об'єкти з характеристиками, близькими до вимог (з використанням математичних методів пошуку подібності). Тому, було прийнято рішення про сумісне використання даних методів шляхом розділення критеріїв на жорсткі та такі, для яких буде виконуватись пошук найближчих об'єктів, оскільки це дозволить більш індивідуально врахувати кожен критерій.

Розгляд лише жорстких критеріїв не має сенсу, адже вони підкоряються (підлягають) правилу «ЯКЩО — ТО». Приклад: якщо користувач хоче використовувати просту платформу для рівня початківця, йому не слід пропонувати платформи рівня професіоналів.

Методи групи, заснованої на близькому виборі, мають найтісніший контакт з вимогами користувачів. В цьому випадку можлива рекомендація об'єктів, які частково задовольняють вимогам.

Розроблений рекомендаційний метод можна представити у вигляді UML діаграми послідовності етапів, які наведено на рис. 3.3 [59].

Вхідними даними для реалізації методу формування рекомендацій є вектор вимог розробника. Першим етапом методу є формування множини апаратно-програмних платформ згідно з заданим рівнем знань розробника, отриманого з вектору вимог. Система виконує фільтрацію апаратно-програмних платформ (далі - платформ) за рівнем знань розробника (жорсткий критерій – див.табл.3.1) та формує список можливих платформ:

- якщо обрано рівень «професіонал», першими надаються платформи, у яких критерій «рівень користувача»  $> 2$ ;
- якщо обрано рівень «початківець» та у всіх інших випадках, першими надаються платформи, у яких критерій «рівень користувача»  $< 2$ .

Після цього, виконується розрахунок подібності для гнучких критеріїв, а саме, визначається міра подібності між введеним вектором вимог розробника та атрибутами платформи за формулою (3.4).

До сформованої на першому етапі множини платформ розраховується міра подібності за «гнучкими» критеріями (табл. 3.1):

- якщо ці критерії не зазначено, повертається множина платформ, сформована на етапі 1;
- якщо введено критерій «кількість аналогових входів» чи «кількість цифрових входів», тоді за виразом (3.7) за мірою «більше — краще» визначаються платформи з множини, сформованої на етапі 1;

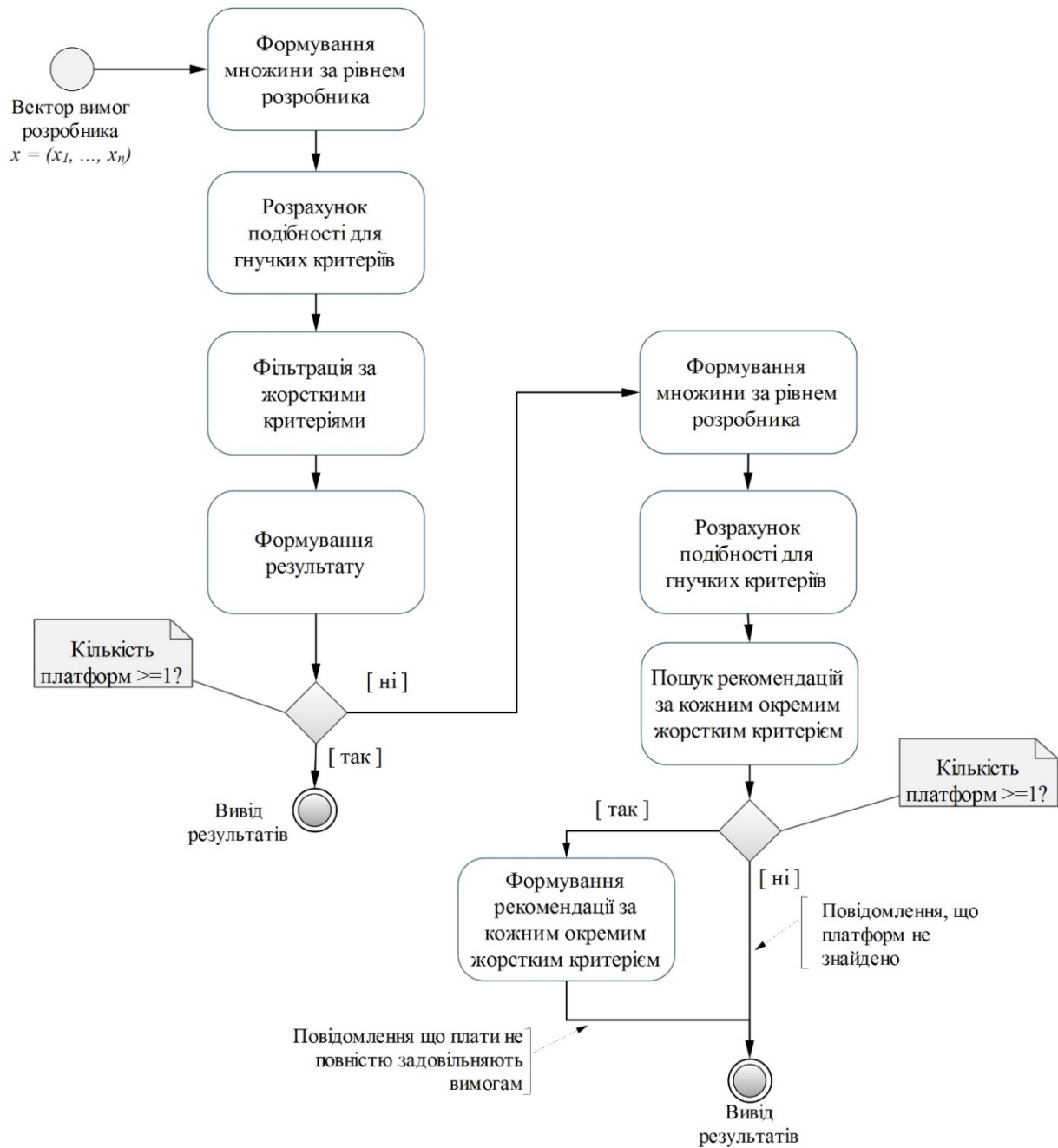


Рис. 3.3. UML діаграма послідовності етапів методу формування рекомендацій щодо апаратно-програмних платформ

враховуючи те, що розробник вводить значення кількості входів, яка йому потрібна, рекомендація меншої кількості не є доцільною, тому, з урахуванням специфіки критеріїв, формула була модифікована умовою:

$$sim(x, y) \geq y \quad (3.8)$$

- якщо введено критерій «живлення», за виразом (3.7) визначаються платформи з множини, визначеної при аналізі цифрових входів, за мірою «ближче – краще»;

- якщо введено критерій «ціна», за виразом (3.6) визначаються платформи з множини, визначеної при аналізі живлення, за мірою «менше – краще».

Визначено, що найкращим варіантом буде призначення ваги  $w = 0,2$  для критерію кількості аналогових, цифрових виходів та живлення, а також ваги  $w = 0,7$  для ціни (рис.3.3). Далі, обчислюється значення подібності апаратно-програмної платформи, з урахуванням ваги кожного гнучкого критерія платформи за формулою (3.4). Платформи сортуються за вагами та повертаються у головний потік алгоритму.

Таблиця 3.1.

## Використовувані метрики за критеріями

Назва критерію	Клас критерію	Метрика	Ваговий коефіцієнт
Рівень користувача	Жорсткий	Жорстке обмеження	1
Кількість аналогових входів	Гнучкий	Ближче-краще, але не менше	0,2
Кількість цифрових входів	Гнучкий	Ближче-краще, але не менше	0,2
Живлення	Гнучкий	Ближче-краще	0,2
Сімейство процесорів	Жорсткий	Жорстке обмеження	1
Мова програмування	Жорсткий	Жорстке обмеження	1
Ціна	Гнучкий	Менше-краще	0,7
Форм-фактор	Жорсткий	Жорстке обмеження	1

Після формування гнучких критеріїв виконується фільтрація за жорсткими критеріями. Платформи, з модифікованої на попередньому етапі множини, сортується за жорсткими критеріями (при їх наявності, інакше алгоритм пропускає етап жорсткого сортування).

Далі виконується формування результуючого списку коефіцієнтів подібності платформ до вимог розробника. Список отриманих рекомендаційних елементів формується за збільшенням показника коефіцієнта подібності.

Після цього, виконується перевірка кількості рекомендованих платформ. Якщо множина платформ, сформована на попередньому етапі, має одну або більше платформ, демонструються чотири найближчі рекомендаційні елементи (у загальному випадку – найкраща за вимогами та три найближчі до неї). Тобто, на виході ми отримуємо до чотирьох апаратно-програмних платформ з мінімальною дистанцією:

$$result = \min d , \quad (3.9)$$

де  $d$  - значення коефіцієнту подібності.

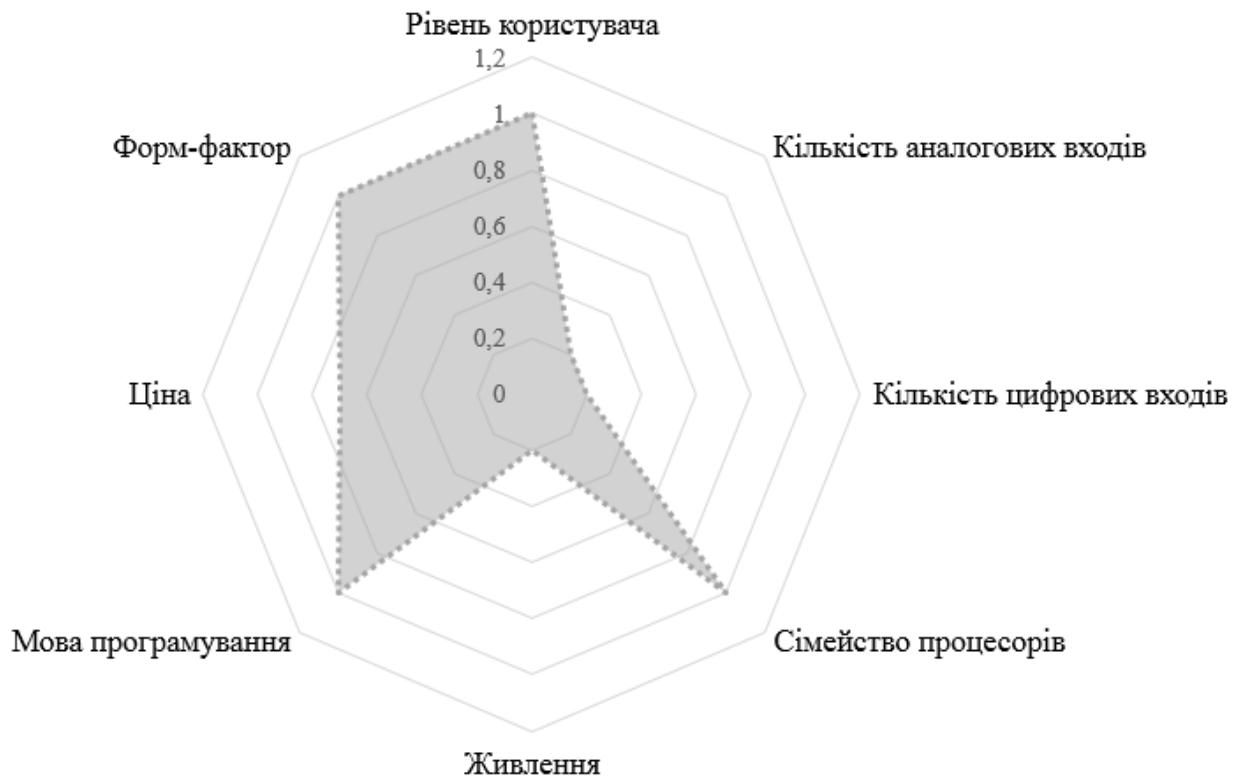


Рис. 3.4. Пелюсткова діаграма співвідношення вагових коефіцієнтів критеріїв



У випадку, коли немає жодної рекомендації, переходимо до формування результату за кожним окремим жорстким критерієм. На цьому етапі для розв'язання суперечливості отриманих вимог використовується метод багатокритеріального вибору найкращої альтернативи за кожним окремим жорстким критерієм. Якщо множина платформ, сформована на попередньому етапі дорівнює нулю, то надається один з можливих наступних варіантів:

- виконуються повторно етапи формування множини за рівнем знань розробника та подібності за гнучкими критеріями, після чого виконується окреме та спільне попарне жорстке сортування за наступними критеріями: сімейство процесорів, мова програмування, форм-фактор платформи; з цих сортувань обирається найкраще (найбільша кількість відповідностей за цими жорсткими критеріями) та виводиться одна платформа по кожному;

- інакше, якщо введені вимоги користувача досить суперечливі, виводиться повідомлення про те, що платформи не знайдено – пропонується повернутися та відредагувати вимоги, введені користувачем.

Таким чином, розроблено метод формування рекомендацій, на основі якого може бути створено інструментарій надання рекомендацій проектувальнику щодо апаратно-програмних платформ при автоматизованому проектуванні ВС.

### **3.5 Висновки до розділу 3**

1. Для організації функціонування рекомендаційної системи обрано рекомендаційний метод на основі знань, оскільки він на відміну від інших не передбачає збір та аналіз поведінки користувачів в минулому, їх оцінок та вподобань щодо об'єктів. Цей метод працює з нагальними вимогами користувача, якими в даному випадку виступають вимоги до проектованої вбудованої системи.

2. Розроблено модель представлення знань у вигляді семантичної

мережі, яка відображає інформацію про апаратно-програмні платформи, дає змогу визначити та структурувати основні поняття (атрибути) бази знань та може бути ефективно використана при розробці бази знань апаратно-програмних платформ рекомендаційної системи.

3. Отримав подальший розвиток метод формування рекомендацій на основі знань, в основу якого покладено міру подібності об'єктів, яка базується на знаходженні близьких об'єктів за числовими значеннями, а також метод обмежень, який знаходить точну відповідність між об'єктами. Крім того, для надання рекомендацій у ситуаціях, коли одна вимога виключає іншу, застосовується метод багатокритеріального аналізу для вибору найкращого рішення за кожною з цих вимог.

Результати розділу опубліковано у наукових роботах [42, 43, 45, 46, 59].

## **РОЗДІЛ 4. РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНО-ТЕХНІЧНОГО КОМПЛЕКСУ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ ВБУДОВАНИХ СИСТЕМ**

В цьому розділі, в рамках рішення завдань дисертаційної роботи, на основі запропонованих методів та моделей розроблено архітектуру програмно-технічного комплексу (ПТК) для автоматизації проектування ВС, програмне та технічне забезпечення віддаленої лабораторії, а також програмне та інформаційне забезпечення рекомендаційної системи. Описано особливості реалізації програмних модулів і бази знань ПТК. Запропоновано методика застосування розробленого ПТК при АП ВС УРО. Наведено приклад практичного застосування ПТК при АП ВС УРО.

### **4.1 Функціональні можливості та архітектура програмно-технічного комплексу**

Базуючись на розробленому в розділах 2, 3 математичному забезпеченні, створено ПТК для автоматизованого проектування ВС на основі технологій віддаленої інженерії. Він призначений для прототипування ВС на основі апаратно-програмних платформ, а також готових проектних рішень з використанням інструментарію рекомендацій та віддалених експериментів. ПТК реалізовано як хмарний сервіс з доступом через мережу Інтернет.

Розроблений програмно-технічний комплекс має забезпечувати наступні основні функціональні можливості при автоматизованому проектуванні ВС:

- надання рекомендацій щодо використання апаратно-програмних платформ, базуючись на введених вимогах до проектованої ВС;
- створення та налагодження керуючої програми для прототипу ВС;

- створення та дослідження прототипу ВС на основі віддалених експериментів, які знаходяться на експериментальному стенді та представляють собою повторно використовувані проектні рішення.

Узагальнена архітектура ПТК наведена на рис. 4.1 та включає в себе рекомендаційну систему, БЗ апаратно-програмних платформ та віддалену лабораторію RELDES (Remote Laboratory for Design of Embedded Systems). РС взаємодіє з БЗ апаратно-програмних платформ через запити, сформовані на основі вимог до ВС, введених користувачем. Віддалена лабораторія через послідовний порт взаємодіє з експериментальним стендом, на якому встановлено набір експериментів (апаратно-програмні платформи з підключеними сенсорами та актуаторами). Віддалена лабораторія також надає доступ користувачам до опису готових проектних рішень у вигляді схем підключення компонентів та текстів програм.

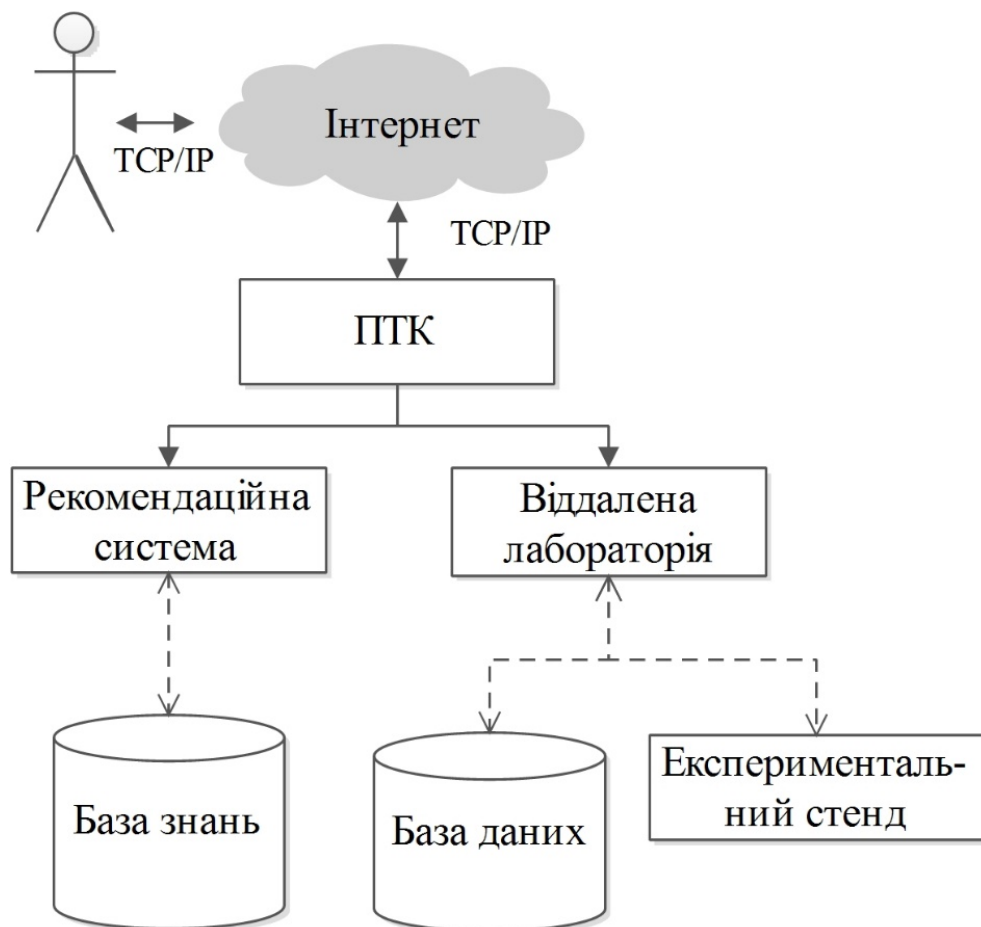


Рис. 4.1. Узагальнена архітектура ПТК

Структурну схему ПТК [61] показано на рис. 4.2. Основними складовими є наступні блоки та модулі.

