

УДК 004.5:004.85

С.І. АЛЬПЕРТ*, М.І. АЛЬПЕРТ**, П.Ю. КАТІН**, Н.О. ЛІТВИНОВА**

ПРОГРАМНО-АПАРАТНА ІНФРАСТРУКТУРА НАЗЕМНОЇ АВТОНОМНОЇ ПЛАТФОРМИ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ

*Науковий Центр аерокосмічних досліджень Землі ІГН НАН України, м. Київ, Україна

**НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

Анотація. За рахунок сучасних мікрокомп'ютерів і платформ на базі мікропроцесорів, а саме: Raspberry Pi, Orange Pi, Nano Pi, Rock Pi, Banana Pi, Asus Tinker Board, розробка прототипів вбудованих систем можлива в «конструкторському» режимі. Програмна частина реалізується на базі операційних систем і на стандартних технологіях на основі відомих мов програмування. Прикладом можуть бути мови програмування C/C++, Python, C#, Java тощо. У цьому випадку канал управління для вбудованої системи можна реалізувати через веб-сервіс, відокремлений каналом зв'язку, або цей пристрій може керуватися автономно. Важливо розуміти, що створення вбудованої системи на стандартній платформі виходить набагато дорожче, ніж купити такий же по функціоналу готовий пристрій серійного виробництва. Тому платформи типу Raspberry Pi має сенс використовувати в основному для одиничних штучних пристроїв. Якщо потрібно зібрати проєкт вбудованих систем і є проблема з вибором апаратної платформи для клієнтської сторони, то на даний час існує широкий вибір плат і рішень для побудови ефективної і недорогої системи з використанням готових модулів. Кількість плат розширення і різних датчиків, відеокамер, підключення до інтернету через Ethernet, Wi-Fi і Bluetooth надає широкий спектр можливостей для побудови практично будь-якого рішення на цій компонентній базі. Основу можна зробити в рамках невеликого бюджету, з мінімальними витратами часу, з використанням великих блоків і готових бібліотек для програмування вбудованих систем. У даній статті подано результати дослідно-конструкторської роботи з розробки програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту, за фактичним результатом досліджень побудовано діаграму розгортання та діаграму компонентів розробленої інфраструктури.

Ключові слова: мікропроцесори, діаграма розгортання, діаграма компонентів, автономна рухома платформа, оброблення зображення, .Net Core, Angular, операційна система Raspberry Pi Desktop, вбудовані системи, мікрокомп'ютер, штучний інтелект.

Abstract. Due to modern microcomputers and platforms based on microprocessors such as, for example, Raspberry Pi, Orange Pi, Nano Pi, Rock Pi, Banana Pi, Asus Tinker Board – the development of prototypes of embedded systems is possible in a «design» mode. The software part is implemented on the basis of operating systems and standard technologies based on well-known programming languages such as C/C++, Python, C#, Java, etc. In such case the control channel for the embedded system can be either implemented via a web service separated by a communication channel or controlled independently. It is important to understand that creating an embedded system on a standard platform is much more expensive than buying a ready-made mass-produced device with the same functionality. Therefore, it makes sense to use platforms like the Raspberry Pi mainly for individual artificial devices. If it is necessary to build a project of embedded systems and there is a problem with choosing a hardware platform for the client side, then currently there is a wide range of boards and solutions for building an efficient and inexpensive system using ready-made modules. The number of expansion cards and various sensors, video cameras, internet connection via Ethernet, Wi-Fi and Bluetooth provides a wide range of opportunities for building almost any solution based on this component base. The foundation can be made within a small budget, with minimal time spent, using large blocks and ready-made libraries for programming embedded systems. This article presents the results of research and development work on the creation of a software and hardware infrastructure of a terrestrial platform with the elements of

artificial intelligence. Based on the actual results of the research, a deployment diagram and a component diagram of such an infrastructure have been constructed.

