УДК 621.391
DOI https://doi.org/10.32782/2663-5941/2025.4.1/08

Катін П.Ю.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Вишневий С.В.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

АРХІТЕКТУРА ПРОГРАМНОЇ СКЛАДОВОЇ РАДІОТЕХНІЧНИХ СИСТЕМ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ НА БАЗІ ПАТЕРНУ «СПОСТЕРІГАЧ»

У статті досліджено архітектуру програмної складової радіотехнічних систем, що дозволяють поєднати різні технології програмування у різних компонентах. Основою апаратної архітектури радіотехнічної системи є головний контролер на основі мікрокомп'ютера з повноцінною ОС або аналогічна потужна мікропроцесорна обчислювальна система з елементами штучного інтелекту. Вона керує іншими складовими радіотехнічної системи, що реалізовані на більш простих мікроконтролерах і відграють роль програмно-апаратних інтерфейсів між головним контролером та датчиками (виконавчими механізмами, радіомодулями, тощо).

У ході дослідження розглянуті різні варіанти архітектур і типових патернів, що можуть бути придатними для вирішення завдання побудови архітектури. У результаті дослідження визначено, що для побудови архітектури доцільно використовувати типовий патерн «Спостерігач». При цьому потрібна модифікація даного патерну з урахуванням вимог поєднання різних програмних технологій.

Визначено, що модифікація «Спостерігач» має бути реалізована за рахунок додавання нових програмних елементів. Він додатково включає програмні складові для розподілу завдання між головним контролером радіотехнічної системи з елементи штучного інтелекту і другою частиною на базі простих мікроконтролерів. Друга частина забезпечує роботу виконавчих механізмів і датчиків з використанням простого мікроконтролера з програмою на основі патерну «Стан».

На відміну від відомих рішень патерну «Спостерігач», отримана архітектура дає можливість поєднати відносно прості модулі на базі мікроконтролерів і складні рішення на основі мікрокомп'ютерів з використанням технологій повноцінної операційної системи.

Отримані результати можуть бути використані для вдосконалення програмної складової радіотехнічних, радіокерованих та інших вбудованих систем на основі штучного інтелекту різного рівня складності та призначення.

Ключові слова: *системи радіокерування, програмна архітектура, об'єктноорієнтована парадигма програмування, процедурна парадигма програмування, штучний інтелект, архітектура ARM Cortex-M, робототехнічні системи, патерн спостерігач, радіотехнічна система, операційна система реального часу.*

Постановка завдання і обмеження. На теперішній час одним із актуальних напрямків дослідження є розробка програмної архітектури радіотехнічних систем (РТС), що можуть потенційно працювати з елементами штучного інтелекту (ШІ). До радіотехнічних систем можна віднести: системи радіокерування, сучасні радіолокаційні станції, радіокеровані повітряні і наземні платформи [1], робототехнічні системи, що працюють на основі радіоканалу, тощо. З'явилися нові роботи у цьому напрямку, наприклад вбудовані системи з елементами радіокерування і елементами ШІ [1–4].

Частина складних РТС може бути представлена у вигляді сукупності програмованих радіотехнічних модулів (ПРМ). Причому базовим контролером радіотехнічної системи (БКРТС) є рішення на основі відносно потужного мікрокомп'ютера,

що реалізує основну логіку радіокерованих систем (РКС) та інших РТС. Допоміжними модулями в цій системі є програмно-апаратні модулі на базі мікроконтролера (МК). В РКС, наприклад, вони забезпечують роботу радіоканалу і виконавчих механізмів. Початковий підхід до узагальнення архітектури, без радіоканалу, показаний у [5]. Програмовані модулі об'єднуються у загальну програмно-апаратну систему із певною програмною архітектурою. Основою архітектури є потужний мінікомп'ютер, якій і являє собою БКРТС, технології програмування якого можуть відрізнятися від технологій ПРМ. Таким чином важливим завданням є розробка програмної архітектури, бажано у рамках відомого шаблону (патерну), що дозволяє забезпечити взаємодію БКРТС з елементами штучного інтелекту і ПРМ на основі відносно простих МК.