Блок 1. Робота з базою даних - виконує запити до бази даних та необхідні дії з цими даними, такі як: додавання користувача до системи, отримання інформації про користувача згідно з його ідентифікаційним номером (id), отримання id користувача згідно з його іменем, резервування експерименту користувачем, перевірка стану експерименту (вільний/зайнятий), коректне завершення експерименту.

Блок 2. Логіка роботи програми - реалізує інтерфейс програми та роботу з даними та запитам користувача, відповідає за зміну мови контенту для користувача, авторизацію та реєстрацію користувача в системі, а також вихід користувача з системи.

Модуль 2.1. Формування та робота з описовими даними - відповідає за реалізацію головної сторінки, сторінки з інформацією про апаратно-програмну платформу; виконує переходи між сторінками та надає необхідну інформацію за запитом користувача.

Модуль 2.2. Формування і робота з експериментальною частиною - відповідає за всю роботу віддалених експериментів а саме: реалізацію сторінки з експериментами, стрічкового меню для швидкого переходу між експериментами, виконання експерименту та резервування експерименту користувачем.

Модуль 2.3. Формування і робота рекомендаційної системи – відповідає за роботу рекомендаційної системи, а саме: реалізацію сторінки рекомендаційної системи та роботу з БЗ апаратно-програмних платформ згідно з методом, наведеним у підрозділі 3.3.

Блок 3. Відображення даних для взаємодії з користувачем - формує html-сторінки з інформацією, необхідною для діалогу системи з користувачем, або відображає результати запитів до бази даних.

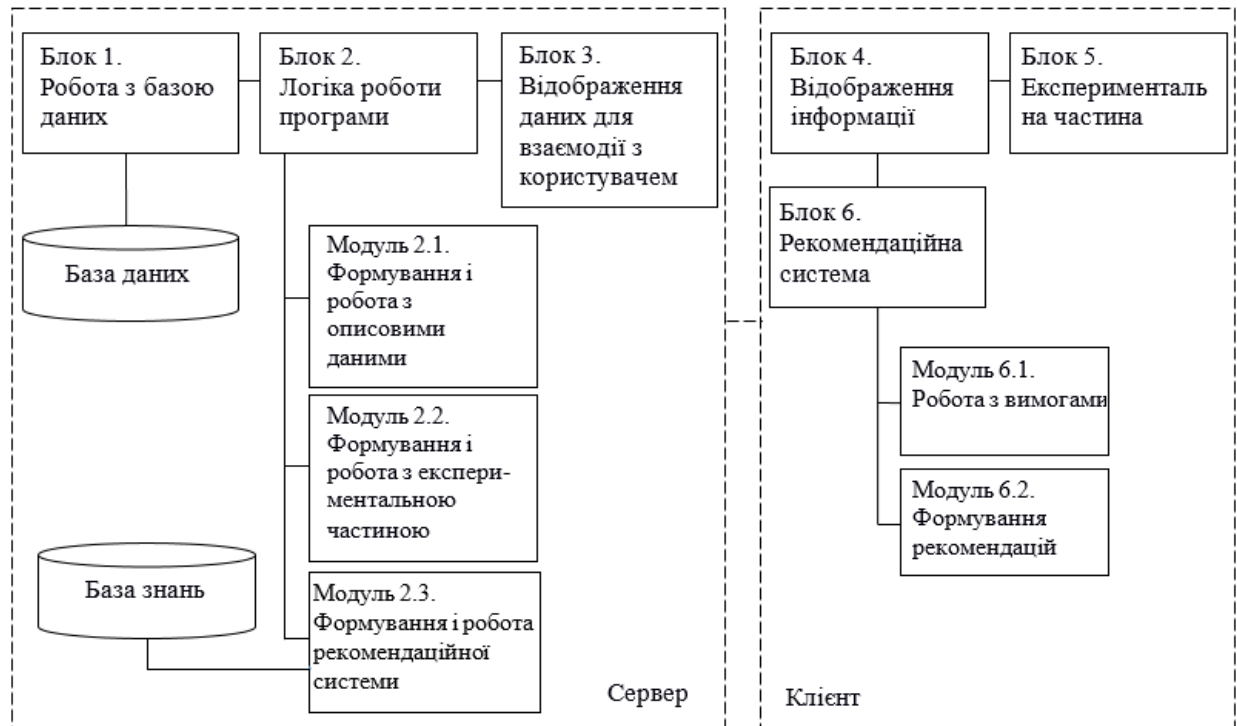


Рис. 4.2. Структурна схема ПТК

Блок 4. Відображення інформації - відповідає за відображення інтерфейсу систем для користувача. Для Блоку 5 - це формування області для відображення відео на сторінці експерименту, а також для розробки та налагодження програмного коду користувачем для керування експериментом. Для Блоку 6 - це відображення рекомендацій щодо апаратно-програмної платформи та доступ до перегляду додаткової інформації (посилання на сайт розробника, технічні параметри та характеристики платформи).

Блок 5. Експериментальна частина – відповідає за проведення експериментів з апаратно-програмними платформами,

Блок 6. Рекомендаційна система – виконує формування рекомендацій щодо апаратно-програмних платформ в залежності від вимог до ВС, введених користувачем.

Модуль 6.1. Робота з вимогами – формує поле для введення вимог користувачем та передає модулю 6.2 сформований вектор вимог.

Модуль 6.2. Формування рекомендацій – відповідає за обробку отриманих вимог для пошуку рекомендацій щодо апаратно-програмних платформ та генерацію результатів пошуку рекомендацій.

Таким чином, запропоновано архітектуру ПТК та структурну схему, що дозволяють реалізувати необхідні функціональні можливості ПТК та організувати ефективне використання засобів віддаленої інженерії при АП ВС.

## **4.2 Опис особливостей розробки програмно-технічного комплексу**

### **4.2.1 Основні рішення щодо реалізації віддаленої лабораторії**

Архітектура віддаленої лабораторії RELDES [51, 52] наведена на рис.4.3 та організована згідно з принципами архітектури клієнт-сервер.

Комунікація між клієнтом і сервером віддаленої лабораторії здійснюється через веб-інтерфейс. Стенд з експериментами підключений до лабораторного сервера через послідовний інтерфейс. Комп'ютер з операційною системою Linux Debian працює як сервер віддаленої лабораторії. Сервер забезпечує доступ до програмування експериментів та трансляції відео лабораторії. Сервер виконує обробку запитів веб-клієнта та виконує наступні дії в залежності від отриманих даних:

- відправка запиту на компіляцію, а також повернення результатів компіляції програми (за допомогою HTTP-протоколу);
- компіляція отриманого початкового коду на стороні сервера [51];
- завантаження бінарного файлу до мікроконтролеру (за умови, що контролер не зайнятий);
- побудова черги користувачів (якщо експеримент зайнятий).

Служба працює на сервері Apache з використанням реляційної бази даних MySQL [60]. Веб-сервер реалізовано на мові програмування PHP5. Такий стек технологій дозволяє реалізувати всі необхідні функції. Для побудови необхідного функціоналу використовується фреймворк CodeIgniter

та шаблон MVC (Model–View–Controller, Модель–Представлення–Контролер). Це популярний фреймворк з відкритим кодом, написаний на мові програмування PHP для розробки повноцінних веб-систем і з'єднань. Розроблена на основі MVC шаблону структура віддаленої лабораторії представлена на рис. 4.4 [58].

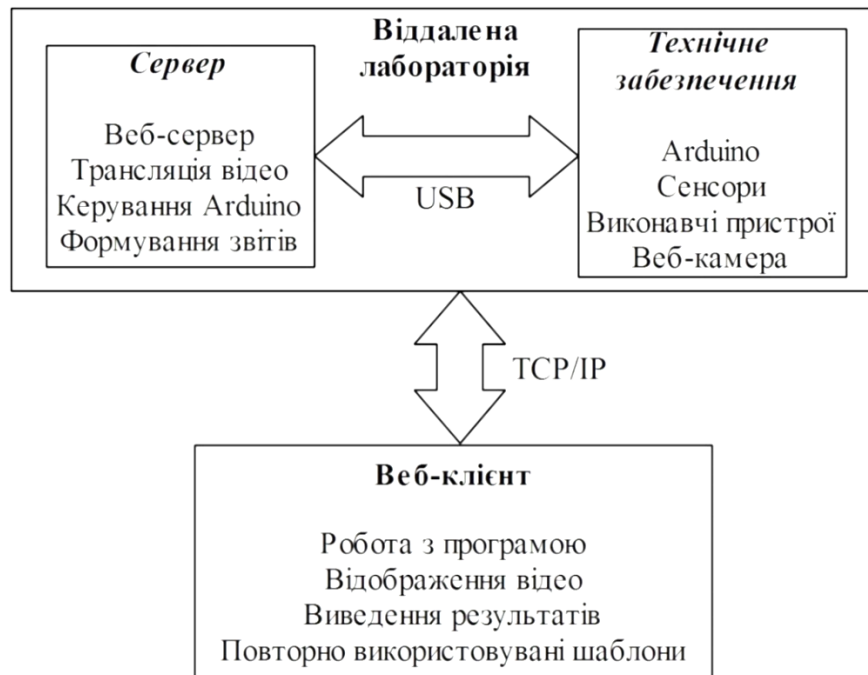


Рис.4.3. Архітектура віддаленої лабораторії

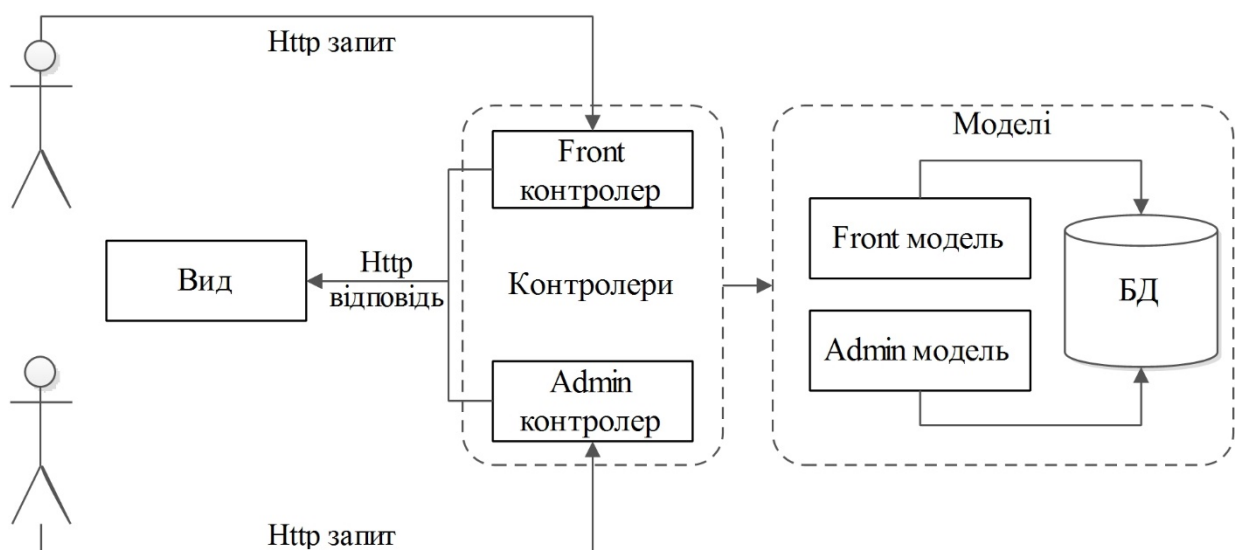


Рис.4.4. Структура віддаленої лабораторії на основі шаблону MVC



Всі запити від користувачів обробляються відповідними Контролерами (рис. 4.5, а). Таким чином, Контролер забезпечує зв'язок між користувачем і системою, а саме контролює введення даних користувачем і використовує Модель і Представлення для реалізації необхідної реакції. Крім того, Контролер реалізує функції навігації сайту (наприклад, функції: `index`, `arduino`, `experiments` та т.і.).

Окремим модулем реалізовано трансляцію відеозображення. Вона реалізована як MPEG-трансляція через веб-сокети з відмальовкою в тег `<canvas>`. Відео з камери кодується за допомогою `ffmpeg`. JavaScript код, який відпрацьовує на `nodejs`, відправляє кадри через веб-сокети. При отриманні користувачем кадрів, вони декодуються в веб-клієнті за допомогою `jsmpeg` (MPEG1 відео-декодер). Після декодування кадр промальовується в тег `<canvas>` за допомогою `WebGL`.

Модель надає дані та методи роботи з цими даними, реагує на запити, змінюючи свій стан. Однак не містить інформації про те, як ці знання можна візуалізувати. Зокрема, Модель займається реєстрацією та авторизацією користувачів. Також Модель надає інформацію про експерименти: зайнятий чи вільний.

Представлення відповідає за відображення (візуалізацію) інформації (рис.4.5, б). Найчастіше в якості Представлення виступає форма (вікно) з графічними елементами. Кожне Представлення має свій вигляд:

- `index` – головна сторінка;
- `header` – стрічкове меню головної сторінки;
- `footer` – нижня інформаційна частина головної сторінки;
- `arduino` – опис платформ `Arduino`;
- `experiments` – загальна частина всіх експериментів;
- `LED` – специфічна частина для експериментів зі світлодіодами;
- `servo` – специфічна частина для експериментів з серводвигуном;
- `distance` – специфічна частина для експериментів з сенсором відстані;
- `display` – специфічна частина для експерименту з дисплеєм.



Рис. 4.5. Контролер (а) та Представлення (б)

Веб-інтерфейс (рис.4.6) реалізовано з використанням мови програмування Java Script, мови розмітки HTML та каскадних таблиць стилів CSS. Сторінку експерименту розділено на декілька блоків. У лівій частині сторінки експерименту формується вертикальне меню, що дозволяє швидко перемикатися між експериментами. Основна область сторінки ділиться на відображення відео (праворуч) та редактор коду (у центрі). У нижній частині сторінки є область повідомлень, в якій відображаються результат компіляції програми, а також наведено приклади коду та схеми підключення експериментів. Система має зручний, інтуїтивно зрозумілий інтерфейс, що дозволяє швидко виконати потрібні дії в обраному експерименті, створити програмний код та спостерігати фізичну реалізацію експерименту.

Технічне забезпечення лабораторії включає експериментальне обладнання, що може розглядатись як сукупність повторно використовуваних апаратних рішень для розробки ВС. Набір експериментів може бути створений з урахуванням різних типів ВС. Для ВС УРО - це робота з серводвигуном, сенсором відстані, світлодіодами, дисплеєм та ін.. Для застосування при АП ВС УРО повторно використовуваних рішень надаються схеми підключення та шаблони програм (рис. 4.7.).

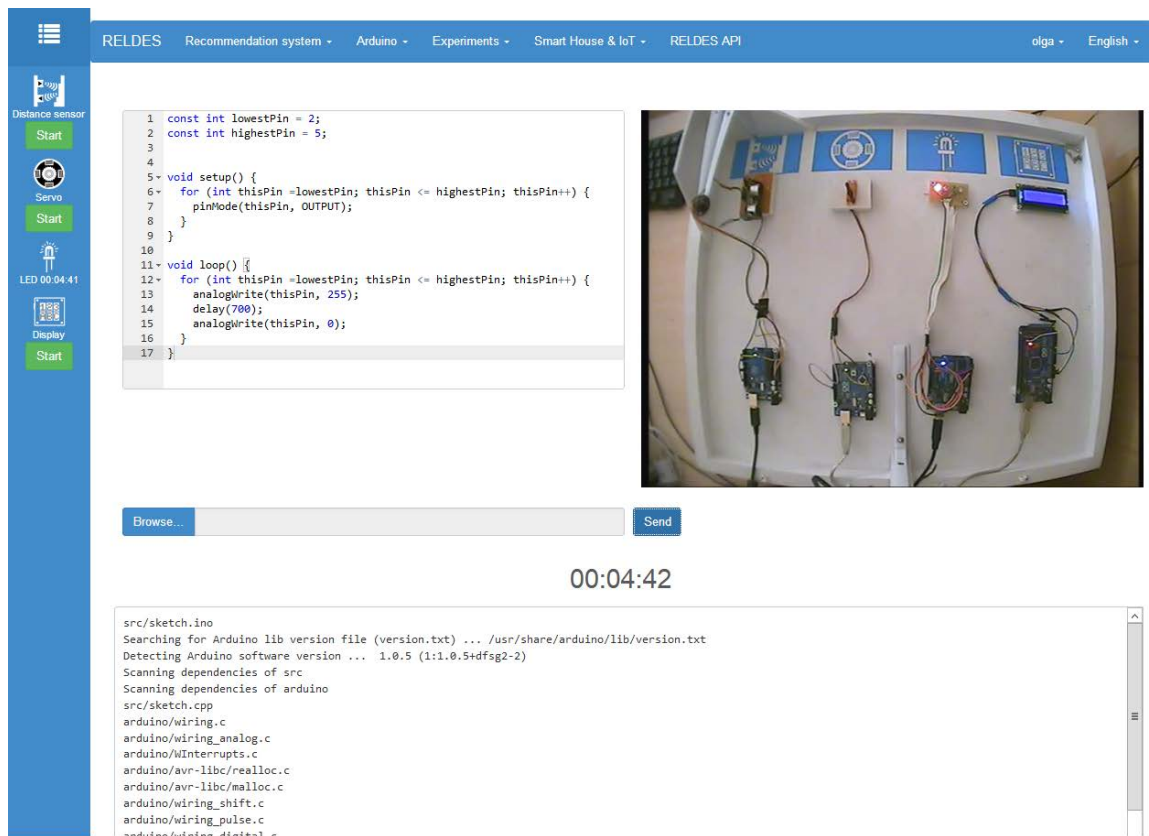


Рис. 4.6. Веб-інтерфейс віддаленої лабораторії

На даний момент усі експерименти побудовані довкола використання апаратно-програмних платформ Arduino, але система є відкритою для розширення та додавання інших платформ. Для уникнення проблеми ініціалізації платформ був створений конфігураційний файл, який привласнює унікальне ім'я платформі з ідентифікатором виробника, крім того, кожній платформі було надане ім'я ArduinoN (де N є номером експерименту), таке рішення полегшує подальше використання системи при збільшенні кількості платформ.

З метою організації доступу до експериментів була створена підсистема організації черги. Також є можливість обмежити тривалість експерименту. Черга працює за принципом FIFO (First in, First Out), тобто користувач, який натиснув на кнопку «Виконати експеримент» стає в чергу, а якщо вона порожня, одразу починає виконання експерименту, інакше, запускається таймер відліку часу перебування у черзі, та по закінченню певного часу, користувач може розпочати експеримент.