Keywords: *microprocessors, deployment diagram, component diagram, autonomous mobile platform, image processing, .NET Core, Angular, Raspberry Pi Desktop operating system, embedded systems, microcomputer, artificial intelligence.*

DOI: 10.34121/1028-9763-2021-1-24-31

1. Вступ

Мікрокомп'ютер – це невеликий, відносно недорогий комп'ютер із мікропроцесором використовується як центральний процесор. Він включає в себе мікропроцесор, пам'ять та мінімальну схему вводу-виводу(I/O), встановлену на одній друкованій платі. Перші мікрокомп'ютери стали популярними в 1970–1980 рр. з появою все більш потужних мікропроцесорів. Попередники цих комп'ютерів, мейнфрейми і мінікомп'ютери, були порівняно набагато більші і дорожчі (хоча сучасні мейнфрейми, такі як машини IBM System Z, використовують один або кілька користувальницьких мікропроцесорів як свої процесори). Багато мікрокомп'ютерів, якщо коли вони оснащені клавіатурою і екраном для введення і виведення, також є персональними комп'ютерами в загальному сенсі.

На робочому місці мікрокомп'ютери використовуються для програмних додатків, включаючи обробку даних і текстів, електронні таблиці даних, професійні презентаційні та графічні програми, системи зв'язку та управління базами даних. Вони використовуються в бізнесі для таких завдань, як бухгалтерія, інвентаризація і зв'язок; в медичних установах для запису і відкликання даних про пацієнтів, управління планами охорони здоров'я, повного розкладу і обробки даних; у фінансових установах для запису транзакцій, відстеження виставлення рахунків, підготовки фінансових звітів і платіжних відомостей, а також навчальних пристроїв та ін. [1–2].

Однак мікрокомп'ютери можуть використовуватися для вирішення тих же завдань у програмних додатках вбудованих систем, що і мікропроцесори. Деякі вбудовані системи, такі як розумні телевізори, холодильники та інші прилади, іноді називають мікрокомп'ютерами. Сучасні мікрокомп'ютери дають можливість побудувати систему зі штучним інтелектом. Багато уваги й аналізу приділено інтерпретації природи свідомості на основі сучасних уявлень про інформаційну складову реальності та її значення для моделювання й створення штучного інтелекту. У курсі лекцій розглянуто математичну складову штучного інтелекту і особливості деяких інтелектуальних задач. На жаль, у статтях [3–5] відсутнє будь-яке інженерне рішення.

Метою даної статті є розробка програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту, в основі якого лежить мікрокомп'ютер. Особливістю даного рішення є те, що система використовує фреймову мережу для автоматизації управління і додатково передає зображення на екран веб-браузера. На практиці активно запроваджуються прикладні й інженерні рішення.

На основі мікрокомп'ютера (ПМК) формулюються загальноприйняті в ІТ-описі елементи архітектури програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту. Під час проведення дослідницької роботи була використана POSIX-операційна система, що дає можливість узагальнити рішення на інші операційні системи (ОС).

2. Постановка проблеми у загальному вигляді

Спалах коронавірусу обмежив бізнес для багатьох компаній. Роботизовані служби доставки можуть обіцяти безконтактну доставку, дуже затребувану послугу в рамках соціального дистанціювання. У той час, як автономні роботи доставки вже використовувалися в деяких міських районах, аеропортах, університетах, готелях і великих

корпоративних кампусах до пандемії, попит на них «зростає експоненціально» [6]. На жаль, існуючі прототипи є досить дорогими і знаходяться на стадії раннього тестування. Для формального опису програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту використовують математичне моделювання. Вартість потужних мікропроцесорів, наприклад, Cortex-A, поступово зменшується до вартості мікропроцесорів малої і середньої потужності. Практичним прикладом є системи з потужним мікропроцесором SoC Broadcom BCM2711 Cortex-A72, плата мікрокомп'ютера (ПМК) Raspberry Pi 4 Model B 4GB й інші аналогічні ПМК. Для вирішення задачі програмно-апаратної інфраструктури наземної автономної платформи з елементами штучного інтелекту було вирішено використовувати Raspberry Pi Desktop OS. Вона має офіційну підтримку розробника. Також є можливість використовувати інші безкоштовні і портовані ОС: Ubuntu MATE, LibreELEC, RISC OS, DietPi, Kali Linux [7]. Отже, у разі використання мікропроцесорів такого типу для побудови програмно-апаратної інфраструктури наземної автономної платформи з елементами штучного інтелекту кінцева вартість розроблення може бути значно меншою за вартість аналогічного контролера, в основі якого лежать мікропроцесори малої і середньої потужностей [1–2].