Відомо багато практичних варіантів реалізації взаємодії у рамках РКС і радіотехнічних систем. Тому, не змінюючи загального характеру досліджень, **введемо обмеження** на програмну і апаратну складову РТС з елементами ШІ, що містить програмовані модулі і БКРТС.

Апаратна архітектура РТС складається із одного БКРТС, що побудований на базі мікрокомп'ютера (або іншого потужного обчислювального рішення) і працює під управлінням повноцінної операційної системи (ОС). Виконавчі механізми, датчики, радіопередавачі та радіоприймачі підключені до БКРТС за допомогою МК через послідовні інтерфейси (I^2C, SPI, USART, тощо) або через порти вводу-виводу загального призначення (ПВВЗП). Фактично ці підсистеми і є ПРМ в рамках загальної архітектури. Апаратною основою таких ПРМ у більшості випадків на теперішній час є МК архітектури ARM Cortex-M.

Таким чином виникає потреба розробити програмну архітектури РТС з програмованими модулями та БКРТС з елементами ШІ. Виконавчі механізми та датчики працюють через ПРМ у рамках модифікованого патерну «Стан» з можливістю додавання нових станів під час роботи РТС без її зупинки. Операційні системи реального часу (ОСРЧ (RTOS)) не використовуються. Програмні технології БКРТС та ПРМ відрізняються і можуть працювати у різних парадигмах програмування: наприклад у ООП парадигмі (для БКРТС з доступом до всіх технологій повноцінних ОС); для рішень на основі МК (для ПРМ з обмеженим використанням технологій, без ОС).

Аналіз останніх досліджень і публікацій. Показав, що роботи у цьому напрямку провадяться. Наприклад у роботах [1–4] наведений загальний опис вбудованих систем, що містять елементи ШІ і можуть стать основою для побудови загальної архітектури, в той же час загальна архітектура для конкретних рішень РТС в цих роботах не розкрита.

У [5] показана спроба узагальнити архітектуру вбудованих систем, при цьому не ураховується вплив штучного інтелекту на програмну архітектуру РТС у цілому і відсутній радіоканал.

Використання патерну «Стан» для малопотужних МК описано у [6], але використання цих результатів в даному випадку не доцільно, оскільки найбільшою популярність для аналогічних рішень користуються МК архітектури ARM Cortex-M.

Основою програмної архітектури РТС з елементами штучного інтелекту можуть бути [7, 8]. У роботі [7] описаний базовий патерн «Стан» для архітектури мікроконтролера ARM Cortex-M. Сформульована методика тестування ефективності рішення. В той же час не урахований вплив ШІ на загальну архітектуру програмного рішення.

Робота [8] розкриває основи дослідження патерну «Стан» в контексті можливості додавання нових станів під час роботи системи з елементами ШІ. Але у цій роботі розглядається рішення програми для МК на рівні ПВВЗП (GPIO), без узагальнення результату в рамках РТС та РКС.

Архітектура патерну «Спостерігач» у контексті практичного застосування описана у [9–11]. В цих роботах не ураховуються потреби узгодити різні технології програмування (БКРТС з повноцінною ОС та на основі МК у варіанті нескінченного циклу опитування або у системі переривань [5].

Для реалізації загальної архітектури доцільно розглянути рішення у варіанті патерну «Master-Slaves» [12]. Ця архітектура дозволяє керувати процесами, розподіляти завдання для ПРМ і отримати результати вимірювання від датчиків ПРМ, що виконують роль Slaves. Як показує практика, така архітектура використовується у наступних випадках:

– для управління складними і просторово розподіленими базами даних;

– для управління паралельними обчисленнями у високопродуктивних кластерних системах;

– в менеджменті складних мережевих інфраструктур для підвищення якості трафіку і удосконалення сервісу, тощо. Тобто архітектура на основі патерну «Master-Slaves» вимагає щоб і БКРТС

і контролери ПРМ були виконані з використанням повноцінної ОС, що суперечить обмеженням.