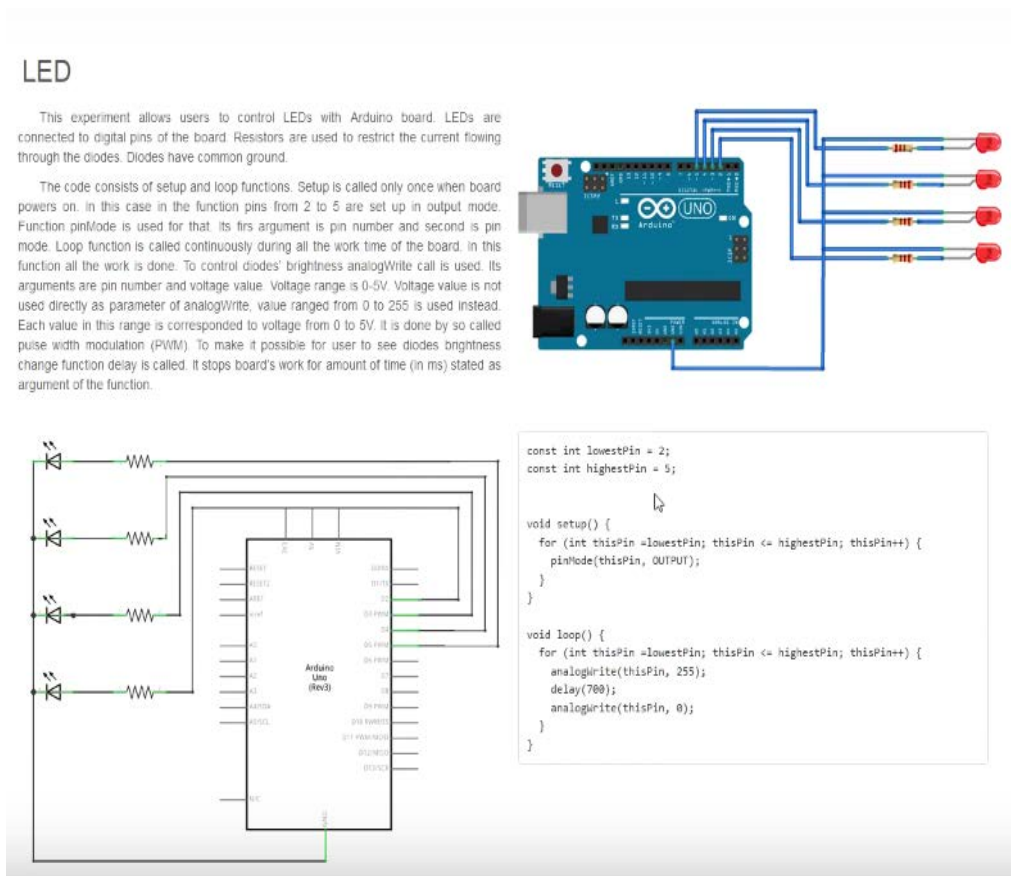


Рис. 4.7. Приклад опису повторно використовуваних апаратних та програмних рішень на веб-сторінці віддаленої лабораторії

На рис. 4.8 зображено діаграму послідовності, яка демонструє послідовність подій, що відбуваються під час роботи підсистеми організації черги. Коли експеримент зайнятий, то створюється ще один таймер, який спрацює по закінченню виділеного часу на експеримент для звільнення експерименту від поточного користувача та початку роботи з експериментом наступного користувача в черзі.

Розроблене програмне та технічне забезпечення віддаленої лабораторії надає користувачам можливість спільного використання обладнання та програмного забезпечення для створення та дослідження прототипу проектованої ВС. Таким чином можуть бути зменшені матеріальні та часові витрати на розробку та, відповідно, собівартість проектованої продукції.

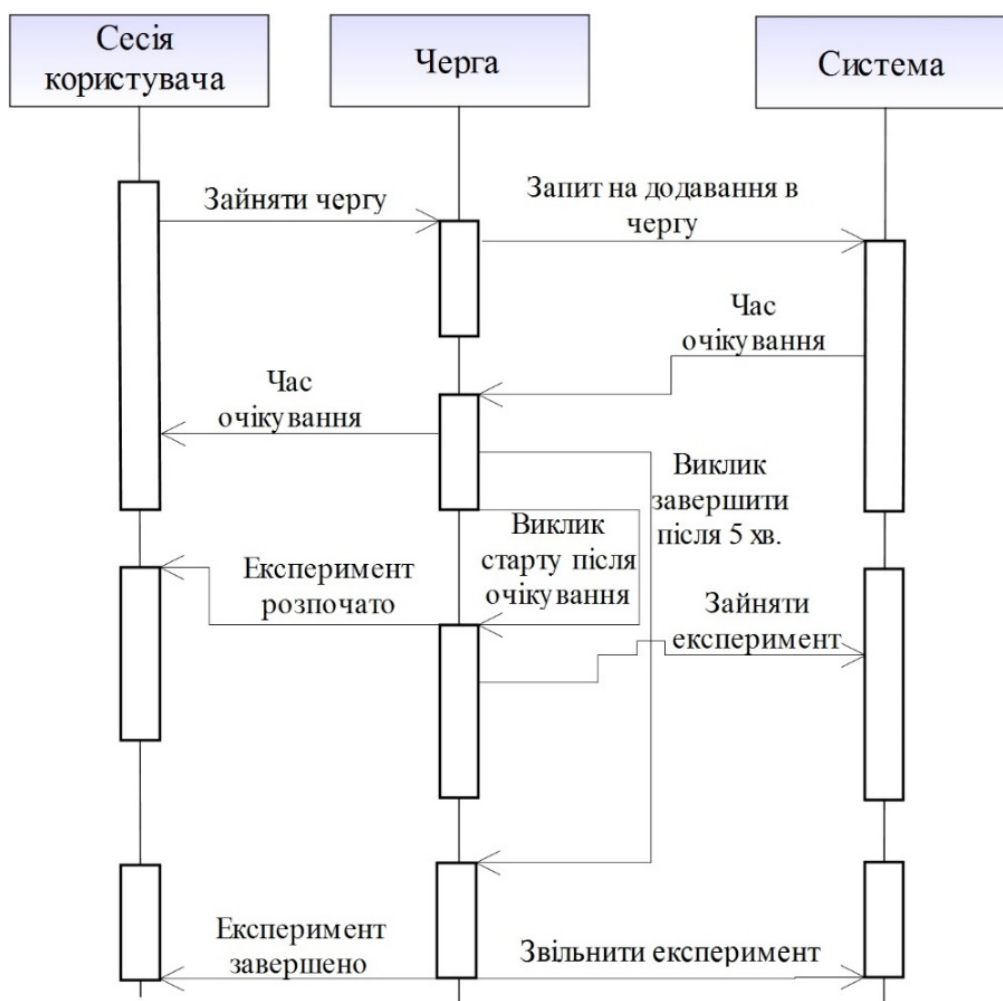


Рис. 4.8. Діаграма послідовності роботи підсистеми організації черги

#### 4.2.2 Основні рішення щодо реалізації рекомендаційної системи

Розроблену архітектуру РС наведено на рис. 4.9. Проектувальник взаємодіє з РС через веб-інтерфейс. У загальному вигляді РС складається з декількох модулів. Модуль формування веб-сторінки та Модуль пошуку рекомендацій безпосередньо взаємодіють з веб-інтерфейсом та базою знань. Модуль формування веб-сторінки, окрім формування головної веб-сторінки з апаратно-програмними платформами, також формує додаткові сторінки з розширеною інформацією про апаратно-програмні платформи. Формування рекомендацій щодо апаратно-програмних платформ виконує Модуль пошуку рекомендацій.

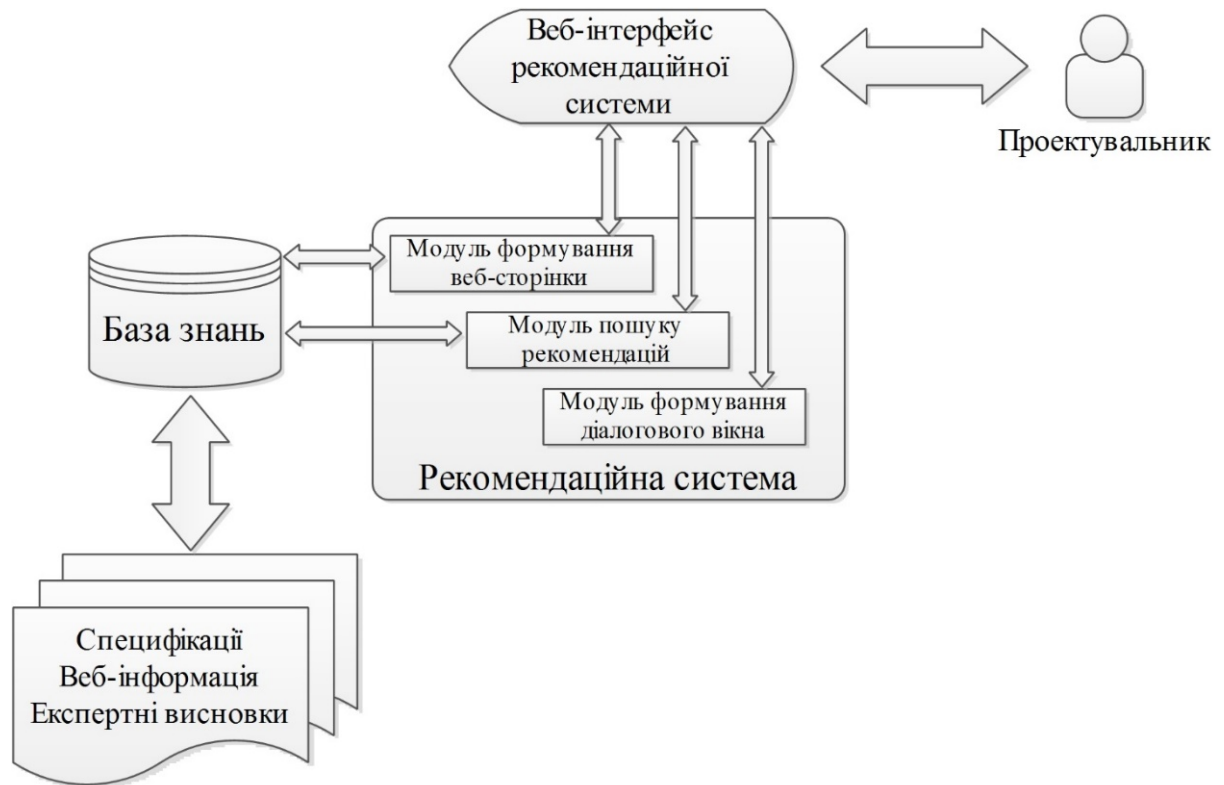


Рис. 4.9. Архітектура рекомендаційної системи

Модуль формування діалогового вікна створює статичну область для роботи з вимогами користувача. База знань формується на основі специфікацій апаратно-програмних платформ, інформації з веб-сайтів розробників платформ, а також експертних висновків.

Вхідними даними для роботи РС є: рівень розробника ВС; планована кількість контактів для підключення периферії; переваги щодо використання окремих сімейств процесорів; знання мови програмування; обмеження цінового діапазону та ін..

Результатами роботи РС є перелік рекомендованих апаратно-програмних платформ. РС обирає найближче значення критеріїв платформ з бази знань згідно вимог, введених користувачем в системі.

Кожен критерій бази знань апаратно-програмних платформ був сформований на основі проведеного аналізу специфікацій цих платформ та на результатах опитування експертів.

Рівень користувача розраховується виходячи з суми двох показників: наявності спільноти в мережі та порогу входження до розробки.

Поріг входження визначається значеннями «0» – для початківців у сфері використання платформ, «1» – для досвідчених розробників, що мають середній рівень навичок роботи з платформами, «2» – для професіональних розробників. Відкритість описує ступінь наявності документації, порадників, інструкцій та готових бібліотек значеннями від «0» (тобто системи з достатнім рівнем документації) до «1» (повна відсутність чи проблеми доступу до описової інформації).

Введення атрибуту «Сімейство процесорів» призначене полегшити пошук платформи за рахунок зменшення різновидів керуючих процесорів шляхом їх розподілу за сімействами, до яких вони відносяться. В результаті пошуку за сімейством процесорів, будуть представлені ті платформи, процесори яких належать до вказаного сімейства.

Мова програмування обирається зі списку з розрахунку на те, що користувач вказує ту мову програмування, якою володіє краще за інші та з використанням якої бажає працювати з платформою.

Ціна платформи впливає на фінансові показники проекту та на даному етапі розробки визначається потраплянням до інтервалу фактичних цін платформ, які вказано у базі даних рекомендаційної системи. В базі даних ціна платформи вказана в гривнях.

Форм-фактор, з метою уникнення надлишковості даних, було визначено для декількох категорій: малі платформи, середні та великі, що визначається площею платформи за характеристиками її форми. Під час аналізу бази платформ, було вирішено поділити множину платформ за такими значеннями: до 30 см<sup>2</sup>, від 30 см<sup>2</sup> до 60 см<sup>2</sup>, а також більше 60 см<sup>2</sup>, відповідно. При цьому, користувач зможе отримати список тих платформ, що потрапляють у зазначені вище інтервали.

В табл. 4.1 наведено початкові дані, які використано при розробці бази знань з урахуванням можливості її подальшого розширення та модифікування.

Таблиця 4.1.

## Систематизовані знання про предметну область

Назва	Сімейство процесорів	Процесор	Форм-фактор	Ціна	Мова розробки	Живлення	Цифрові входи	Аналог. входи	Відкритість	Поріг виходження
Arduino Uno	AVR	ATmega328	6,8x5,3	204-630	подоба C++	5	14	6	0	0
Arduino Leonardo	AVR	ATmega32u4	6,8x5,3	186-632	подоба C++	5	20	12	0	0
Arduino Mega	AVR	ATmega2560 – 16AU	10,2x5,3	311-1355	подоба C++	5	54	16	0	0
Arduino Due	ARM	ATSAM3X8E Cortex-M3	10,2x5,3	577-1166	подоба C++	3,3	54	16	0	1
Arduino Mini	AVR	ATmega328P	3,3x1,8	69-220	подоба C++	5	14	8	0	0
Arduino Micro	AVR	ATmega32U4	3,3x1,8	143-594	подоба C++	5	12	4	0	0
Arduino Yun	AVR	ATmega32U4	6,8x5,3	1830	подоба C++	5	14	6	0	1
MSP430 LaunchPad	MSP430	MSP-EXP430G2	6,8x5,3	547	подоба C	5	16	8	0	0
Iskra Neo	AVR	ATmega32U4	6,8x5,3	600	подоба C	5	20	12	1	0
Iskra Mini	AVR	ATmega328	3,3x2	200-300	—	5	14	8	1	0
Iskra JS	ARM	ARM Cortex-M4	6,8x5,3	940	JavaScript	3,3	26	12	1	1
Strela	AVR	ATmega32U4	10,5x7,9	1400	подоба C++	5	11	8	1	0
Romeo BLE	AVR	ATmega328P	9,4x8	1146	подоба C++	5	14	8	0,5	0
STM32 Nucleo L4	STM	STM32L432KCU6	3,6x2	486	подоба C	3,3	13	8	0,3	1
STM32 Nucleo	STM	STM32F411	10x7,5	694	подоба C	3,3-5	50	16	0,3	1
STM32 Nucleo-144	STM	STM32F746ZGT6	14x7,5	939	подоба C	3,3-5	160	24	0,3	1
STM32 Nucleo-32	STM	STM32F303K8T6	3,6x2	544	—	3,3	14	10	0,3	1
STM32 Nucleo-64	STM	STM32F446RET6	10x7,5	544	—	3,3-5	50	16	0,3	1
Teensy 3.2	ARM	32bit ARM Cortex-M4	3,6x1,8	696	C	3,3	26	21	0,5	2
Netduino 2	STM	STM32F2 Cortex-M3	6,8x5,3	959	C#, .Net	3,3-5	14	6	0,3	2
Netduino Plus 2	ARM	32bit ARM Cortex-M4	6,8x5,3	1436	C#, .Net	3,3-5	14	6	0,3	2
Raspberry Pi 3 Model B	ARM	Broadcom BCM2837 64bit ARM Cortex-A53	8,5x5,4	1378	Python, Java, C/C++, Bash	5	27	0	0	2
BeagleBone Black	Sitara	Sitara AM3359	8,7x5,5	1960	Python, Ruby, JavaScript	5	65	7	0,5	2
pcDuino	AllWinner	AllWinner A20	12,1x6,5	1515	Python, Java, C/C++	5	16	6	0,5	2
Intel Galileo	Intel Quark	Quark SoC X1000	1,5x10,7	2172	Python, Java, C/C++	5	20	6	0,5	2



Для формування БЗ використано словник предметної області, який був сформований при побудові семантичної мережі та включає в себе поняття (слова), використовувані у рекомендаційній системі. У процесі розробки БЗ (табл. 4.2) було створено декілька сутностей, що містять інформацію про платформи (табл. 4.3– 4.6). Схематичне зображення сутностей та зв'язків між ними зображено у вигляді логічних схем на рис. 4.10 та 4.11.

Таблиця 4.2.

## Загальний опис сутностей бази знань рекомендаційної системи

Назва сутності	Назва	Опис
Board	Платформи	Головна сутність, що містить повний опис платформи
Processor	Процесор	Сутність, що містить інформацію про процесори платформ
FamilyProcessor	Сімейство процесорів	Довідкова сутність, що містить інформацію про сімейство процесорів
Category	Категорія	Сутність, що містить інформацію про категорії, на які розподіляється множина платформ
Languages	Мови програмування	Сутність, що містить інформацію про мови програмування, які використовуються при роботі з платформами системи

Таблиця 4.3

## Опис сутності «Платформи» («Board»)

Назва поля	Тип значення	Опис
ID	Integer	Ідентифікатор платформи
Name	Varchar(48)	Ім'я платформи
Slug	Varchar(250)	Технічне поле для роботи з базою даних та для формування URL («ім'я_платформи_процесор»)
Processor_ID	Integer	Зовнішній ключ, що вказує на процесор платформи
Community_openness	Decimal	Поле, що містить значення відкритості суспільства
Entry_threshold	Integer	Поле, що містить значення порогу входження
Power	Decimal	Поле, що містить значення живлення платформи
Analog_port	Integer	Кількість аналогових входів

Назва поля	Тип значення	Опис
Digital_port	Integer	Кількість цифрових входів
Width	Decimal	Значення форм-фактору (Width x Length)
Length	Decimal	Значення форм-фактору (Width x Length)
Min_Price	Decimal	Мінімальна ціна платформи
Max_Price	Decimal	Максимальна ціна платформи
Image	Varchar(100)	Поле, що містить адресу зображення платформи
Category_ID	Integer	Зовнішній ключ, що вказує на категорію платформи

Таблиця 4.4

Опис сутності «Процесор» («Processor») та сутності «Сімейство процесорів» («FamilyProcessor»)

Назва поля	Тип значення	Опис
«Процесор»		
ID	Integer	Ідентифікатор процесору
Name	Varchar(48)	Ім'я процесору
Family_ID	Integer	Зовнішній ключ, що вказує на сімейство процесорів
«Сімейство процесорів»		
ID	Integer	Ідентифікатор сімейства процесорів
Name	Varchar(48)	Ім'я сімейства процесорів

Таблиця 4.5

Опис сутності «Категорія» («Category»)

Назва поля	Тип значення	Опис
ID	Integer	Ідентифікатор процесору
Slug	Varchar(48)	Ім'я категорії (технічне поле для URL)
Name	Varchar(48)	Ім'я категорії англійською, українською та російською

Таблиця 4.6

Опис сутності «Мови програмування» («Languages»)

Назва поля	Тип значення	Опис
ID	Integer	Ідентифікатор процесору
Name	Varchar(48)	Ім'я мови програмування