3. Аналіз вже створених рішень і публікацій

Проаналізувавши літературу, пов'язану із прогресом високопотужних контролерів для застосування у промисловому виробництві, можна зауважити, що реалізація систем зі штучним інтелектом на практиці є можливою на основі ПМК з відносно невисокою вартістю. Даний аналіз підтверджує, що основою контролера програмно-апаратної інфраструктури наземної автономної платформи з елементами штучного інтелекту може бути плата Raspberry Pi 4 або її аналоги [1, 8–9]. Плата працює під керуванням ОС Raspberry Pi Desktop. Далі надаються технічні характеристики ПМК на базі Raspberry Pi 4 Model B, а саме:

- процесор 64-бітний SoC Broadcom BCM2711 чотириядерний ARMv8 Cortex-A72;
- графічний процесор VideoCore VI;
- оперативна пам'ять 4GB SDRAM LPDDR4;
- підтримка карт пам'яті MicroSD;
- 10/100/1000 Мбіт Gigabit Ethernet (контролер Broadcom BCM54213PE);
- 2,4 ГГц і 5 ГГц IEEE 802.11.b/g/n/ac WI-FI і Bluetooth 5.0 Low Energy (BLE), які забезпечуються мікросхемою Cypress CYW43455;
- 1 x CSI-2 для підключення камери по інтерфейсу MIPI;
- 2 x micro-HDMI;
- 1 x DSI (Display Serial Interface) для підключення штатного дисплею;
- 1 x композитний відеовихід (CVBS відео, PAL і NTSC) 3,5 мм роз'єм;
- 40 портів вводу-виведення загального призначення (GPIO), UART (Serial), I²C/TWI, SPI з селекторами між двома пристроями; піни живлення: 3,3 В, 5 В і земля;
- живлення 5 В, 3,0 А через порт USB type-C або GPIO; Power over Ethernet (PoE) через окремий PoE HAT (окремі 4 піни);
- розміри: 85,6 мм x 56,5 мм x 17 мм;
- вага: 45 г.

Інші ПМК цього класу мають аналогічні характеристики, потужність і функціональність. Середня вартість складає \$75 на момент написання статті, а наявність вільно розповсюджуваної ОС “Raspberry Pi Desktop” дозволяє встановити різні середовища розробки ПЗ. Дана ОС є сумісною з технологіями Python і має велику кількість допоміжних, безкоштовних, надійних бібліотек, щоб виконати поставлене завдання. Також ці бібліотеки надають можливість контролювання програмно-апаратної інфраструктури наземної автономної платформи з елементами штучного інтелекту.

Розглянемо деякі аналоги, в основі яких лежать мікропроцесори. У статті [12] розглянуто 8-бітний мікропроцесор STM8S903F3. Він підходить для управління сервоприводами і двигуном. Але 8-бітні мікропроцесори не можуть вирішити завдання штучного інтелекту, так як не вистачає ресурсів на платі.

У статті [13] використовується мікрокомп'ютер і реалізовано контроль рухомої платформи через відео. При розробці не було втілено широкої інфраструктури для штучного інтелекту.

У статті [14] реалізовано апаратну інфраструктуру і скінченний автомат, але штучний інтелект (ШІ) не втілено при створенні скінченного автомата.

У статтях [16–17] використано об'єктно-орієнтований підхід у застосуванні скінченних автоматів. Є правильно задана ієрархія при побудові програми, але не реалізовано апаратну частину і відсутній штучний інтелект.

4. Аналіз структури прототипу

Над прототипом програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту (ПАІН) було проведено дослідно-конструкторські роботи апаратної інфраструктури. В перспективі вона може обробляти і збирати інформацію у ході доставки речей користувачу. Побудовано апаратну складову, систему для передачі команд управління й систему для передачі відео з камери. Було використано USB-2.0 веб-камеру для збору інформації. Камера у поєднанні з ПМК дає можливість збору, передачі й обробки відеопотоку з роздільною здатністю 1280*720 пікселів та частотою до 30 FPS. Розроблена програма дає можливість передавати зображення прямо на сторінку веб-браузера через локальну мережу по Wi-Fi, використовуючи протокол HyperText Transfer Protocol (HTTP) у реальному часі.