Постановка завдання. Отже, результати аналізу свідчать про достатню кількість робіт у цьому напрямку, при цьому опис базового аналога архітектури РТС для виконання поставленого завдання немає.

Таким чином метою статті є розробка програмної архітектури для складної РТС з потенційно можливим використанням штучного інтелекту. Базовий контролер має працювати з відносно дешевим периферійним рішенням ПРМ на основі патерну Стан. Програмні технології БКРТС і ПРМ у рамках РТС відрізняються за видом і рівнем складності.

Виклад основного матеріалу. З урахуванням постановки завдання і введених обмеженнях була проведена робота щодо оцінки використання найбільш прийнятних типових шаблонів (патернів) у архітектурі з присутністю головного контролера і підпорядкованих модулів.

На першому етапі розглянемо патерн «Master-Slave» як основу для архітектури РТС. При цьому буде ураховуватися наявність ШІ. В цілому патерн відповідає завданню. Він містить керівничий контролер (Master Node) і один (або декілька) підпорядкованих контролерів (Slave Node(s)). Для взаємодії цих компонентів вимагається складний комунікаційний протокол і механізм розподілу завдань. Це дуже важко виконати для ПРМ у рамках послідовних інтерфейсів (I^2C, SPI, USART, тощо) або через ПВВЗП. Існують більш прості рішення у варіанті «Master-Slave» для МК. Але вони вимагають використання RTOS, процес відбувається через багатопотоковість. В більш простих рішеннях для МК потрібно утворення нескінченних циклів опитування у БКРТС і у контролерах ПРМ. Таке рішення є прийнятним для ПРМ. В той же час вони є застарілим для програмної частини БКРТС. Такий програмний продукт важко супроводжується, модифікується і є неприйнятним для сучасних рішень.

Найбільш придатним варіантом реалізації архітектури для РТС на основі БКРТС і модулем ПРМ є патерн «Спостерігач» (Observer), оскільки він забезпечує ієрархію взаємодії БКРТС і ПРМ і розподіл завдань. Проте треба ураховувати, що класичне рішення «Спостерігач» (Observer) не можливо, з урахуванням введених обмежень тому, що класичний патерн «Спостерігач» (Observer) працює у рамках одної програмної технології. Це демонструє рисунок з елементами UML (діаграми класів), що показана на рис. 1. Рисунок демонструє загальну архітектуру патерну «Спостерігач» (Observer) в контексті застосування до РТС. Вона, відповідно завданням і обмеженню, розділяється на дві базових частини.

Перша базова частина – це програмний модуль БКРТС з елементами штучного інтелекту, що виділений на рис. 1 окремою лінією. Він являє собою частину реалізації «Спостерігача» (Observer), а саме – елемент «Видавець».

Другу частину патерну «Спостерігач», а саме елемент «Підписник», реалізує ПРМ на базі МК, їх може бути декілька. За умовами вони містять програму на основі патерну «Стан». Про-