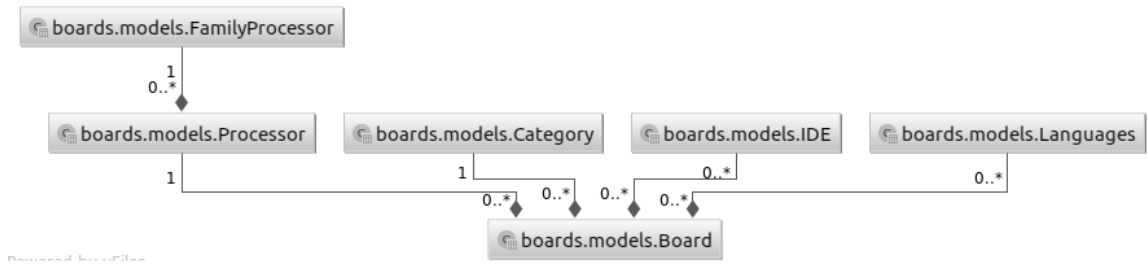


Рис. 4.10. Логічна схема зв'язків між сутностями

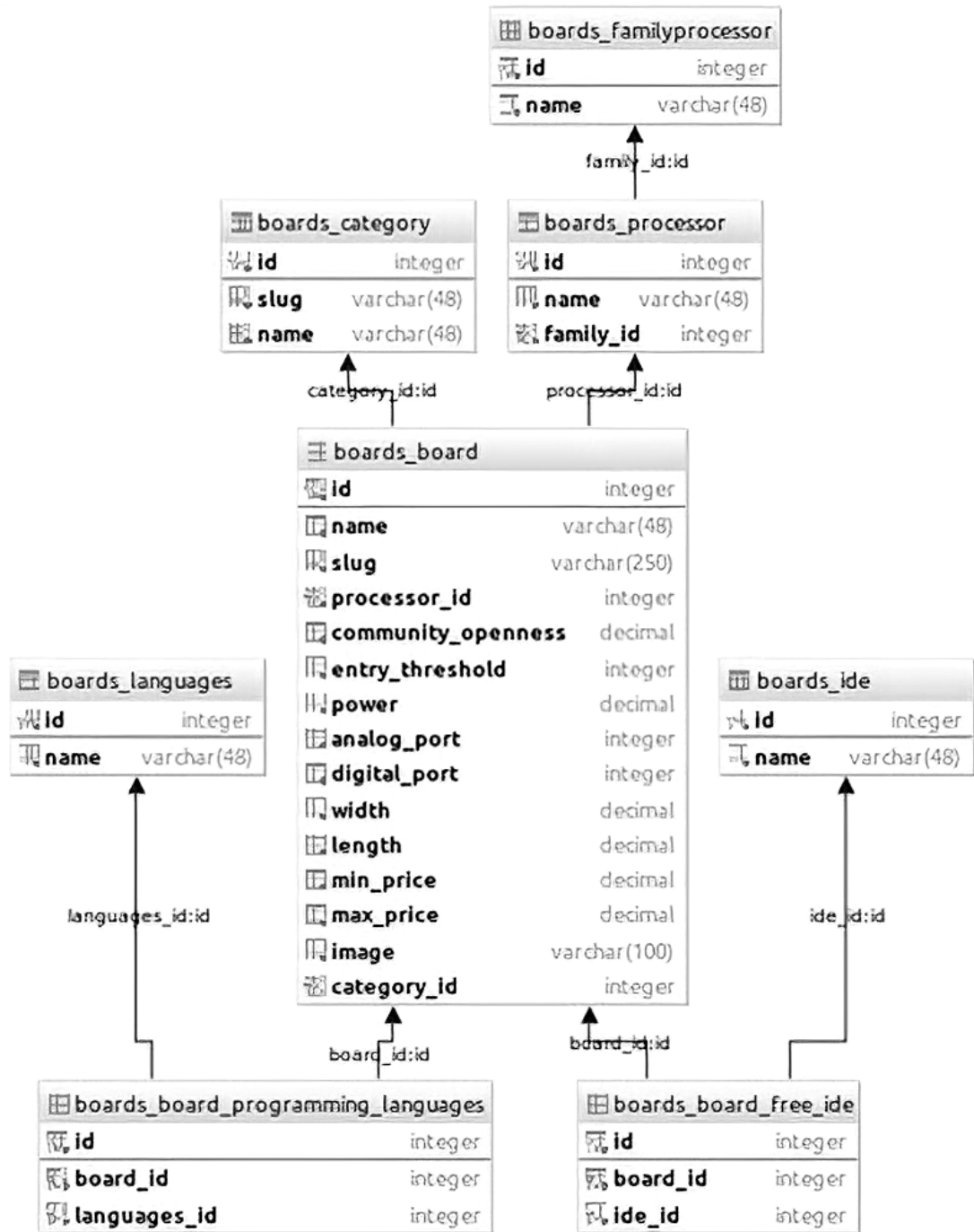


Рис. 4.11. Логічна схема сутностей та зв'язків

РС розроблена з використанням мови програмування Python та фреймворку Django для прискорення процесу розробки. РС, як і віддалена лабораторія реалізується на основі архітектурного шаблону MVC.

Інтерфейс сторінки формування рекомендацій щодо апаратно-програмних платформ зображено на рис. 4.12. Екранна форма поділена на дві частини – формування рекомендованих платформ з бази знань (праворуч) та видобуток вимог користувача (ліворуч). Користувач має можливість переглянути інформацію, що міститься в БЗ, задаючи різні критерії пошуку.

Для формування рекомендацій щодо апаратно-програмної платформи користувачу пропонується ввести набір запитів в області вводу вимог та натиснути кнопку «Відправити». Після цього розпочнеться робота алгоритму формування рекомендації щодо апаратно-програмних платформ. В процесі пошуку, дані щодо апаратно-програмних платформ беруться з БЗ та за допомогою наданого функціоналу Django передаються html-сторінці, на якій виводяться у вигляді карток-рекомендацій. Для запропонованого варіанту платформи надається можливість перейти на сайт компанії розробника для ознайомлення з більш детальною інформацією щодо параметрів та характеристик платформи, а також на головну сторінку лабораторії RELDES для віддалених експериментів з платформою.

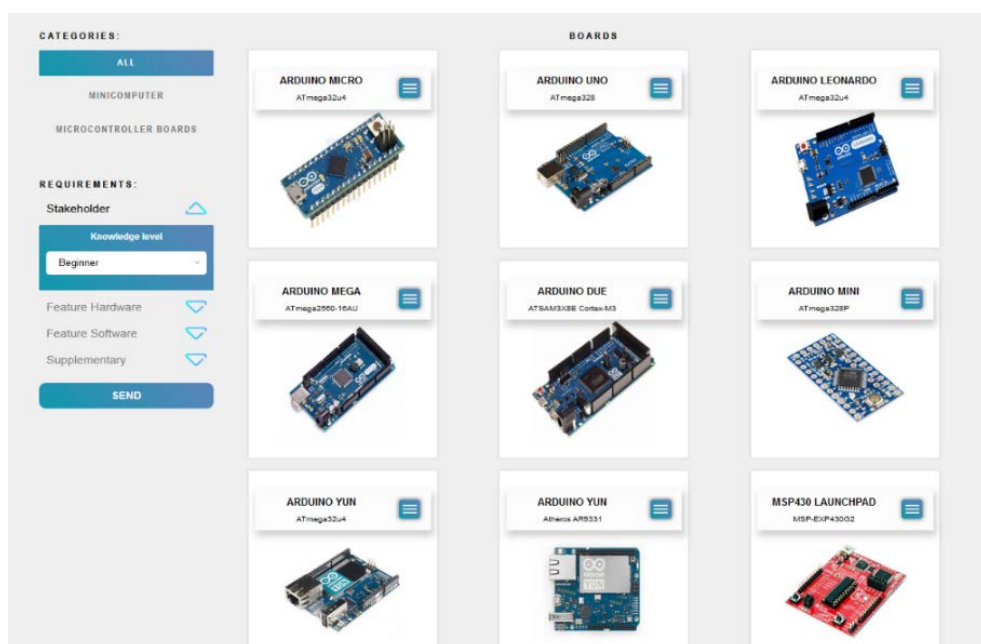


Рис. 4.12. Інтерфейс рекомендаційної системи

Таким чином, розроблено програмне та інформаційне забезпечення рекомендаційної системи, яка надає підтримку проектувальнику у вигляді рекомендації щодо використання апаратно-програмних платформ при автоматизованому проектуванні ВС, виходячи з вимог до проектованої вбудованої системи.

### **4.3 Практичне застосування розробленого комплексу програмних та технічних засобів для автоматизованого проектування вбудованої системи управління рухомими об'єктами**

Створений ПТК застосовувався при розробці ВС УРО, особливістю якої є можливість управління групою рухомих об'єктів, контролю їх параметрів (швидкість, заряд батареї живлення та ін.) та створення ефекту присутності операторів в досліджуваній зоні. Для ефективного впровадження ПТК в процес автоматизованого проектування ВС розроблено методику, яка охоплює наступні проектні етапи: розробка специфікації; розробка архітектури системи; вибір апаратно-програмних платформ на основі рекомендаційної системи; розробка програмного забезпечення; створення та дослідження прототипу системи на основі віддаленої лабораторії.

#### **4.3.1 Методика застосування програмно-технічного комплексу при автоматизованому проектуванні вбудованих систем управління рухомими об'єктами**

Методика застосування ПТК може бути представлена у вигляді діаграми діяльності [50], що наведена на рис. 4.13. Щоб отримати доступ до роботи з комплексом, користувач має бути зареєстрованим в системі. Для

реєстрації та авторизації користувача у базі даних зберігається логін і пароль користувача.

Для вибору апаратно-програмної платформи проектувальник має ввести вимоги в запропонованій формі на сторінці РС (рис.4.12):

- рівень знань розробника ВС (початківець, професіонал);
- планована кількість контактів для підключення периферії (аналогових та цифрових);
- переваги щодо використання окремих сімейств процесорів;
- напруга живлення;
- знання мови програмування;
- обмеження цінового діапазону;
- форм фактор (малий, середній, великий) та ін.

Відповідь на всі запити не є обов'язковою, але дозволяє більш точно сформулювати рекомендації щодо платформи.

Після відправки набору запитів, РС обробляє їх та формує рекомендації. У вікні виводу результату користувач спостерігає картки-рекомендації. Їх може бути одна, або декілька, в залежності від повноти вхідних даних. Після аналізу рекомендованих платформ та вибору необхідної, користувач переходить на сторінку віддаленої лабораторії, де йому надається інструкція з проведення експериментів, а також опис експериментів.

Проектувальник може швидко створити прототип проектованої системи на основі пропонованих готових проектних рішень або власних оригінальних. Користувач може завантажити програмний код до мікроконтролеру апаратно-програмної платформи за допомогою веб-інтерфейсу, при цьому він може створити власний код з нуля або завантажити пропонований файл-шаблон з готовим кодом. Таким чином, розробник має можливість створювати та редагувати код програми безпосередньо в веб-інтерфейсі клієнта без необхідності застосування інструментів розробки на його комп'ютері.

Після того, як користувач відправив програму, вона доставляється на сервер по HTTP-протоколу. Отриманий текст програми зберігається в файл та передається на компіляцію. Якщо в програмі є помилки, сервер формує та повертає результати компіляції. Після успішної компіляції згенерований код завантажується до мікроконтролеру апаратно-програмної платформи (наприклад, Arduino). При цьому, може використовуватись, наприклад, утиліта Ino, яка дозволяє працювати з Arduino з командного рядка. Це дає можливість:

- швидко створювати нові проекти;
- компонувати програму з декількох вихідних файлів і бібліотек;
- завантажувати програму на пристрій;
- здійснювати Serial-зв'язок з пристроєм.

Після виконання компіляції і завантаження програми до платформи Arduino, відправляються повідомлення на клієнтську частину. Завдяки Ino можна отримати детальну інформацію про компіляцію та завантаження програми, що дозволяє користувачеві легко знайти помилку. Якщо операції пройшли успішно, перегляд звітів необов'язковий і користувач може спостерігати результат роботи своєї програми у вікні трансляції відеозображення експерименту.

Таким чином, користувач спостерігає реальну роботу обладнання (апаратно-програмна платформа з підключеними сенсорами та актуаторами) та оцінює працездатність прототипу шляхом аналізу робочих процесів. Існує додаткова можливість виводу робочих параметрів та характеристик прототипу у монітор послідовного порту.

Після оцінювання прототипу проектованої системи, проектувальник може перейти до етапів власне розробки системи та тестування.

Таким чином, розроблено методику застосування створеного програмно-технічного комплексу при автоматизованому проектуванні вбудованих систем управління рухомими об'єктами, яка може використовуватись в практиці інженерного проектування для ефективного впровадження засобів віддаленої інженерії.

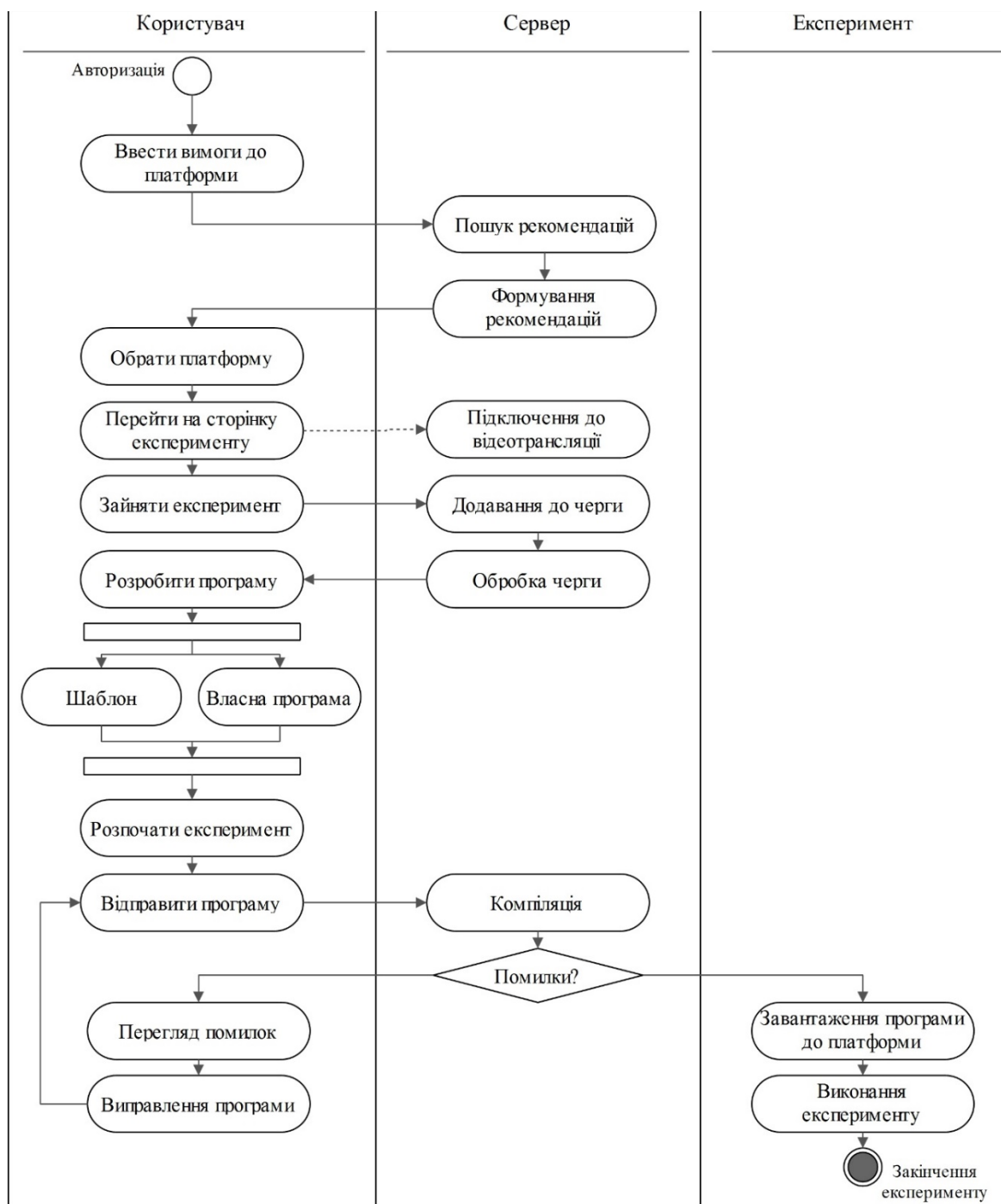


Рис. 4.13. Діаграма діяльності користувача з ПТК

### 4.3.2 Розробка специфікації вимог

У відповідності з удосконаленим методом проектування ВС, запропонованим у підрозділі 2.3, на першому етапі було виконано розробку специфікації вимог до проектованої ВС УРО [46].



Описані нижче приклади являють собою взаємопов'язану послідовність вимог, що стосуються окремої частини даної проекрованої системи, оскільки повний опис є достатньо об'ємним та наведений у проектній документації.

Приклад 1: Вимоги користувача.

STRQ7: Зміна максимально допустимої швидкості рухомого об'єкта в залежності від категорії користувача (початківець, професіонал), а також рівня навичок управління.

Приклад 2: Варіант використання.

UC3: Перемикання швидкості рухомого об'єкта.

Дійові особи: Оператор.

Вихідні умови: оператор увімкнув Центральний блок керування (ЦБК) і виконав попередні налаштування заїзду рухомого об'єкта.

Нормальний напрям:

- оператор перемикає швидкісний режим рухомого об'єкта;
- система відправляє команду на рухомий об'єкт про перемикання швидкості;
- рухомий об'єкт приймає команду і обробляє її;
- на дисплеї з'являється повідомлення про перемикання швидкості рухомого об'єкта в обраний режим.

Альтернативний напрям: немає.

Винятки: команда не була відправлена - система видає номер помилки на дисплей.

Приклад 3: Системні вимоги.

SYST4: Система повинна перемикати (збільшувати та зменшувати) максимальну швидкість радіокерованого рухомого об'єкта.

При перемиканні швидкостей система повинна виводити повідомлення на дисплей про швидкісний режим, що був обраний.

Приклад 4: Обмеження.

SUPL2: Наявність двох режимів задавання максимальної швидкості

радіокерованого рухомого об'єкта:

- режим Beginner – 6 км/год.
- режим Professional – 20 км/год.

В даному випадку вимога розбивається на функціональні вимоги до апаратного і програмного забезпечення.

Приклад 5: Вимоги до апаратного забезпечення.

FEAT\_HW7: Двигун, встановлений на рухомій платформі, повинен як мінімум забезпечувати швидкість 20 км/год.

Функціональна вимога до ПЗ визначається, виходячи із заданих системних вимог і апаратної складової. Оскільки в нашому проекті на рухомому об'єкті був встановлений двигун, що забезпечує максимальну швидкість 60 км/год., то функціональна вимога до ПЗ буде виглядати, як наведено у наступному прикладі.

Приклад 6: Вимоги до ПЗ.

*Подія:* Встановлення швидкості на ЦБК в режим Beginner.

*Реакція:* Встановлення 10% швидкості від максимальної швидкості об'єкта. На дисплеї ЦБК з'являється напис, який підтверджує, що швидкість об'єкта була змінена «Speed Mode: Beginner».

*Подія:* Встановлення швидкості на ЦБК в режим Professional.

*Реакція:* Встановлення 30% швидкості від максимальної швидкості об'єкта. На дисплеї ЦБК з'являється напис, який підтверджує, що швидкість об'єкта було змінено «Speed Mode: Professional».

Також, прикладами вимог зацікавлених осіб до створюваної ВС УРО є наступні:

- STRQ1: об'єкт повинен бути дистанційно керованим;
- STRQ3: створити ефект присутності користувача в рухомому об'єкті;
- STRQ10: забезпечити збирання і зберігання станів системи за весь час роботи.

Дані вимоги в подальшому були сформульовані в системні вимоги:

- SYST3: передача даних від пульта управління до об'єкта повинна здійснюватися з використанням радіоканалу на відстані не менше 50 м;
- SYST4: передавати відео-зображення по радіоканалу від камери, встановленої на рухомому об'єкті на відео-окуляри користувача;
- SYST12: організувати зберігання даних про поточний стан системи на флеш-пам'ять, встановлену в ЦБК (таку вимогу можна трасувати в функціональну вимогу до ПЗ: програма повинна зберігати одержувані дані з ЦБК в форматі txt з розділювачами).