За допомогою одного двигуна постійного струму з номінальною напругою від 4 до 6 вольт продемонстровано управління виконавчими механізмами, а саме: задніми колесами. Також було використано два сервоприводи – один для регулювання повороту передніх коліс і один для повороту USB-2.0 веб-камери. Управління двигуном і сервоприводами здійснювалося за допомогою зміни сигналу на портах виводів GPIO плати контролера на основі Raspberry Pi 4.

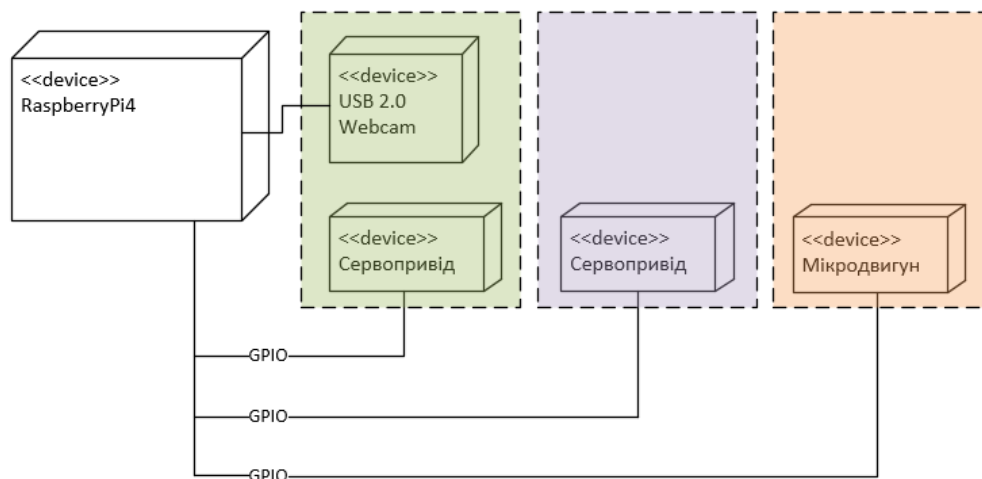


Рисунок 1 – Діаграма розгортання ПАІН

На рис. 1 зображено діаграму розгортання частини ПАІН на основі ПМК. Основним елементом управління ПАІН, що виконує всі операції обчислення та операції управління, є

мікрокомп'ютер. На діаграмі розгортання на рис. 1 зображено виконавчі механізми, які підключені через інтерфейс вводу-виведення ПКМ. На рис. 1 загалом відображено три контури управління, які проходять через ПКМ. ПКМ управляє безпосередньо запуском двигуна, сервоприводами і забезпечує передачу зображення у режимі онлайн. Окремо виділено контур передачі візуальної інформації через USB-камеру, що забезпечує передавання відео до ПКМ. Загалом система складається з підсистем керування двигуном, двома сервоприводами і відеоконтролю.

Отже, практичні результати підтвердили можливість реалізації діаграми розгортання (рис. 1).

На рис. 2 показана UML-діаграма компонентів, яка відображає компоненти даної рухомої платформи. Діаграму показано на рис 2.

Основою контролера ПАІН є Raspberry Pi 4 ПКМ (рис. 2). Компоненти плати під управлінням ОС Raspberry Pi Desktop зображено на рис. 2. У дослідницькій роботі як програмну інфраструктуру було використано Django Framework на основі Python.

Проаналізуємо діаграму компонентів більш докладно. Основні елементи управління двигуном і сервоприводами відокремлені від підсистеми обробки відео. Передача зображення з відеокамери відбувається через USB-інтерфейс. З відеоматеріалів можна отримати окремі зображення і обробити їх фреймовою мережею для налаштування автономного керування або надати управління користувачу через веб-додаток.