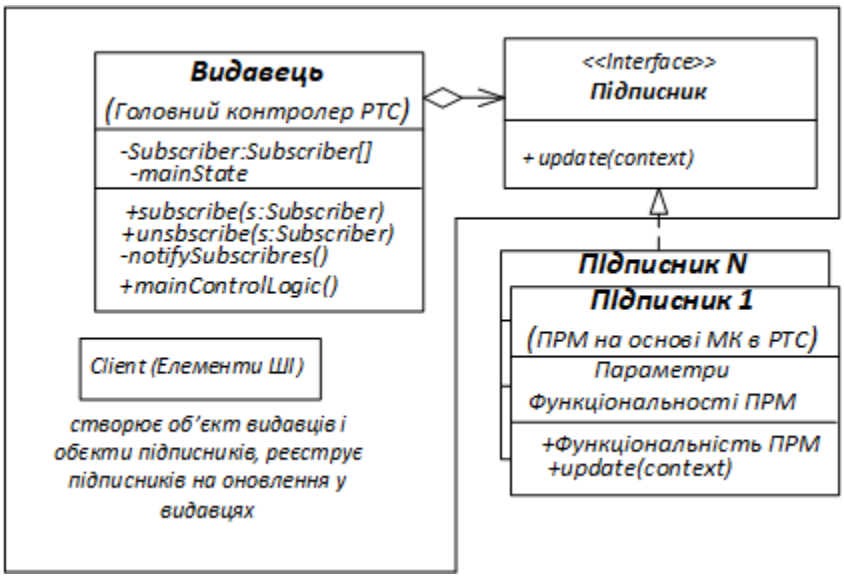

Рис. 1. Основа програмної архітектури у патерні «Спостерігач»

грами ПРМ відрізняється технологічно від програми БКРТС. Цей модуль, відповідно завданню, зв'язаний з «Видавцем» через послідовні інтерфейси (I^2C, SPI, USART, тощо) або через ПВВЗП (GPIO). Тобто просте наслідування, як показано на рис. 1, у програмній архітектурі для РТС, у варіанті БКРТС – ПРМ не можливо. Потрібно допрацювання архітектурного рішення.

Така програмна архітектура РТС та РКС з складовими штучного інтелекту показана на рис. 2, на схемі, що містить елементи UML (діграми класів).

У цьому варіанті до базового патерну «Спостерігач» у БКРТС внесені суттєві зміни. До програмної частини БКРТС додано два нових елемента. Перший елемент призначений для кодування повідомлень «Видавця», оскільки ПРМ зв'язаний з БКРТС через послідовний інтерфейс МК. Крім того до операційної системи БКРТС додаються драйвери для взаємодії мікрокомп'ютера і ПРМ.

Таким чином БКРТС є не просто елементом патерну «Спостерігач», він є активним керівничим блоком, що може аналізувати і виконувати координацію роботи ПРМ (підписника). Це стає можливим оскільки «Видавець» працює у сполученні з «Підписником» в рамках патерну «Спостерігач». Програмна частина ПРМ побудована на основі патерну стан (State). Таким чином в даній архітектурі роль «Підписників» відіграють ПРМ радіотехнічної системи. Звичайно, що взаємну роботу БКРТС і ПРМ реалізують програмний модуль для кодування сповіщень і драйвери послідовних інтерфейсів БКРТС.

Ролеву модель програмної архітектури забезпечують елементи штучного інтелекту БКРТС. Взаємодія між елементами архітектури «Видавець» і «Підписник» (між БКРТС і ПРМ) може бути синхронною (варіант роботи патерну «Стан» ПРМ у нескінченному циклі опитування [5]). Доступний також асинхронний режим. Це варіант роботи патерну «Стан» ПРМ у режимі виконання переривань МК [5].

Апаратна складова, що реалізує ПРМ, розробляється на основі розповсюджених дешевих МК, у багатьох випадках застосовуються МК на базі архітектури Advanced RISC Machine (ARM) Cortex-M [7, 8] або інші рішення аналогічної потужності. Елементи ПРМ на базі цього класу мікроконтролерів (Cortex-M) не можуть повною мірою реалізовувати програмі елементи ШІ. Уза-