За результатами роботи з вимогами за допомогою програми RequisitePro, був розроблений комплект документації, що включає в себе всі вимоги до розроблюваної ВС УРО. Після того, як були сформульовані вимоги зацікавлених осіб та системні вимоги, був виконаний аналіз вимог та розроблено архітектуру системи.

### **4.3.3 Розробка архітектури системи**

У відповідності з удосконаленим методом паралельного проектування ВС, описаним у підрозділі 2.3, на наступному етапі було розроблено архітектуру проектованої системи [61, 66]. Вона складається з рухомого об'єкта (передбачено можливість підключення одного, двох або трьох об'єктів), центрального блоку керування, а також станції оператора (відповідно, для кожного рухомого об'єкта) (рис.4.14). Особливістю системи є наявність ЦБК, що дозволяє контролювати дії користувачів та стани керованих рухомих об'єктів, а також використання відео-окулярів для створення ефекту присутності користувача всередині рухомого об'єкта, що рухається в досліджуваній зоні.

На кожному рухомому об'єкті розміщено радіопередавач для взаємодії із ЦБК, камеру та відео передавач, що передає зображення на відео-окуляри оператора, панель індикації, блок живлення, а також мікроконтролерний блок,

до якого підключені ці компоненти. Мікроконтролерний блок рухомого об'єкта обробляє та виконує керуючу інформацію, отриману від ЦБК.

Станція оператора складається з джойстика для управління рухом об'єкта, а також відео-окулярів, на які у режимі реального часу відбувається передача відео зображення з рухомого об'єкта.

Запропонована архітектура системи керування рухомим об'єктом, дозволяє відокремити керуючий модуль і периферію (сенсори, актуатори). Функціонал керуючого модуля можна реалізувати як з використанням готової апаратно-програмної платформи, так і створюючи оригінальні рішення, в залежності від специфіки використання системи та фінансово-економічних обмежень.

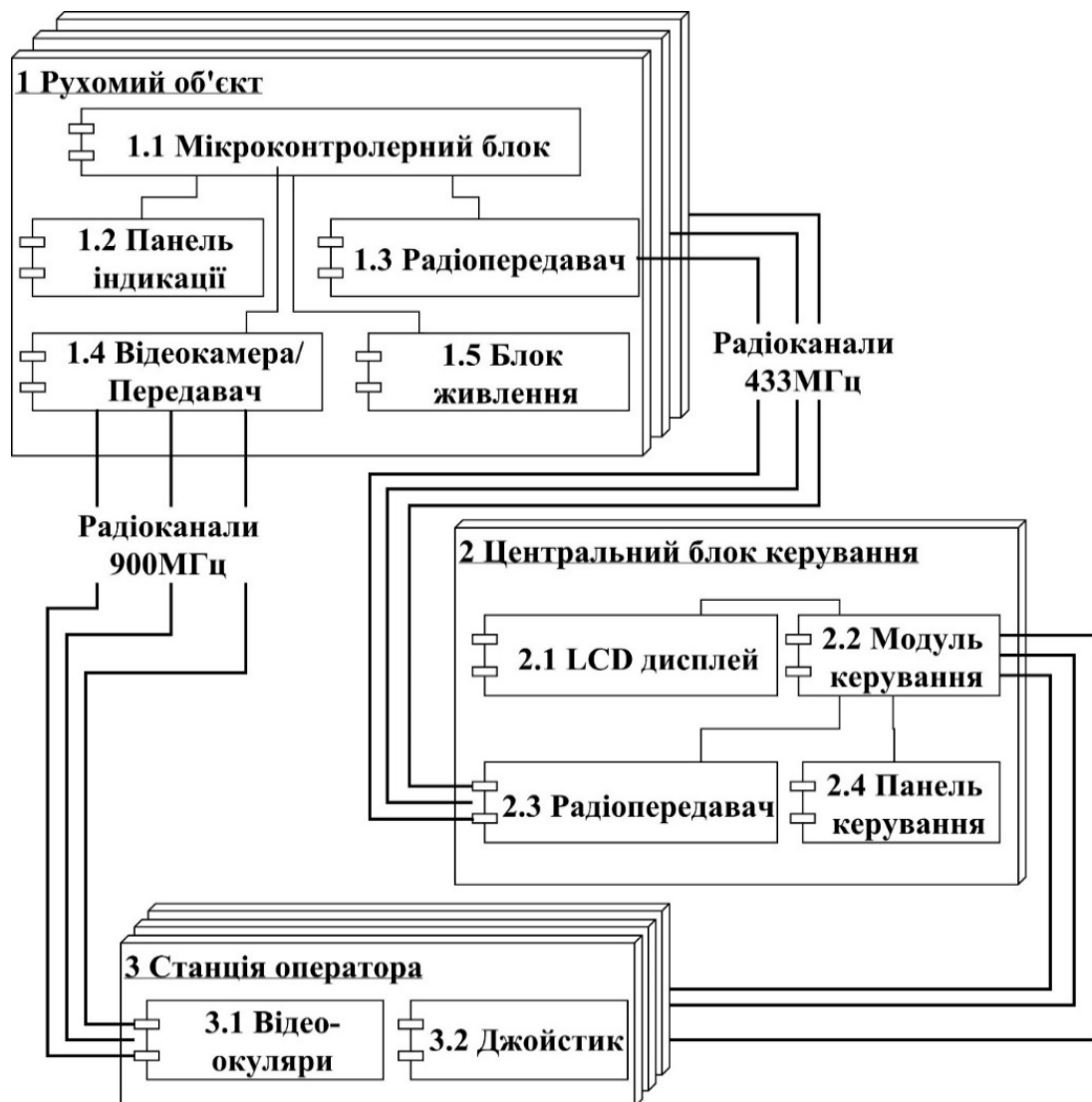


Рис. 4.14. Архітектура ВС УРО

ЦБК є найважливішою складовою системи, що забезпечує налаштування та контроль рухомих об'єктів (прив'язка до потрібного радіоканалу, калібрування положення передніх коліс, контроль швидкості руху, відстані до рухомого об'єкта та рівня заряду джерела живлення).

Розроблений алгоритм керування для ВС УРО наведено у вигляді UML діаграми послідовності дій (рис. 4.15). Після включення живлення системи та завантаження ЦБК, на його екрані відобразатиметься поточний стан та режим швидкості обраного каналу. За замовчуванням, у всіх операторів виставлений режим мінімальної швидкості руху об'єктів. Про запуск програми рухомого об'єкта свідчатиме загорання зеленого світлодіоду на панелі індикації рухомого об'єкта. На наступному кроці необхідно натиснути клавішу "Bind" на рухомому об'єкті для активації режиму прив'язки.

Загориться синій світлодіод, який свідчить про очікування команди від ЦБК із номером каналу, що буде закріплений за рухомим об'єктом. Після вибору каналу та натискання клавіші прив'язки на панелі ЦБК, синій світлодіод повинен згаснути. Загорання синього світлодіоду відбувається також при втраті зв'язку із ЦБК. Після цього, система готова до запуску. Для калібрування положення передніх коліс потрібно виконати перемикання відповідного режиму та за допомогою клавіш "FT+" і "FT-" виконати необхідні налаштування. Після натискання кнопки "Start" активується режим керування та почнеться передача відео-зображення на відео-окуляри оператора. За допомогою джойстика, підключеного до ЦБК, відбувається керування рухомим об'єктом.

Радіопередавачі, що використані в системі для передачі керуючої інформації, працюють у діапазоні LPD - Low Power Device (433 МГц), що має високу проникаючу здатність та є стійким до перешкод, це дозволяє з успіхом використовувати його для зв'язку в умовах забудови, крім того, обладнання для цього діапазону є досить компактним, що підвищує рівень мобільності та завадостійкості системи.

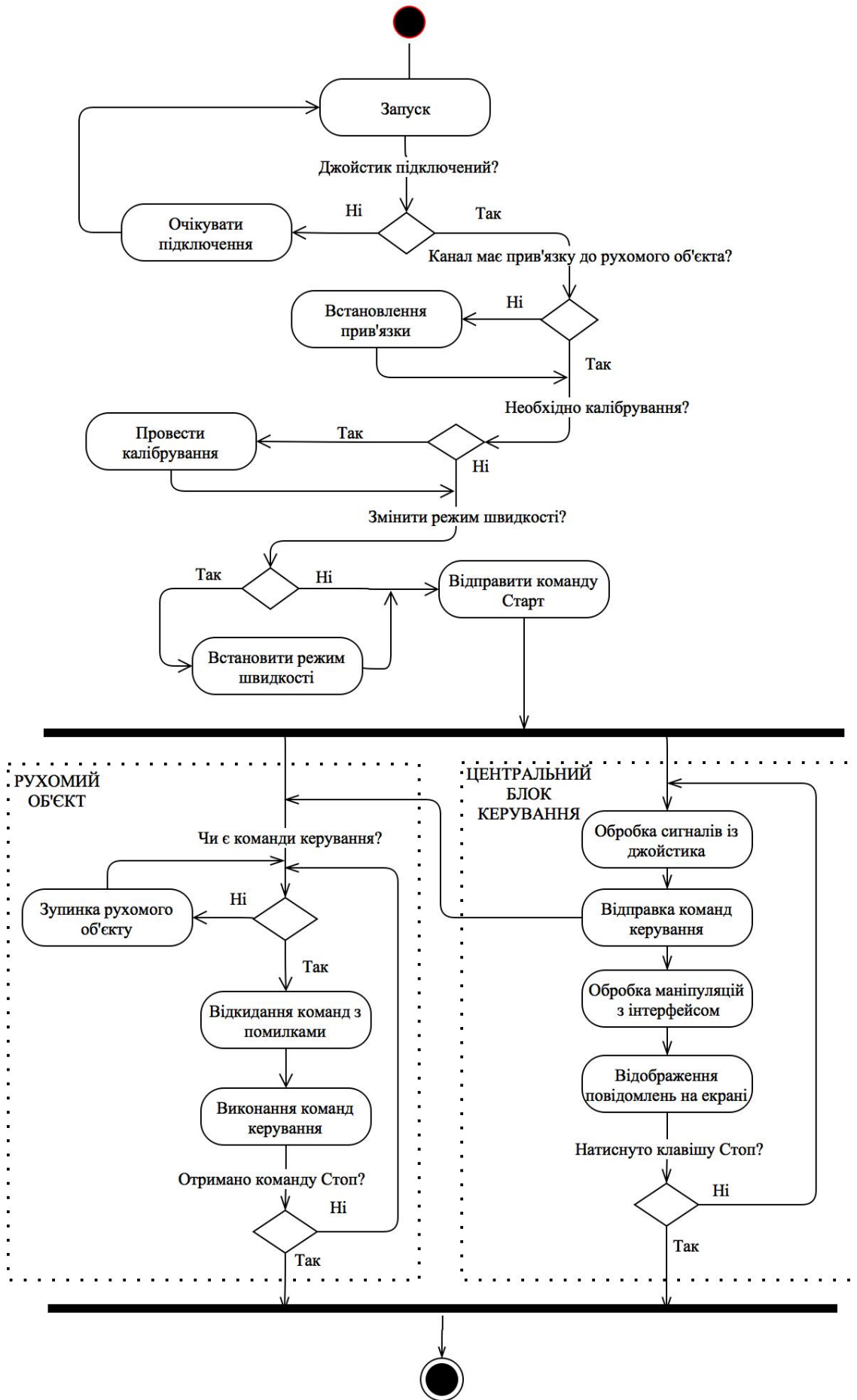


Рис. 4.15. Діаграма послідовності дій ВС УРО

Таким чином, використання для реалізації Модуля керування ЦБК готової апаратно-програмної платформи дозволить прискорити час розробки проекту, а Мікроконтролерний блок рухомого об'єкта, з метою мінімізації масо-габаритних показників конструкції, можливо реалізувати на оригінальній друкованій платі з використанням МК.

#### **4.3.4 Вибір апаратно-програмної платформи для реалізації прототипу центрального блоку керування**

Згідно з запропонованою в п. 4.3.1 методикою, наступним етапом став пошук апаратно-програмної платформи для Модуля керування ЦБК, яка б відповідала розробленим вимогам до системи, зокрема до ЦБК. Виявлено початкові дані: рівень знань розробника – початковий, планована кількість цифрових контактів = 18, кількість аналогових = 2, мова програмування – C/C++, живлення = 5V, габарити конструкції – середні.

Для формування рекомендацій щодо апаратно-програмних платформ було також задано обмеження за типами платформ (на основі МК або мінікомп'ютери). В результаті роботи РС, що входить до складу розробленого ПТК, за платформами на основі МК системою було рекомендовано (рис.4.16) платформу Arduino Mega (також, були надані 3 додаткові варіанти платформ, які не повністю відповідають вимогам, але близькі до них: Iskra Neo, MSP Launchpad, Arduino Uno). При пошуку серед мінікомп'ютерів (рис.4.17) точного збігу за всіма введеними вимогами знайдено не було, і система рекомендувала платформу Raspberry Pi 3 за окремими жорсткими критеріями.

Далі, прототип на основі апаратно-програмної платформи Arduino Mega був створений та досліджений з використанням ВЛ RELDES. Перевірка функціонування прототипу показала, що така реалізація дозволяє забезпечити необхідні функціональні вимоги до розроблюваного ЦБК. Крім того, було прийняте рішення про додаткове дослідження інших варіантів реалізації ЦБК з метою перевірки правильності сформованих рекомендацій.

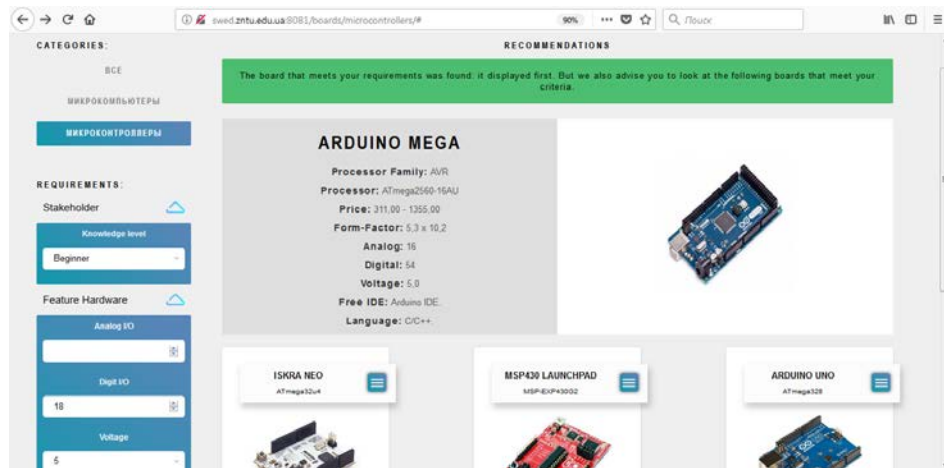


Рис. 4.16. Результати формування рекомендацій за всіма платформами

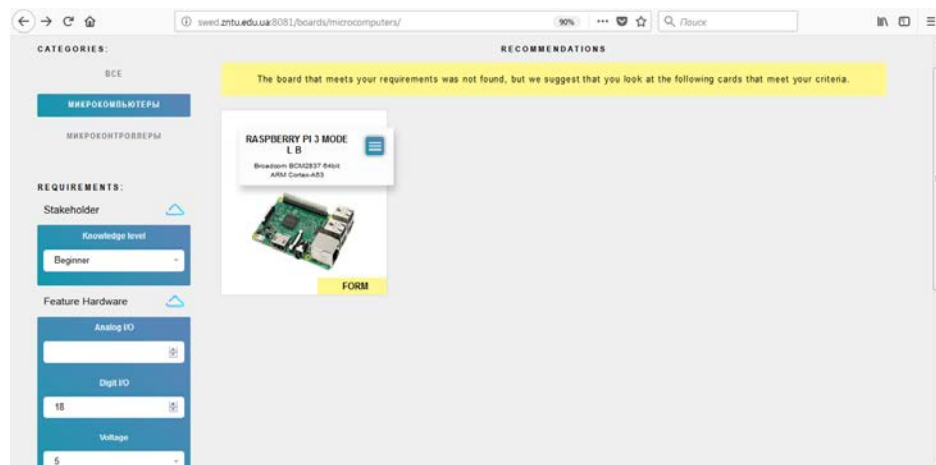


Рис. 4.17. Результати формування рекомендацій за мінікомп'ютерами

Таким чином, з використанням рекомендаційної системи, що входить до складу розробленого ПТК було виконано вибір апаратно-програмних платформ згідно з вимогами до проектованої системи та визначено можливі варіанти розвитку проекту.

#### 4.3.5 Дослідження варіантів реалізації прототипу вбудованої системи управління рухомими об'єктами

Згідно з методикою, запропонованою в п.4.3.1, було виконано створення прототипів ЦБК ВС УРО на основі апаратно-програмних платформ Arduino Mega та Raspberry Pi 3, а також на основі оригінальної плати з встановленим МК ATmega 16 [62-65].



Реалізація ЦБК на платформі Arduino є досить компактною та забезпечує можливість автономного живлення. Розроблене ПЗ дозволяє обробляти дії диспетчера з панеллю керування та відображати повідомлення на LCD-екрані, зчитувати аналогові значення з джойстика, нормалізувати їх та передавати за допомогою радіопередавача керуючу інформацію рухомому об'єкту з використанням UART протоколу. Програма також забезпечує вибір режиму швидкості, калібрування джойстика і одночасне керування декількома рухомими об'єктами. Але, як показали дослідження, реалізація ЦБК на основі Arduino потребує використання додаткової монтажної плати для збільшення кількості виводів землі і живлення та макетних з'єднань для взаємодії з платформою Arduino, що знижує надійність та компактність системи.

Як відомо, платформа Raspberry Pi 3 має високу обчислювальну потужність, але не має вбудованого аналого-цифрового перетворювача. Тому, додаткова друкована плата, що складається з 8-канального аналого-цифрового перетворювача MCP3008 та логічного перетворювача рівня, побудованого на двох транзисторах MOSFET, була створена для роботи з аналоговими пристроями. Ця плата дозволяє керувати аналоговими сигналами з джойстика та перетворювати логічні рівні сигналів 5 В та 3.3 В, отримані від радіопередавача. Середовище Altium Designer було обране для розв'язку завдання проектування електронної схеми, топології провідних шарів та тривимірної моделі друкованої плати. ПЗ було розроблено на мові C++ для операційної системи Raspbian з використанням середовища Qt Creator. Створена програма надає зручний інтерфейс, забезпечує обробку виключень та забезпечує функції для керування декількома рухомими об'єктами: прив'язка до потрібного каналу, вибір режиму швидкості, Старт/Стоп. З метою забезпечення одночасного управління трьома рухомими об'єктами, для кожного активного каналу створюється окремий потік, що є можливим завдяки використуваній операційній системі. Реалізація на Raspberry Pi дає більше можливостей для подальшого розширення функціоналу (обробка медіа, підключення до мережі та т.і.).

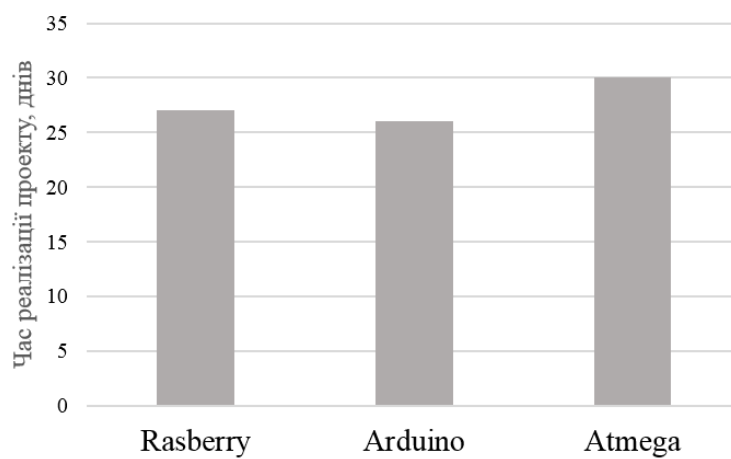
Розроблені прототипи ЦБК дали змогу перевірити правильність функціонування модулю керування ЦБК та сформувати ТЗ на розробку оригінальної плати на основі МК ATmega16. Розроблена з використанням системи Altium Designer оригінальна друкована плата не має надлишкового функціоналу, що властивий реалізації на Arduino або Raspberry Pi. Вартість дослідного зразка вища за вартість прототипу, що зумовлено додатковими витратами на проектування та виготовлення оригінальної плати.

Далі, були протестовані різні варіанти реалізації системи за двома сценаріями, мета яких полягала в оцінюванні робочих відстаней керування, що є головною метрикою якості системи. Перший сценарій передбачав вимірювання відстаней, на яких рухомий об'єкт повністю втрачає керуючий сигнал (в цьому випадку він зупиняється та загоряється спеціальний світлодіодний індикатор). За другим сценарієм здійснювалось вимірювання відстані, при якій якість переданого сигналу на відео-окулярах стає недостатньою для керування об'єктом (поява безперервних сильних перешкод на переданому зображенні або повна відсутність сигналу). Всі розроблені версії ЦБК показали майже однакові результати - приблизно 50 метрів. Результати порівняння варіантів реалізацій наведено в табл. 4.7 та на рис.4.18.

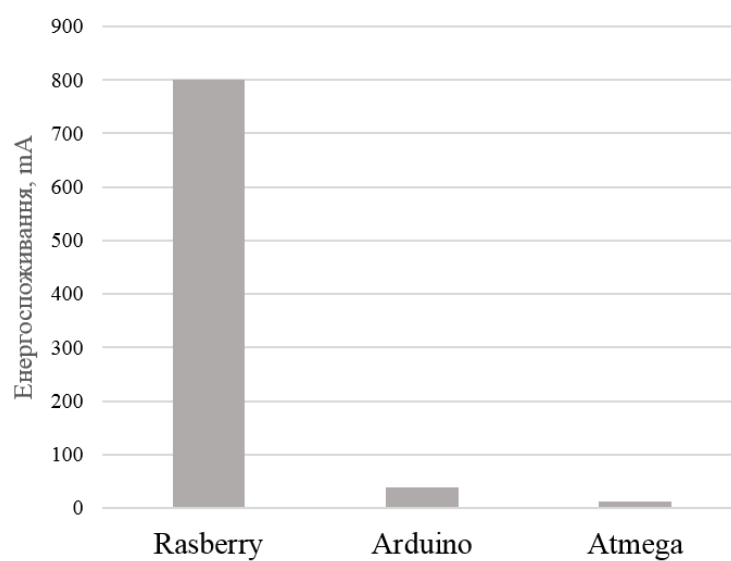
Таблиця 4.7.

Результати порівняння різних варіантів реалізації прототипу ВС УРО

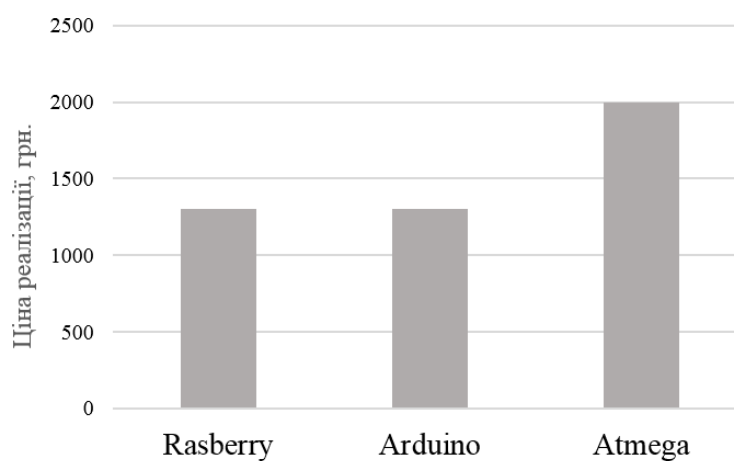
<b>Критерії порівняння</b>	<b>Варіант на Raspberry Pi</b>	<b>Варіант на Arduino Mega</b>	<b>Варіант на ATmega 16</b>
Робоча відстань керування	~50 метрів	~50 метрів	~50 метрів
Відстань якісного сигналу на відео-окулярах	~50 метрів	~50 метрів	~50 метрів
Ціна реалізації	~1300 грн.	~1300 грн.	~2000 грн.
Необхідність додаткових конструктивів	Аналого-цифровий перетворювач	Додаткова комунікаційна плата	Оригінальна друкована плата
Енергоспоживання	800mA	38mA	13mA
Час реалізації проекту	~ 27 днів	~26 днів	~ 30 днів



а)



б)



в)

Рис.4.18. Діаграми порівняння реалізацій прототипу ВС УРО за часом реалізації (а), енергоспоживанням (б) та ціною реалізації (в)

Відсоток економії часу за рахунок використання розробленого ПТК можна визначити за формулою (4.5):

$$E = \left(1 - \frac{T_p}{T_t}\right) \cdot 100\%, \quad (4.5)$$

де  $T_p$  – час проектування з використанням запропонованого методу проектування, днів;  $T_t$  – час проектування з використанням традиційного підходу, днів.

Підвищення продуктивності праці проектувальника визначається як:

$$P = \left(\frac{\Delta T}{T_t - \Delta T}\right) \cdot 100\%, \quad (4.6)$$

де  $\Delta T = T_t - T_p$  – показник економії часу, днів.

Таблиця 4.8.

Результати розрахунку показників ефективності проектування

Показник	На Arduino Mega	На Raspberry Pi 3
Економія часу	13%	10%
Підвищення продуктивності праці	15%	11%

Таким чином, використання готових програмно-апаратних платформ (Arduino Mega або Raspberry Pi) при розробці ВС УРО дозволяє прискорити процес проектування за рахунок швидкого прототипування ВС, розробити ТЗ на проектування оригінальних апаратних рішень за умови виявлення недоліків та проблем, а також виконати розробку та тестування системи (рис.2.4).

Оскільки якість відео зображення для дистанційно керованих об'єктів є важливим критерієм оцінки роботи системи, було проведено оцінювання метрик якості послідовності відео кадрів, отриманих з рухомого об'єкта на відео-окуляри оператора. Для оцінки якості отриманого відеосигналу використовувався інструмент оцінювання якості відео MSU Video Quality

Measurement Tool 9.1 [114]. Оскільки джерела відео, доступні для порівняння, знімалися тією ж камерою, але в різних умовах, відповідний список показників якості, що можуть бути застосовані, досить обмежений. Було обрано дві найбільш інформативних метрики: "MSU Brightness Flicking" (рис.4.19) та "MSU Noise Estimation" (рис.4.20). Це дозволило порівняти різну сукупність відео кадрів. MSU Brightness Flicking дозволяє оцінити коливання яскравості між сусідніми відео кадрами послідовності. MSU Noise Estimation дозволяє оцінити рівень шуму для кожного відео кадру послідовності.

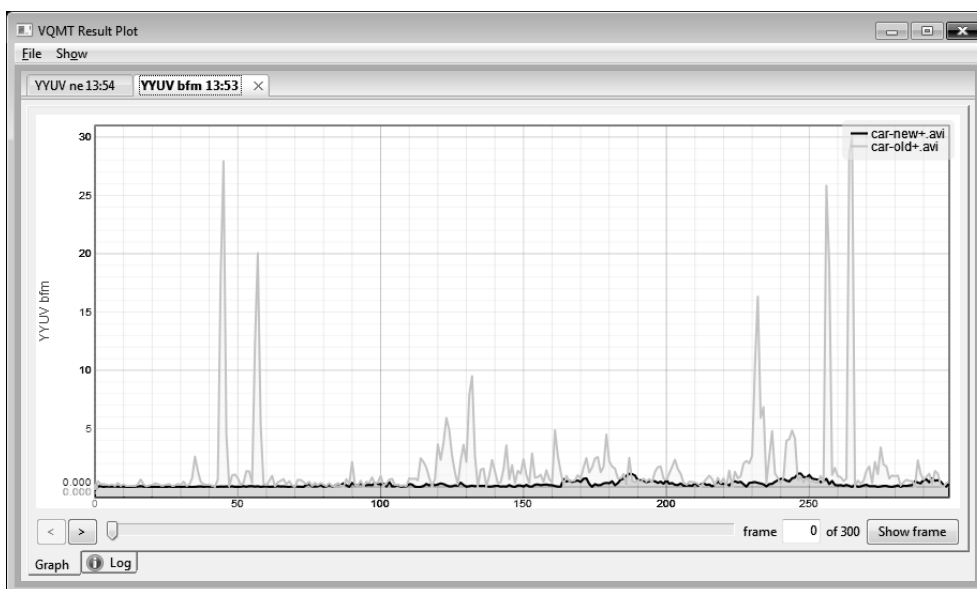


Рис. 4.19. ЦБК Brightness Flicking

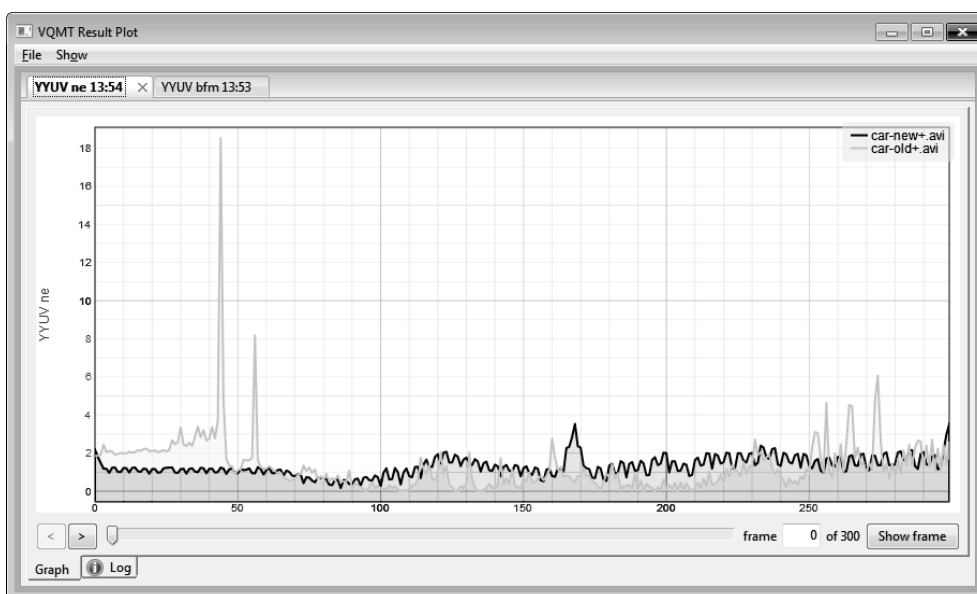


Рис. 4.20. ЦБК Noise Estimation

Аналіз виконувався для різних реалізацій модуля керування, зокрема для попередньої реалізації на основі МК ATmega8. Отримані графіки демонструють явне зниження коливання яскравості та рівня шуму для реалізації модуля керування ЦБК на основі оригінальної плати на МК ATmega16, що свідчить про правильність прийнятих проектних рішень.

Таким чином, використання розробленого програмно-технічного комплексу дозволило зменшити витрати часу на розробку прототипу ВС УРО на 10-13% за рахунок скорочення часу пошуку відповідної вимогам проекту апаратно-програмної платформи за допомогою рекомендаційної системи, а також швидкого прототипування та дослідження прототипу системи з використанням обладнання віддаленої лабораторії. Це сприяло підвищенню продуктивності праці проектувальників та, відповідно, зменшенню собівартості проекту.

#### **4.4 Висновки до розділу 4**

1. Розроблено архітектуру та структурну схему програмно-технічного комплексу автоматизації проектування ВС, які дозволяють реалізувати необхідні функціональні можливості ПТК та організувати ефективне використання засобів віддаленої інженерії.

2. Розроблено програмне та технічне забезпечення віддаленої лабораторії, що надає користувачам можливість спільного використання обладнання та програмного забезпечення для створення та дослідження прототипу проектованої ВС з метою зменшення матеріальних та часових витрат на розробку та, відповідно, собівартості проектованої продукції. Набір віддалених експериментів та апаратно-програмних платформ для їх реалізації може бути розширений для надання додаткових можливостей проектувальникам.

3. Розроблено програмне та інформаційне забезпечення рекомендаційної системи, яка надає підтримку проектувальнику, оскільки

дозволяє визначити доцільність застосування певної готової апаратно-програмної платформи, відповідно, підвищити рівень автоматизації проектувальних робіт та зменшити часові витрати на пошук проектних рішень.

4. Запропоновано методику застосування розробленого ПТК при АП ВС УРО, що може використовуватись для його ефективного впровадження в практику інженерного проектування.

5. Практичне застосування запропонованого ПТК при АП дозволило успішно реалізувати проект розробки ВС УРО та прискорити процес проектування на 10-13%.

Результати розділу опубліковано у наукових роботах [46, 50-52, 58, 60-66].

## ЗАГАЛЬНІ ВИСНОВКИ

У дисертаційній роботі розв'язано актуальне наукове завдання розроблення моделей, методів та інструментальних засобів ВІ для підвищення ефективності АП ВС за рахунок скорочення термінів проектування, зниження витрат на проектування та підвищення рівня автоматизації проектувальних робіт. Це було досягнуто завдяки реалізації технологій ВІ на основі:

- віддаленої лабораторії для організації спільного використання обладнання та програмного забезпечення, а також швидкого прототипування проекрованої ВС;

- рекомендаційної системи для автоматизованого вибору апаратно-програмної платформи при переході між системним та функціонально-логічним рівнями при АП ВС.

При цьому отримано такі наукові та практичні результати:

1. Вперше розроблено метод АП ВС з використанням технологій ВІ, що дає змогу організувати спільне використання обладнання та програмного забезпечення (без необхідності їх придбання та налагодження) для прототипування проекрованої ВС та, таким чином, знизити собівартість проекрованої продукції.

2. Отримав подальший розвиток метод паралельного проектування ВС, що відрізняється від існуючого спільним застосуванням паралельного та платформно-орієнтованого підходів, а також засобів ВІ, що дає змогу зменшити час переходу між системним та функціонально-логічним рівнями при проектуванні ВС та підвищити рівень автоматизації проектувальних робіт за рахунок автоматизованого вибору апаратно-програмної платформи.

3. Удосконалено модель формування вимог до ВС та метод роботи з ними, які на відміну від існуючих, враховують структурні особливості ВС, що містять апаратну та програмну складові, дають змогу розподіляти вимоги між програмними та апаратними компонентами ВС при АП та, таким чином, повніше враховувати вимоги до ВС та виконувати необхідні процеси роботи з



вимогами при АП.

4. Отримав подальший розвиток метод формування рекомендацій щодо апаратно-програмних платформ при АП ВС на основі знань, який на відміну від існуючого, поєднує у собі методи обмежень та подібних об'єктів, використовує методи багатокритеріального аналізу, що дає змогу надавати рекомендації навіть при суперечливості вимог користувача.

5. Розроблено структуру ПТК, що включає інструментарій ВІ, використання якого дає змогу підвищити ефективність процесу АП ВС за рахунок зменшення часу проектування та собівартості проекрованої продукції, а також підвищення рівня автоматизації проектувальних робіт.

6. Розроблено програмне та інформаційне забезпечення РС, яка надає підтримку проектувальнику у вигляді рекомендацій щодо використання апаратно-програмних платформ при АП ВС, виходячи з вимог до проекрованої ВС.

7. Розроблено програмне та технічне забезпечення ВЛ, яка дає змогу виконувати швидке прототипування проекрованої ВС, перевірку на сумісність і працездатність апаратної та програмної частин, тим самим зменшувати час реалізації проекту.

8. Розроблено методику застосування створеного ПТК при АП ВС, яка може використовуватись у практиці інженерного проектування ВС різного призначення для ефективного впровадження засобів ВІ з метою покращення фінансово-економічних показників проекту.

9. Практичне використання розробленого ПТК при АП ВС УРО дало змогу зменшити витрати часу на розробку на 10-13%, підвищити рівень автоматизації проектувальних робіт, а також продуктивність праці проектувальників на 11-15%.

10. Результати дисертаційної роботи впроваджено в процес АП систем керування об'єктами спецпризначення НВП ХАРТРОН-ЮКОМ, ВС керування комплексом електротехнічного обладнання ПКФ МОТОР та в навчальний процес Запорізького національного технічного університету

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ebert C., Jones C. Software: Facts, Figures, and Future. *Computer*. 2009. Vol. 42(4). P.42–52.
2. Embedded Systems: Technologies and Markets. *BCC Research*. URL: <http://www.bccresearch.com/market-research/information-technology/embedded-systems-technologies-markets-ift016d.html>. (Last accessed: 15.03.2018)
3. Embedded System Market – Global Industry Analysis, Size, Share, Growth, Trends and Forecast, 2015–2021. *Transparency Market Research*. URL: <http://www.transparencymarketresearch.com/embedded-system.html>. (Last accessed: 15.03.2018)
4. Embedded System Market Size. *Global Market Insight*. URL: <https://www.gminsights.com/industry-analysis/embedded-system-market>. (Last accessed: 15.03.2018)
5. Wiegers K. E. Software Requirements: 2nd Edition. Microsoft Press, 2003. 544 p.
6. Leffingwell D., Wording D. Managing Software Requirements: A Use Case Approach. Addison-Wesley, 2003. 402 p.
7. Cockburn A. Writing effective Use Case. Addison Wesley, 2000. 270 p.
8. Платунов А. Встраиваемые системы управления. *CONTROL ENGINEERING Россия*. 2013. № 1 (43). С. 16–24.
9. Smith C.U., Frank G.A., Cuadrado J.L. An architecture design and assessment system for software/hardware codesign. *DAC'85: proceedings of 22nd ACM/IEEE Design Automation Conference*. USA: IEEE Press Piscataway, NJ, 1985. P. 417–424.
10. Wolf H.W. Hardware-Software Co-Design of Embedded Systems. *IEEE: proceedings of the IEEE*, July 1994. Germany, 1994. Vol.82(7). P. 967–989.
11. Ptolemy: A mixed - Paradigm Simulation/Prototyping platform in C++ / Buck J., Ha S., Lee E.A., Messerschmitt D.G. *C++ At Work Conference: proceedings of the conference*, September 1991. Santa Clara, California, 1991.