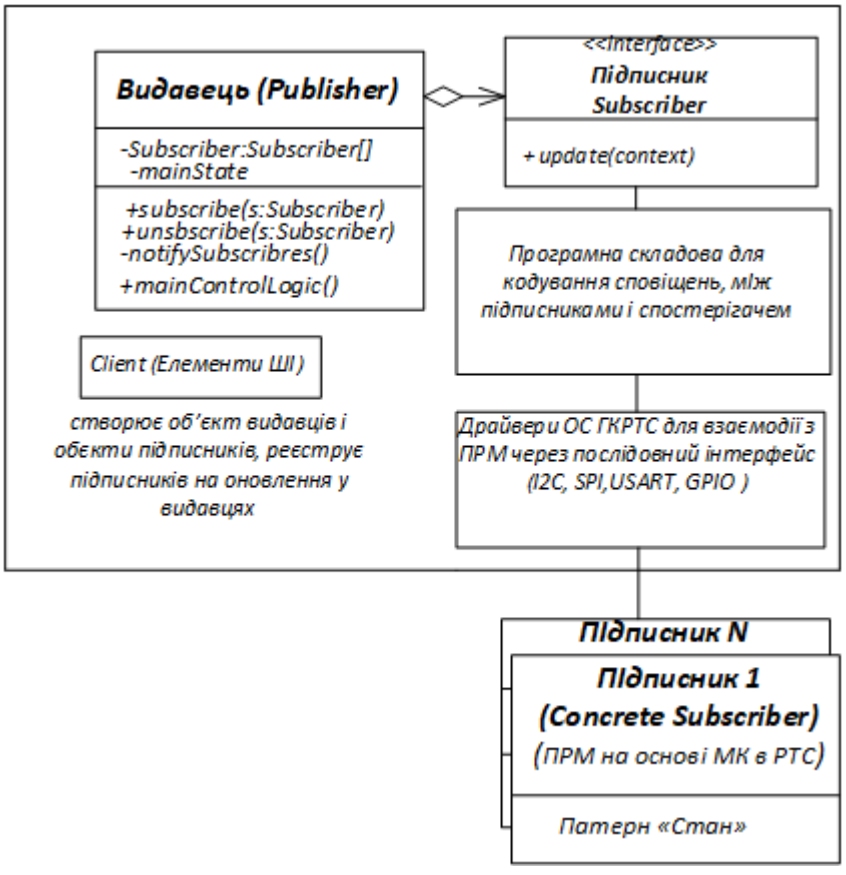

Рис. 2. Архітектура на базі патерну «Спостерігач»

гальнена схема програмно-апаратної інфраструктури РТС та РКС показана на рис. 3. Основна операційна система БКРТС реалізує елементи штучного інтелекту, має впроваджувати процес самонавчання, самооптимізації і реалізовувати самовідновлення. А модуль ПРМ виконує функції взаємодії із датчиками і виконавчими механізмами, надає на спостерігач інформацію для обробки ШІ.

Модулі ПРМ працюють у відомій версії шаблону «Стан» [7], що реалізує математичну модель КА. Основним недоліком цей версії патерну «Стан» є потреба додавання нових станів на етапі розробки програмного забезпечення, у тому числі необхідно перепрограмування МК. Тобто після того, як був доданий новий стан, потрібно переписувати програму мікроконтролера. Використання такого підходу у даній архітектурі є суттєвим недоліком. Сама ідея самонавчання ШІ передбачає, що число станів у «Підписників» може динамічно змінюватися, відповідно програмна складова ПРМ має додавати цей стан у процесі роботи, без зупинення.

Від цього недоліку вільне рішення, що оприлюднено у роботі [8]. Таким чином БКРТС (рис. 3) використовує інформацію від програмної складової ПРМ. Обробляє результати, виконує функції ШІ, в разі необхідності передає команду на створення додаткового стану у програмі ПРМ на основі МК.

Для перевірки прийнятності результаті у практичному застосування у рамках даної архітектури була розроблена програма мовою С++ для мікроконтролера архітектури ARM Cortex-M на STM32F401. Під час аналізу оцінювалися часові показники програмного рішення з використанням можливостей MDK Keil. Для визначення часу між перемиканнями використана методика, що описана у [7, 8]. Результати показали прийнятність отриманого рішення.

Висновки. Із урахуванням поставлених умов і обмежень, була розроблена архітектура програмної складової для радіотехнічних системи з елементами штучного інтелекту на основі патерну «Спостерігач». Отримані архітектурні рішення можуть застосовуватися не тільки у РТС і системах радіокерування. Вони можуть бути використані у інших вбудованих інтелектуальних системах на базі ШІ.