12. System-Level Design: Orthogonalization of Concerns and Platform-Based Design / Keutzer K., Malik S., Richard Newton A., Rabaey M.J., Sangiovanni Vincentelli A. *IEEE transactions on computer-aided design of integrated circuits and systems*, 2000. Vol.19(12). P.1523–1543.
13. Teich J. Hardware/Software Codesign: The Past, the Present, and Predicting the Future. *IEEE: proceedings of the IEEE*, May 13, 2012. Germany, 2012. Vol.100. P. 1411–1429.
14. Лобур М.В. Особливості проектування вбудованих систем. *Вісник НУ «Львівська політехніка» № 501. Комп'ютерні системи проектування. Теорія і практика*. 2004. С.69–75.
15. Платунов А.Е., Постников Н.П. Высокоуровневое проектирование встраиваемых систем. Часть 1: учеб. пособие. СПб.: НИУ ИТМО, 2011. 121 с.
16. Сопряженное проектирование встраиваемых систем (Hardware/Software co-design). Часть 1. Учебное пособие / С.В. Быковский, Я.Г. Горбачев, А.О. Ключев, и др. СПб.: Университет ИТМО, 2016. 108 с.
17. A tradeoff between microcontroller, DSP, FPGA and ASIC technologies. *EE Times*. URL: [https://www.eetimes.com/document.asp?doc\\_id=1275272](https://www.eetimes.com/document.asp?doc_id=1275272). (Last accessed: 15.03.2018).
18. Барретт С.Ф., Пак Дж.Д. Встраиваемые системы. Проектирование приложений на микроконтроллерах семейства 68HC12/HCS12 с применением языка С. Издательский дом «ДМК-пресс», 2007. 643 с.
19. Berger S.A. *Embedded Systems Design: An Introduction to Processes, Tools, and Techniques*. CMP Books, 2002. 272 p.
20. Белов А.В. Создаем устройства на микроконтроллерах. СПб.: Наука и Техника, 2007. 304 с.
21. Васильев А.Е. Микроконтроллеры. Разработка встраиваемых приложений: уч.пособие. СПб.: БХВ-Петербург, 2008. 304 с.
22. Высокопроизводительные 8-разрядные RISC микроконтроллер семейства AVR. Рынок микроэлектроники. URL: <http://www.gaw.ru/html.cgi/txt/ic/Atmel/micros/avr/start.htm>. (дата звернення: 15.03.2018).

23. Ламбер Е. 8-разрядные микроконтроллеры AVR корпорации Atmel: новинки тенденции развития. *Компоненты и технологии*. 2009. №6. С. 62–65.
24. How to select your next development board. *Electronicsforu*. URL: <http://electronicsforu.com/buyers-guides/hardware-buyers-guide/select-next-development-board/3>. (Last accessed: 15.03.2018).
25. Choosing the right embedded development board. *Electronicsforu*. URL: <http://electronicsforu.com/buyers-guides/hardware-buyers-guide/embedded-development-board>. (Last accessed: 15.03.2018).
26. Гид по выбору платформы разработки. *Амперка*. URL: <http://amperka.ru/page/development-board-guide>. (дата звернення: 15.03.2018).
27. *Recomender Systems: Handbook* / Ricci F., Rokach L., Shapira B., Kantor P.B. Springer Science+Business Media, LLC. 2011. 845 p.
28. Burke R. Hybrid web recommender systems. *The Adaptive Web - Methods and Strategies of Web Personalization* / P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.). Springer-Verlag Berlin Heidelberg. 2007. P. 377–408.
29. Tarus K.J., Niu1 Z., Mustafa G. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*. 2017. P. 1–28.
30. Fuzzy models for recommendation systems / Stekh Y., Lobur M., Artsibasov V., Glushko O. *CAD in Machinery Design – Implementation and Educational Problems: proceedings of the XXIII Polish-Ukrainian Conference, 10 - 11 October, 2014. Lviv, 2014*. P. 155–159.
31. Дьяконов А.Г. Алгоритмы для рекомендательной системы: технология LENKOR. *Бизнес информатика*. 2012. №1(19). С. 32–39.
32. Пятикоп Е.Е. Исследование метода коллаборативной фильтрации на основе сходства элементов. *Наукові праці ДонНТУ, Серія “Інформатика, кібернетика та обчислювальна техніка”*. 2013. Вип. 2(18). С. 109–114.
33. Internet shop “Arduino in Ukraine”. *Arduino-ua*. URL: <https://arduino.ua/>. (Last accessed: 15.03.2018).

34. Электронные компоненты. *Космодром*. URL: <http://www.kosmodrom.com.ua>. (дата звернення: 15.03.2018).
35. WEBENCH® Design Center. *Texas Instruments*. URL: <http://www.ti.com/lscds/ti/analog/webench/overview.page>. (Last accessed: 15.03.2018).
36. STM32 Open Development Environment. URL: <http://www.st.com/en/ecosystems/stm32-open-development-environment.html>. (Last accessed: 15.03.2018).
37. The Grid of Online Laboratory Devices Ilmenau (GOLDi). URL: <http://www.goldi-labs.net>. (Last accessed: 15.03.2018).
38. WebLab-Deusto. URL: <http://weblab.deusto.es/website>. (Last accessed: 15.03.2018).
39. Remote Labs. Enriching digital education. URL: <http://www.labshare.edu.au>. (Last accessed: 15.03.2018).
40. e-LABORATORY PROJECT. URL: <http://www.ises.info/index.php/en>. (Last accessed: 15.03.2018).
41. The iLab Project. URL: <https://icampus.mit.edu/projects/ilabs>. (Last accessed: 15.03.2018).
42. Parkhomenko A., Gladkova O. Analysis and application of existent approaches in microcontroller system designing. *Perspective technologies and methods in MEMS design: proceedings of IX international conference, 16-20 April 2013*. Lviv: NU “Lviv Polytechnic”, 2013. P. 59–61.
43. Исследование современных технологий автоматизированного проектирования электронных устройств на микроконтроллерах / Поздняков О.А., Гладкова О.Н., Пархоменко А.В., Цветков О.В. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: тези доповіді VI міжнародної науково-практичної конференції, 19-21 вересня 2012 р. Запоріжжя: ЗНТУ, 2012. С. 248–250.*
44. Parkhomenko A., Gladkova O. Investigation of peculiarities of analysis of system and software requirements for designing automated system. *The experience*

*of designing and application of CAD systems in microelectronics*: proceedings of XII international conference, 19-23 February 2013. Lviv: NU “Lviv Polytechnic”, 2013. P. 268–270.

45. Parkhomenko A., Gladkova O. Development and practical application of requirements model for designing embedded systems. *Perspective technologies and methods in MEMS design* : proceedings of X international conference, 22-24 June 2014. Lviv: NU “Lviv Polytechnic”, 2014. P. 20–22.

46. Parkhomenko A., Gladkova O. Complex requirements analysis for the high-level design of Embedded Systems. *Вісник НУ «Львівська політехніка». Серія «Комп'ютерні системи проектування. Теорія і практика»*. 2014. № 808. С. 3–9.

47. Пархоменко А., Гладкова О. Інтерактивна віддалена лабораторія дослідження апаратно-програмних платформ. *Інтернет-Освіта-Наука-2014*: тези доповіді міжнародної науково-практичної конференції, 14-17 жовтня 2014 р. Вінниця: ВНТУ, 2014. С. 111–113.

48. Пархоменко А.В., Гладкова О.Н., Иванов Е.В. Исследование особенностей создания и применения лаборатории удаленного доступа при решении задач проектирования встроенных систем. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій*: тези доповіді VII міжнародної науково-практичної конференції, 17-19 вересня 2014 р. Запоріжжя: ЗНТУ, 2014. С. 205–206.

49. Internet-based Technologies for design of embedded systems / A.Parkhomenko, O. Gladkova et al. *The experience of designing and application of CAD systems in microelectronics*: proceedings of 13th international conference, 24-27 February, 2015. Lviv: NU “Lviv Polytechnic”, 2015. P. 167–171.

50. Internet-based technologies for design of embedded systems / A. Parkhomenko, O. Gladkova et al. *Journal of Control Science and Engineering*. 2015. Vol.3(2). P. 55–63.

51. Investigation of remote lab design technologies / A. Parkhomenko, A. Sokolyanskii, O. Gladkova, S. Kurson. *Perspective technologies and methods in MEMS design*: proceedings of XI international conference, 2-6 September, 2015. Lviv:

NU “Lviv Polytechnic”, 2015. P. 92–95.

52. Parkhomenko A.V., Gladkova O.N. Virtual tools and collaborative working environment in embedded system design. *Remote engineering and virtual instrumentation: proceedings of XI international conference, 26-28 February, 2014. Porto (Portugal), 2014. P. 91–93.*

53. Development and application of remote laboratory for embedded systems design / A. Parkhomenko, O. Gladkova et al. *Remote engineering and virtual instrumentation: proceedings of 13th international conference, 25-28 February, 2015. Bangkok (Thailand), 2015. P. 69–73.*

54. Development and application of remote laboratory for embedded systems design / A. Parkhomenko, O. Gladkova, E. Ivanov et al. *International Journal of Online Engineering. 2015. Vol.11(3). P. 27–31.*

55. Reusable solutions for embedded systems’ design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, et al. *Remote engineering and virtual instrumentation: proceedings of 13th international conference, 24-26 February, 2016. Madrid (Spain), 2016. P. 313–317.*

56. Investigation of reuse concepts for embedded systems design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, Ya. Zalyubovskiy. *Perspective technologies and methods in MEMS design: proceedings of XII international conference, 20-24 April, 2016. Lviv: NU “Lviv Polytechnic”, 2016. P.78–80.*

57. Гладкова О.М. Підходи та особливості проектування RESTful API. *Тиждень науки – 2016: тези доп. щоріч. наук.-практ. конф. викладачів, науковців, молодих учених, аспірантів, студентів ЗНТУ, 18–22 квіт. 2016 р. Запоріжжя: ЗНТУ, 2016. С. 562–564.*

58. Implementation of reusable solutions for remote laboratory development / A. Parkhomenko, O. Gladkova, A. Sokolyanskii et al. *International Journal of Online Engineering. 2016. Vol.12(7). P. 24–29.*

59. Гладкова О.М., Пархоменко А.В. Дослідження та практична реалізація рекомендаційної системи для вибору апаратно-програмних платформ при автоматизованому проектуванні вбудованих систем. *Наукові праці ДонНТУ.*

Серія «Інформатика, кібернетика та обчислювальна техніка». 2017. №2(25). С. 22–31.

60. Свідоцтво № 64253 Україна. Комп'ютерна програма “Віддалена лабораторія проектування вбудованих систем” / А.В. Пархоменко, О.М. Гладкова, О.В. Соколянський; Власник ЗНТУ; дата реєстрації в ДСІВ України 26.02.2016 р.

61. Свідоцтво № 67404 Україна. Комп'ютерна програма “Автоматизована система управління рухомим об'єктом” / А.В. Пархоменко, О.М. Гладкова; власник ЗНТУ; дата реєстрації в ДСІВ України 22.08.2016 р.

62. Modernization of mobile object control system based on Raspberry Pi and Arduino platforms / A. Parkhomenko, O. Kravchenko, D. Kravchenko, O. Gladkova. *Embedded systems and trends in teaching engineering: proceedings of the international symposium, 12-15 September, 2016*. Nitra (Slovakia): Constantine the Philosopher University, 2016. P.249–253.

63. Challenges and solutions for mobile object control system / D. Kravchenko, O. Kravchenko, A. Parkhomenko, O. Gladkova. *Intelligent data acquisition and advanced computing systems: technology and applications: proceedings of 10th IEEE international conference, 21-23 September, 2017*. Bucharest (Romania), 2017. P.988–993.

64. Path finding algorithm for moving robots and obstacles avoidance / S. Bоеckx, P. Pelgrims, A. Parkhomenko, O. Gladkova et al. *Ambient intelligence and embedded systems: proceedings of the international symposium, 14-16 September, 2017*. Vaasa (Finland). URL: <http://amies-2017.international-symposium.org/proceedings.html>. (Last accessed: 15.03.2018).

65. Дослідження та розробка автоматизованої системи віддаленого керування групою рухомих об'єктів / А.В. Пархоменко, О.М. Гладкова, О.П. Кравченко, Д.П. Кравченко. *Вісник СХУ ім. В. Даля*. 2017. №8(238). С. 67–74.

66. Система автоматизованого керування групою рухомих об'єктів / А.В. Пархоменко, О.М. Гладкова, О.П. Кравченко: пат.123942 Україна: МПК G05B 19/05. № u201710387 ; заявл. 27.10.2017 ; опубл. 12.03.2018, Бюл. №5.



67. Shibu K.V. Introduction to Embedded Systems. Tata McGraw-Hill Education, 2009. 740 p.
68. Unmanned vehicle. URL: [https://en.wikipedia.org/wiki/Uncrewed\\_vehicle](https://en.wikipedia.org/wiki/Uncrewed_vehicle). (Last accessed: 15.03.2018).
69. Дистанційно керований комплекс «Фантом» випробували у Гончарівську. *Україна молода*. URL: <http://www.umoloda.kiev.ua/number/0/2006/114085>. (дата звернення: 15.03.2018).
70. Бойовий дистанційно керований багатофункціональний комплекс «Піранья». *Військова панорам*. URL: <https://wartime.org.ua/30001-boyoviy-distancyno-kerovaniy-bagatofunkcionalniy-kompleks-pranya.html>. (дата звернення: 15.03.2018).
71. Посметьев В.И., Никонов В.О., Посметьев В.В. Оценка актуальности и обоснование выбора схемы лесовозного автомобиля с дистанционным управлением. *Лесотехнический журнал*. 2017. Т. 7. №1 (25). С. 211-218.
72. Full Self-Driving Hardware on All Cars. *Tesla*. URL: <https://www.tesla.com/autopilot?redirect=no>. (Last accessed: 15.03.2018).
73. Перший безпілотний КрАЗ - перший «розумний» український автомобіль. *KRAZ*. URL: <http://www.autokraz.com.ua/index.php/uk/novini-ta-media/news/item/2839-pershyi-bezpilotnyi-kraz-pershyi-rozumnyi-ukrainskyi-avtomobil>. (дата звернення: 15.03.2018).
74. Дистанционное управление. URL: [https://ru.wikipedia.org/wiki/Дистанционное\\_управление](https://ru.wikipedia.org/wiki/Дистанционное_управление). (дата звернення: 15.03.2018).
75. Vehicle control unit based on radio communication. *Espacenet Patent search*. URL: [https://worldwide.espacenet.com/publicationDetails/biblio?adjacent=true&locale=en\\_EP&FT=D&date=20170329&CC=CN&NR=206049553U&KC=U%23#](https://worldwide.espacenet.com/publicationDetails/biblio?adjacent=true&locale=en_EP&FT=D&date=20170329&CC=CN&NR=206049553U&KC=U%23#). (Last accessed: 15.03.2018).
76. Radio controlled car APP platform race system. *Espacenet Patent search*. URL: <https://worldwide.espacenet.com/publicationDetails/>

biblio?DB=EPODOC&II=1&ND=3&adjacent=true&locale=en\_EP&FT=D&date=20170426&CC=CN&NR=106600433A&KC=A#. (Last accessed: 15.03.2018).

77. Система автоматизованого керування робочим органом транспортної мережі за допомогою радіо керування. *Цифрова патентна бібліотека*. URL: [http://library.uipv.org/document?fund=2&id=126978&to\\_fund=2](http://library.uipv.org/document?fund=2&id=126978&to_fund=2). (Last accessed: 15.03.2018).

78. The New Embedded System Design Methodology For Improving Design Process Performance / M. Abdurohman, Kuspriyanto, S. Sutikno, A. Sasongko. *International Journal of Computer Science and Information Security*. 2010. Vol. 8(1). P. 35–43.

79. Пархоменко А.В., Гладкова О.Н., Вершинин Р.А. Разработка информационной технологии проектирования электронных средств с применением PRO/ENGINEER и ALTIUM. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: тези доповідей VI Міжнародної науково-практичної конференції, 19–21 вересня 2012 р. Запоріжжя: ЗНТУ, 2012. С. 251–253.*

80. Tero V. An embedded object approach to embedded system development. OULU University press, 2009. 130 p.

81. Platform-based design and development: current trends and needs in industry / T.W. Simpsonand, T. Marion et al. *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference: proceedings of conference, 10-13 September, 2006. Philadelphia, Pennsylvania, USA, 2006. P. 1–10.*

82. Parkhomenko A., Gladkova O. Investigation of peculiarities software development for embedded systems. *Modern Problems of Radio Engineering, Telecommunications, and Computer Science: proceedings of XII-th international conference, February 25–March 1, 2014. Lviv-Slavske, Ukraine, 2014. P.382–383*

83. Ежегодное исследование рынка встраиваемых систем. *Электронные компоненты*. URL: [http://www.elcomdesign.ru/reviews/reviews\\_146.html](http://www.elcomdesign.ru/reviews/reviews_146.html). (Last accessed: 15.03.2018).

84. ДСТУ ISO/IEC 14288:2004. Інформаційні технології. Процеси життєвого циклу системи (ISO/IEC 14288:2002, IDT). [Чинний від 2007-07-01]. К.: Держстандарт України, 2004. 48 с.

85. ГОСТ Р ИСО/МЭС 12207-2010. Information technology. System and software engineering. Software life cycle processes (ISO/IEC 12207:2008). [Введ. 2012-03-01]. Москва: Стандартинформ, 2011. 104с.

86. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. [Введ. 1990-01-01]. М.: ИПК издательство стандартов, 2004. 11 с.

87. ГОСТ Р 41904-2002. Программное обеспечение встроенных систем. Общие требования к разработке и документированию. [Введ. 2002-07-24]. ИПК Издательство стандартов, 2002. 62с.

88. Основы программной инженерии. *SWEBOK*. URL: [www/ URL: http://swebok.sorlik.ru](http://www.swebok.sorlik.ru). (дата звернення: 15.03.2018).

89. Формализация требований на практике / В.В. Кулямин, Н.В. Пакулин, О.Л. Петренко та ін. М.: ИСП РАН. 2006. 50 с.

90. Online engineering. *Wikipedia*. URL: [https://en.wikipedia.org/wiki/Online\\_engineering](https://en.wikipedia.org/wiki/Online_engineering). (Last accessed: 15.03.2018).

91. International Association of Online Engineering. URL: <http://online-engineering.org/>. (Last accessed: 15.03.2018).

92. PCB design. *Autodesk Circuits*. URL: <https://circuits.io/pcb>. (Last accessed: 15.03.2018).

93. EasyEDA. URL: <http://easyeda.com>. (Last accessed: 15.03.2018).

94. Virtual Breadboard. URL: <http://www.virtualbreadboard.com/Index.html>. (Last accessed: 15.03.2018).

95. Simulator for Arduino. *Vitronics*. URL: <http://www.vitronics.com.au/Simulator-for-Arduino.html>. (Last accessed: 15.03.2018).

96. Simuino. URL: <https://code.google.com/p/simuino/>. (Last accessed: 15.03.2018).

97. Tinkercad. URL: <https://www.tinkercad.com/>. (Last accessed: 15.03.2018).
98. Raspberry Pi Azure IoT Online Simulator. URL: <https://azure-samples.github.io/raspberry-pi-web-simulator/>. (Last accessed: 15.03.2018).
99. Arduino. URL: <https://www.arduino.cc/>(Last accessed: 27.03.2018).
100. RaspberryPi. URL: <https://www.raspberrypi.org/products/> (Last accessed: 27.03.2018).
101. Parametric Search Tool. *Microchip*. URL: <http://www.microchip.com/ParamChartSearch/chart.aspx?branchID=1005>. (Last accessed: 15.03.2018).
102. Isinkaye F.O., Folajimi Y.O., Ojokoh B.A. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*. 2015. Vol. 16(3). P. 261–273
103. Recommender Systems. An Introduction / D. Jannach, M. Zanker, A. Felfernig, G. Friedrich. Cambridge University Press, 2011. 352 p.
104. Гомзин А.Г., Коршунов А.В. Системы рекомендаций обзор современных подходов. *Труды института системного программирования РАН*. 2012. Т.22, С. 401–417
105. Белоцкий Е.А., Суети А.В. Построение рекомендательной системы по подбору высших учебных заведений для абитуриентов. *Вестник СПбГУ. Сер.10. Прикладная математика. Информатика*. 2016. Вып. 1. С. 66– 77
106. Pazzani J.M., Billsus D. Content-Based Recommendation Systems. *The Adaptive Web - Methods and Strategies of Web Personalization* / P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.). Springer-Verlag Berlin Heidelberg. 2007. P. 325–341.
107. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.
108. O'Mahony M.P., Hurley N.J., Silvestre G.C.M. An Evaluation of the Performance of Collaborative Filtering. *Artificial Intelligence & Cognitive Science:*

proceedings of 14th Irish Conference, 19 December, 2003. P. 171–175.

109. Huang A. Similarity Measures for Text Document Clustering. *NZCSRSC: proceedings of 6th New Zealand Computer Science Research Student Conference*, April, 2008. New Zealand. 2008. P. 49–56.

110. Rafi M., Shaikh M.Sh. An improved semantic similarity measure for document clustering based on topic maps. *Cornell University Library*. URL: <https://arxiv.org/ftp/arxiv/papers/1303/1303.4087.pdf>. (Last accessed: 15.03.2018).

111. Антонова А.Ю., Клышинский Э.С. Об использовании мер сходства при анализе документации. *Электронные библиотеки: перспективные методы и технологии, электронные коллекции: труды 13 Всероссийской научной конференции*, 2011. Воронеж, Россия, 2011. С. 134–138.

112. Lathia N., Hailes S., Capra L. The effect of correlation coefficients on communities of recommenders. *SAC'08: proceedings of 23rd Annual ACM Symposium on Applied computing*. Fortaleza, Ceara (Brazil), 16–20 March, 2008. P. 2000–2005.

113. Кузнецов В.І., Євтушенко Г.Л. Системне моделювання складних об'єктів на базі методів багатокритеріального аналізу. *Системные технологии моделирования сложных систем: монография / под общей редакцией проф. А.И. Михалёва*. Днепр: НМетАУ-ИБК “Системные технологии”, 2016. С. 349–374.

114. MSU Video Quality Measurement Tool. *Compression*. URL: [http://www.compression.ru/video/quality\\_measure/vqmt\\_download.html](http://www.compression.ru/video/quality_measure/vqmt_download.html) (Last accessed: 15.03.2018).

**Список публікацій здобувача за темою дисертації та відомості  
про апробацію результатів дисертації**

**Список публікацій за темою дисертації:**

1. Parkhomenko A., Gladkova O. Complex requirements analysis for the high-level design of Embedded Systems. *Вісник НУ «Львівська політехніка». Серія «Комп'ютерні системи проектування. Теорія і практика»*. 2014. № 808. С. 3–9.
2. Гладкова О.М., Пархоменко А.В. Дослідження та практична реалізація рекомендаційної системи для вибору апаратно-програмних платформ при автоматизованому проектуванні вбудованих систем. *Наукові праці ДонНТУ. Серія «Інформатика, кібернетика та обчислювальна техніка»*. 2017. № 2(25). С. 22–31.
3. Дослідження та розробка автоматизованої системи віддаленого керування групою рухомих об'єктів / А.В. Пархоменко, О.М. Гладкова, О.П. Кравченко, Д.П. Кравченко. *Вісник СХУ ім. В. Даля*. 2017. № 8(238). С. 67–74.
4. Implementation of reusable solutions for remote laboratory development / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Y. Zalyubovskiy. *International Journal of Online Engineering*. 2016. Vol. 12(7). P. 24–29. (SCOPUS, Web of Science)
5. Development and application of remote laboratory for embedded systems design / A. Parkhomenko, O. Gladkova, E. Ivanov, A. Sokolyanskii, S. Kurson. *International Journal of Online Engineering*. 2015. Vol.11(3). P.27–31. (SCOPUS, Web of Science)
6. Internet-based technologies for design of embedded systems / A. Parkhomenko, O. Gladkova, S. Kurson, A. Sokolyanskii, E. Ivanov. *Journal of Control Science and Engineering*. 2015. Vol. 3(2). P. 55–63. (ProQuest, CrossRef)
7. Система автоматизованого керування групою рухомих об'єктів / А.В. Пархоменко, О.М. Гладкова, О.П. Кравченко: пат.123942 Україна: МПК G05B 19/05. № u2017110387 ; заявл. 27.10.2017 ; опубл. 12.03.2018, Бюл. №5.
8. Свідоцтво № 64253 Україна. Комп'ютерна програма “Віддалена

лабораторія проектування вбудованих систем” / А.В. Пархоменко, О.М. Гладкова, О.В. Соколянський; Власник ЗНТУ; дата реєстрації в ДСІВ України 26.02.2016 р.

9. Свідоцтво № 67404 Україна. Комп'ютерна програма “Автоматизована система управління рухомим об'єктом” / А.В. Пархоменко, О.М. Гладкова; власник ЗНТУ; дата реєстрації в ДСІВ України 22.08.2016 р.
10. Challenges and solutions for mobile object control system / D. Kravchenko, O. Kravchenko, A. Parkhomenko, O. Gladkova. *Intelligent data acquisition and advanced computing systems: technology and applications: proceedings of 10th IEEE international conference, 21-23 September, 2017. Bucharest (Romania), 2017. P.988–993.*
11. Path finding algorithm for moving robots and obstacles avoidance / S. Boeckx, P. Pelgrims, A. Parkhomenko, O. Gladkova, D.V. Merode. *Ambient intelligence and embedded systems: proceedings of the international symposium, 14-16 September, 2017. Vaasa (Finland). URL: <http://amies-2017.international-symposium.org/proceedings.html>*
12. Modernization of mobile object control system based on Raspberry Pi and Arduino platforms / A. Parkhomenko, O. Kravchenko, D. Kravchenko, O. Gladkova. *Embedded systems and trends in teaching engineering: proceedings of the international symposium, 12-15 September, 2016. Nitra (Slovakia): Constantine the Philosopher University, 2016. P. 249–253.*
13. Reusable solutions for embedded systems' design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, V. Shepelenko, Ya. Zalyubovskiy. *Remote engineering and virtual instrumentation: proceedings of 13th international conference, 24-26 February, 2016. Madrid (Spain), 2016. P. 313–317.*
14. Investigation of reuse concepts for embedded systems design / A. Parkhomenko, O. Gladkova, A. Sokolyanskii, Ya. Zalyubovskiy. *Perspective technologies and methods in MEMS design: proceedings of XII international conference, 20-24 April, 2016. Lviv: NU “Lviv Polytechnic”, 2016. P. 78–80.*
15. Гладкова О.М. Підходи та особливості проектування RESTful API.



*Тиждень науки – 2016*: тези доп. щоріч. наук.-практ. конф. викладачів, науковців, молодих учених, аспірантів, студентів ЗНТУ, 18–22 квітня 2016 р. Запоріжжя: ЗНТУ, 2016. С. 562–564.

16. Investigation of remote lab design technologies / A. Parkhomenko, A. Sokolyanskii, O. Gladkova, S. Kurson. *Perspective technologies and methods in MEMS design*: proceedings of XI international conference, September 2-6, 2015. Lviv: NU “Lviv Polytechnic”, 2015. P. 92–95.
17. Пархоменко А., Гладкова О. Інтерактивна віддалена лабораторія дослідження апаратно-програмних платформ. *Інтернет-Освіта-Наука-2014*: тези доповіді міжнародної науково-практичної конференції, 14-17 жовтня 2014 р. Вінниця: ВНТУ, 2014. С. 111–113.
18. Пархоменко А.В., Гладкова О.Н., Иванов Е.В. Исследование особенностей создания и применения лаборатории удаленного доступа при решении задач проектирования встроенных систем. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій*: тези доповіді VII міжнародної науково-практичної конференції, 17-19 вересня 2014 р. Запоріжжя: ЗНТУ, 2014. С. 205–206.
19. Parkhomenko A.V., Gladkova O.N. Virtual tools and collaborative working environment in embedded system design. *Remote engineering and virtual instrumentation*: proceedings of XI international conference, 26-28 February, 2014. Porto (Portugal), 2014. P. 91–93.
20. Parkhomenko A., Gladkova O. Analysis and application of existent approaches in microcontroller system designing. *Perspective technologies and methods in MEMS design*: proceedings of IX international conference, 16-20 April, 2013. Lviv: NU “Lviv Polytechnic”, 2013. P. 59–61.
21. Parkhomenko A., Gladkova O. Investigation of peculiarities of analysis of system and software requirements for designing automated system. *The experience of designing and application of CAD systems in microelectronics*: proceedings of XII international conference, 19-23 February, 2013. Lviv: NU “Lviv Polytechnic”, 2013. P.268–270.

22. Исследование современных технологий автоматизированного проектирования электронных устройств на микроконтроллерах / О.А. Поздняков, О.Н. Гладкова, А.В. Пархоменко, О.В. Цветков. *Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій: тези доповіді VI міжнародної науково-практичної конференції, 19-21 вересня 2012 р. Запоріжжя: ЗНТУ, 2012. С. 248–250.*

***Відомості про апробацію результатів дисертації:***

*1. міжнародні конференції:*

- «Data acquisition and advanced computing systems: technology and applications» (Бухарест, Румунія, 2017);
- «Remote engineering and virtual instrumentation» (Порто, Португалія, 2014; Мадрид, Іспанія, 2016);
- «Перспективні технології і методи проектування МЕМС» (Львів-Поляна, 2013, 2015, 2016);
- «Досвід розробки та застосування приладо-технологічних САПР в мікроелектроніці» (Львів-Поляна, 2013);

*2. міжнародні науково-практичні конференції:*

- «Інтернет-Освіта-Наука» (Вінниця, 2014);
- «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій» (Запоріжжя, 2012, 2014);

*3. міжнародні симпозиуми:*

- «Embedded systems and trends in teaching engineering» (Нітра, Словаччина, 2016);
- «Ambient intelligence and embedded systems» (Вааса, Фінляндія, 2017);

*4. науково-практична конференція:*

- «Тиждень науки ЗНТУ» (Запоріжжя, 2016).

**Акти впровадження результатів дисертаційної роботи**

ЗАТВЕРДЖУЮ

Директор ТОВ ПКФ МОТОР

посада керівника, назва організації (підприємства)

  
підпис

О.Г. Фісюк  
ПІБ

« 13 » березня 2018 р.



## АКТ

### впровадження результатів дисертаційної роботи

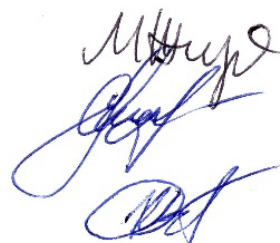
Комісія, у складі керівника ЕТН Хрунова М.С., інженера-проектувальника Колпакова Д.А. та інженера-проектувальника Стасілович Х.О., цим актом засвідчує, що тема дисертаційної роботи Гладкової Ольги Миколаївни «Автоматизація проектування вбудованих систем засобами віддаленої інженерії», виконаної у Запорізькому національному технічному університеті, властива нашому підприємству та її результати були впроваджені в практику інженерного проектування вбудованої системи керування комплексом електротехнічного обладнання.

Розроблена Гладковою О.М. рекомендаційна система вибору апаратно-програмних платформ на основі моделі формування вимог до вбудованих систем та методу роботи з ними, дозволяє підвищити ефективність процесу автоматизованого проектування, оскільки надає підтримку розробникові на етапах вибору та апробації проектних рішень виходячи з вимог до проектованої вбудованої системи.

Створена на основі технологій віддаленої інженерії рекомендаційна система повністю працездатна та протестована при розробці вбудованої системи керування комплексом електротехнічного обладнання. Використання інструменту надання рекомендацій є доцільним оскільки дозволяє скоротити час проектування за рахунок вибору апаратно-програмних платформ у відповідності з вимогами до проектованої вбудованої системи.

Голова комісії

Члени комісії



М.С. Хрунов

Х.О. Стасілович

Д.А. Колпаков

ЗАТВЕРДЖУЮ

Генеральний директор  
ТОВ «НВП ХАРТРОН-ЮКОМ»  
посада керівника, назва організації (підприємства)

підпис

Т.А. Коновальчук

ГПБ

« 22 » листопада 2018 р.



## АКТ

### впровадження результатів дисертаційної роботи

Комісія у складі головного конструктора Єфименка М.В., начальника групи Подмастер'єва С.В., начальника групи Бакуменка В.І. цим актом засвідчує, що тема дисертаційної роботи Гладкової Ольги Миколаївни «Автоматизація проектування вбудованих систем засобами віддаленої інженерії», виконаної у Запорізькому національному технічному університеті, властива нашому підприємству та її результати були впроваджені в процес автоматизованого проектування систем керування обертальним рухом КА «Січ-2М».

Розроблений Гладковою О.М. програмно-технічний комплекс автоматизованого проектування вбудованих систем, на основі створених методів, моделей та алгоритмів віддаленої інженерії, повністю працездатний та апробований при розробці систем керування обертальним рухом КА «Січ-2М». Використання програмно-технічного комплексу є доцільним, оскільки дозволяє підвищити рівень автоматизації проектувальних робіт на системному та функціонально-логічному рівнях, скоротити час проектування та витрати на доробку виробів за рахунок ефективного вибору апаратно-програмних платформ та швидкого прототипування на основі технологій віддаленої інженерії.

Голова комісії

М.В. Єфименко

Члени комісії

С.В. Подмастер'єв

В.І. Бакуменко

ЗАТВЕРДЖУЮ

Перший проректор Запорізького  
національного технічного  
університету, д.т.н., професор



В.Г.Прушківський

«19» квітня 2018 р.

АКТ

впровадження результатів дисертаційної роботи  
Гладкової Ольги Миколаївни

Комісія у складі завідувача кафедри програмних засобів д.т.н., професора Субботіна С.О., к.т.н., доцента Олійника А.О., к.т.н. доцента Федорончак Т.В. засвідчує цим актом, що результати дисертаційної роботи Гладкової О.М. впроваджені в лекційний курс та лабораторний практикум з дисциплін «Інженерія вбудованих систем», «Технології та системи віртуальної та віддаленої інженерії» за спеціальністю 121 Інженерія програмного забезпечення, а також «Сучасні CAD/CAM/CAE системи в проектуванні та виробництві наукоємної продукції» за спеціальністю 122 Комп'ютерні науки.

Вивчення запропонованих у дисертаційній роботі методик, моделей та алгоритмів дозволяє сформуванню у студентів знання та практичні навички щодо вибору апаратно-програмних платформ, а також застосування технологій віддаленої інженерії при автоматизованому проектуванні вбудованих систем.

Розроблені в рамках дисертаційної роботи Гладкової О.М. комплекси «Віддалена лабораторія проектування вбудованих систем» та «Автоматизована система управління рухомим об'єктом» використовуються як нова сучасна програмно-апаратна база при курсовому та дипломному проектуванні студентів за зазначеними спеціальностями.

Завідувач кафедри  
програмних засобів,  
д.т.н., професор

С.О.Субботін

Доцент кафедри  
програмних засобів  
к.т.н., доцент

А.О.Олійник

Доцент кафедри  
програмних засобів  
к.т.н., доцент

Т.В.Федорончак

**Свідоцтва про реєстрацію авторського права на твір  
(комп'ютерну програму) та патент на корисну модель**

УКРАЇНА



ДЕРЖАВНА СЛУЖБА

ВЛАСНОСТІ УКРАЇНИ

ІНТЕЛЕКТУАЛЬНОЇ

# СВІДОЦТВО

про реєстрацію авторського права на твір

№ 64253

Комп'ютерна програма "Віддалена лабораторія проектування вбудованих систем"

(вид, назва службового твору)

Автор(и) Пархоменко Анжеліка Володимирівна, Гладкова Ольга Миколаївна, Соколянський Олександр Володимирович

(повне ім'я, псевдонім (за наявності))

Авторські майнові права належать Запорізький національний технічний університет, вул. Жуковського, 64, м. Запоріжжя, 69063

(повне ім'я фізичної та/або повне офіційне найменування юридичної особи, адреса)

Дата реєстрації

26.02.2016



Голова Державної служби  
інтелектуальної  
власності України

В.о. Голови А.А. Малиш



УКРАЇНА



ДЕРЖАВНА СЛУЖБА

ВЛАСНОСТІ УКРАЇНИ

ІНТЕЛЕКТУАЛЬНОЇ

# СВІДОЦТВО

про реєстрацію авторського права на твір

№ 67404

Комп'ютерна програма "Автоматизована система управління рухомим об'єктом"

(вид, назва службового твору)

Автор(и) Пархоменко Анжеліка Володимирівна, Гладкова Ольга Миколаївна

(повне ім'я, псевдонім (за наявності))

Авторські майнові права належать Запорізький національний технічний університет, вул. Жуковського, 64, м. Запоріжжя, 69063

(повне ім'я фізичної та/або повне офіційне найменування юридичної особи, адреса)

22.08.2016

Дата реєстрації



Голова Державної служби  
інтелектуальної  
власності України  
В.о. Голови А.А. Малиш

УКРАЇНА



# ПАТЕНТ

НА КОРИСНУ МОДЕЛЬ

№ 123942

**СИСТЕМА АВТОМАТИЗОВАНОГО КЕРУВАННЯ ГРУПОЮ  
РУХОМИХ ОБ'ЄКТІВ**

Видано відповідно до Закону України "Про охорону прав на винаходи і корисні моделі".

Зареєстровано в Державному реєстрі патентів України на корисні моделі 12.03.2018.

Заступник міністра економічного розвитку і торгівлі України

М.І. Тітарчук





МІНІСТЕРСТВО  
ЕКОНОМІЧНОГО  
РОЗВИТКУ І ТОРГІВЛІ  
УКРАЇНИ

УКРАЇНА

(19) **UA** (11) **123942** (13) **U**  
(51) МПК  
**G05B 19/05** (2006.01)

## (12) ОПИС ДО ПАТЕНТУ НА КОРИСНУ МОДЕЛЬ

(21) Номер заявки: <b>u 2017 10387</b>	(72) Винахідник(и): <b>Пархоменко Анжеліка Володимирівна (UA), Гладкова Ольга Миколаївна (UA), Кравченко Олексій Павлович (UA)</b>
(22) Дата подання заявки: <b>27.10.2017</b>	(73) Власник(и): <b>ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ, вул. Жуковського, 64, м. Запоріжжя, 69063 (UA)</b>
(24) Дата, з якої є чинними права на корисну модель: <b>12.03.2018</b>	(74) Представник: <b>Висоцька Наталя Іванівна</b>
(46) Публікація відомостей про видачу патенту: <b>12.03.2018, Бюл.№ 5</b>	

## (54) СИСТЕМА АВТОМАТИЗОВАНОГО КЕРУВАННЯ ГРУПОЮ РУХОМИХ ОБ'ЄКТІВ

### (57) Реферат:

Система автоматизованого керування групою рухомих об'єктів складається з робочого органу, при цьому в робочому органі встановлений блок пам'яті, забезпечений відповідними командами блока зв'язку, з'єданого з пультом керування і через блок зв'язку з'єданого з пультом керування системою і блоком виводу інформації. Як робочий орган виступає один або декілька рухомих об'єктів з довільним шляхом руху, на яких додатково встановлені панель індикації та блок відеопередачі, який використовує частоту 900 МГц, містить відеокамеру, відеопередавач та з'єднаний з відеоокулярами оператора, як блок пам'яті використовується мікроконтролерний блок на основі програмованого мікроконтролера; блок зв'язку використовує частоту 433 МГц, а для живлення робочого органу застосовано акумуляторні батареї 9 В; до пульта керування додатково під'єднано панель керування та джойстики операторів, а як програмний мікропроцесор застосовано модуль керування на основі другого програмованого мікроконтролера.

UA 123942 U