Відмінністю архітектурного рішення від класичного патерну «Спостерігач» є об'єднання двох технологій програмування:

– програмних технологій для БКРТС, що доступні на повноцінних ОС;

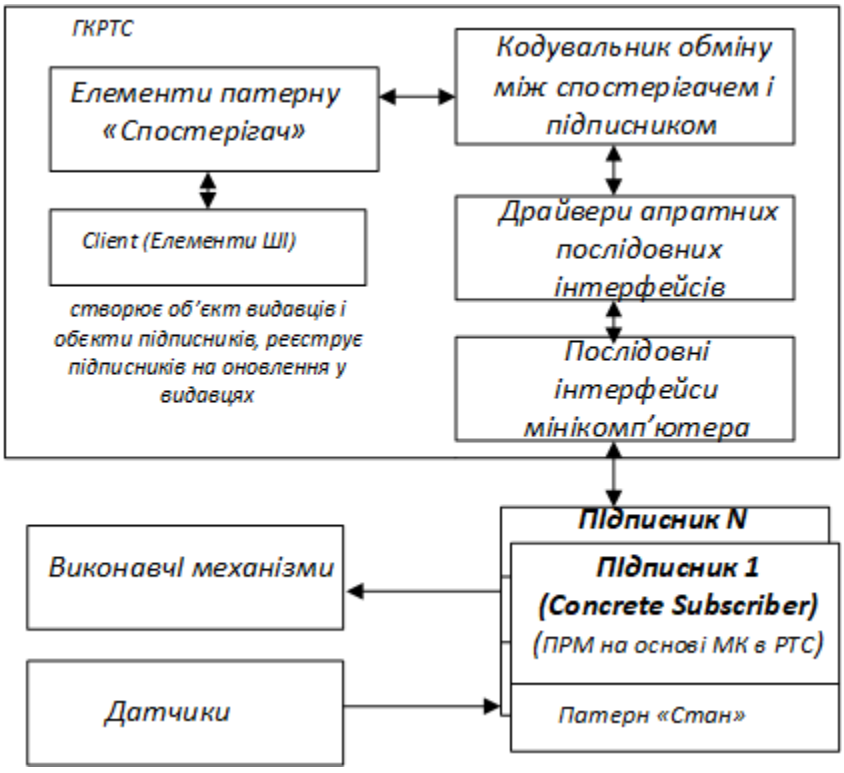

Рис. 3. Узагальнена схема програмно-апаратних складових

– технології програмування МК архітектури (ARM) Cortex-M у патерні «Стан» з можливістю динамічного додавання нових станів [8], у ООП парадигмі і процедурному варіанті, в синхронному і асинхронному режимах.

Перевагами архітектури є чітка модульність конструкції РТС, можливість розподілу завдань між розробниками класичного програмного забезпечення у ООП парадигмі на борту БКРТС і розробниками програм для мікроконтролера, що можуть працювати у різних парадигмах. Висока надійність, легкість супроводження і налагодження програми БКРТС і програми ПРМ.

До потенційних недоліків можна віднести можливе перевантаження БКРТС, затримки під час обміну даними. В той же час з урахуванням умов і введених обмежень, ці недоліки не будуть впливати на загальну роботу системи.

У якості нового напрямку досліджень може бути розробка архітектури з використанням РТОС і наявністю багатопотоковості у ПРМ.

Список літератури:

1. Альперт С. І., Альперт М. І., Катін П. Ю., Літвінова Н. О. Програмно-апаратна інфраструктура наземної автономної платформи з елементами штучного інтелекту. *Математичні машини і системи*. 2021. № 1. С. 24–31. doi: 10.34121/1028-9763-2021-1-24-31.

2. Santos P. H. O., Soares G. L., Machado-Coelho T. M., Godinho de Oliveira B. A., Ekel P., Freitas F., Martins C. A. Multi-objective genetic algorithm implemented on a STM32F407 microcontroller. *2018 IEEE Congress on Evolutionary Computation* (*CEC*) : proceedings. Rio de Janeiro, 8–13 July 2018. Rio de Janeiro, 2018 P. 486–492.

3. Boutekkouk F., Mahalaine R., Zina M., Saliha L., Djouani R., Djalila B. Intelligent embedded software: new perspectives and challenges. *Intelligent System*. 2018. P. 3–22.

4. Lu J., Zhou H. Implementation of artificial intelligence algorithm in embedded system. *Journal of Physics: Conference Series*. 2021. P. 1–11. doi: 10.1088/1742-6596/1757/1/012015

5. Chmelov V. O., Katin P. Y., Shemaev V. M. Development of typical «State» software patterns for Cortex-M microcontrollers in real time. *Eastern-European Journal of Enterprise Technologies*. 2020. Vol. 3(9 (105)). P. 29–38.

6. Katin P. Y. Development of variant of software architecture implementation for low-power general purpose microcontrollers by finite state machines. *Eureka: Physics and Engineering*. 2017. No. 3. P. 49–55.

7. Катін П. Ю., Похиленко О. А. Шаблони типу «Стан» для створення інфраструктури системного програмного забезпечення мікроконтролерів архітектури Cortex-M у режимі реального часу для вбудованих систем. *Електронне моделювання*. 2021. Т. 43, № 2. С. 51–67. URL: https://emodel.org.ua/uk/archive-ukr/2021/43-2-u/c-51-67 (дата звернення: 25.05.2025).

8. Похиленко О., Катін П. Патерн «СТАН» для вбудованих систем з можливістю динамічного створення станів. *Технічні науки та технології*. 2021. № 1(23). С. 118–127. URL: http://tst.stu.cn.ua/article/view/233571. (дата звернення: 25.05.2025).

9. Understanding the Observer Pattern – Use Cases and Implementations Explained. URL:https://surl.li/krrnig (дата звернення: 31.05.2025).

10. Kulkarni N., Bansal S. Observer Design Pattern Applied on Real-Life Store Use Case. *Journal of Artificial Intelligence & Cloud Computing*. 2022. Vol. 1(2). P. 1–6.

11. Observer Design Pattern. URL: https://www.geeksforgeeks.org/system-design/observer-pattern-set-1-introduction/ (дата звернення: 31.05.2025).

12. The Master-Slave Design Pattern. URL: https://www.cs.sjsu.edu/~pearce/oom/patterns/behavioral/masterslave.htm (дата звернення: 31.05.2025).

Katin P.Yu., Vyshnevyi S.V. SOFTWARE ARCHITECTURE OF THE RADIO SYSTEMS WITH ARTIFICIAL INTELLIGENCE ELEMENTS BASED ON THE "OBSERVER" PATTERN

The article explores the architecture of the software component of radio engineering systems, which enables the integration of various programming technologies across different components. At the core of the hardware architecture is a main controller based on a microcomputer with a full-fledged operating system or a similarly powerful microprocessor-based computing system equipped with artificial intelligence elements. This controller manages other components of the radio engineering system, which are implemented on simpler microcontrollers and serve as software-hardware interfaces between the main controller and sensors (actuators, radio modules, etc.).

In the course of the research, various architectural options and common design patterns were examined for their suitability in solving the problem of system architecture design. As a result, it was determined that

the 'Observer' design pattern is appropriate for this purpose. However, this pattern requires modification to accommodate the integration of different programming technologies.

It was found that the modified 'Observer' pattern should be implemented by adding new software elements. These elements include components that help distribute tasks. The distribution happens between the main controller of the radio engineering system with artificial intelligence elements and a secondary unit based on simple microcontrollers The latter handles actuators and sensors using a simple microcontroller programmed according to the 'State' pattern.

Unlike known implementations of the 'Observer' pattern, the proposed architecture enables the integration of relatively simple microcontroller-based modules with complex solutions built on microcomputers that utilize full-fledged operating system technologies.

The obtained results can be applied to improve the software components of radio engineering, radio-controlled, and other embedded systems based on artificial intelligence of varying complexity and purpose.

Key words: *radio control systems, software architecture, object-oriented programming paradigm (OOP), procedural programming paradigm, artificial intelligence (AI), ARM Cortex-M architecture, robotic systems, observer pattern, radio engineering system, real-time operating system (RTOS).*

Дата надходження статті: 02.07.2025
Дата прийняття статті: 07.07.2025
Опубліковано: 27.10.2025