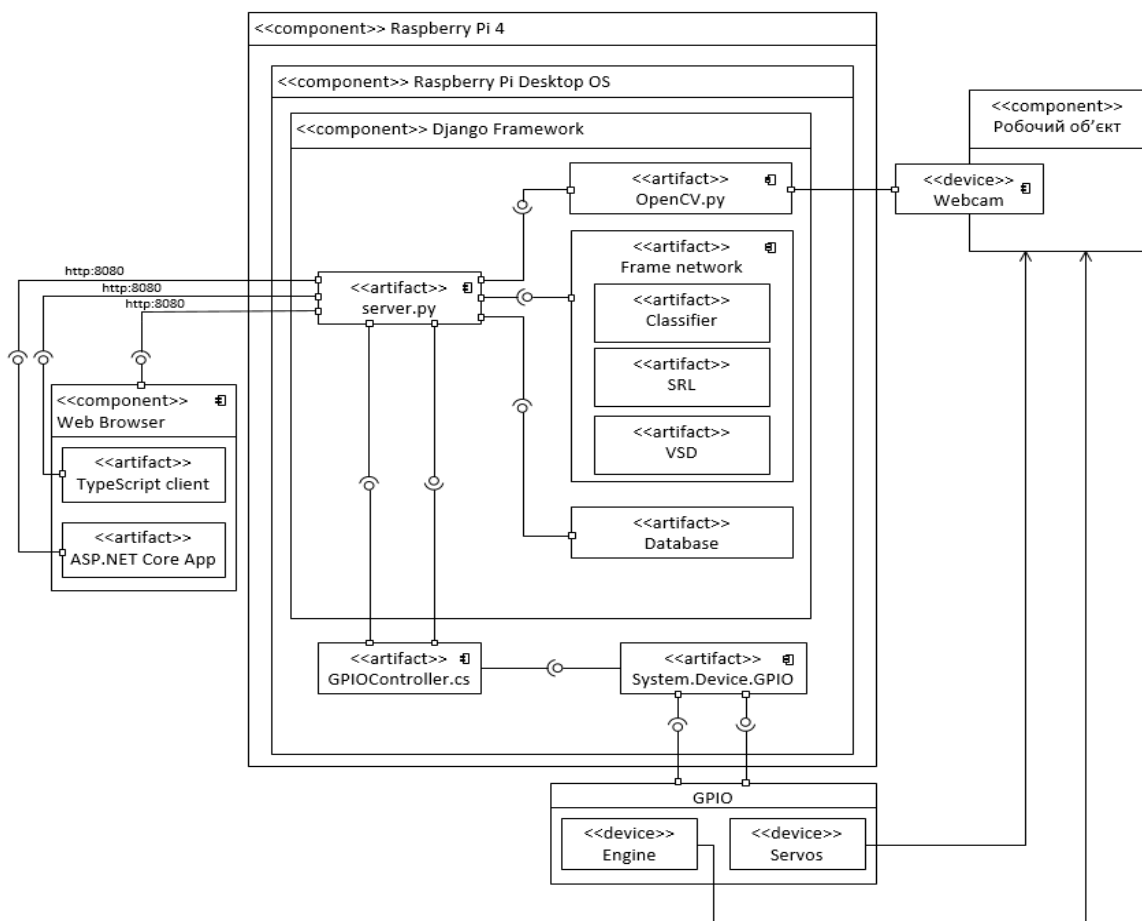


Рисунок 2 – Діаграма компонентів ПАІН без елементів ШІ

Для забезпечення роботи системи комп'ютерного зору використана бібліотека OpenCV-Python. Вона може бути інтегрована в Django Framework. Її особливість полягає у тому, що вона надає елементарний доступ до інтерфейсу захоплення відео в API OpenCV.

У ході дослідження виявлено, що саме OpenCV надає найбільш простий доступ до відеозображення з веб-камери. VideoWriter надає дані у форматі avi, який підтримується на будь-якій ОС. Бібліотека OpenCV дає можливість розробляти програмне забезпечення, яке дозволяє захоплювати відео з веб-камери [5].

Для управління апаратним інтерфейсом GPIO обрано бібліотеку System.Device.GPIO. Ця бібліотека забезпечує прямий доступ до плати мікропроцесора. Вона випускається за ліцензією Массачусетського технологічного інституту (MIT). Цю бібліотеку також можна використовувати з різними мовами програмування.

Відеодані зберігаються для подальшого ручного аналізу інформації. Під час дослідно-конструкторської роботи зображення зберігалися локально. Для збереження також можна використовувати хмарні сервіси для економії місця на microSD. Зображення з посиланнями зберігаються разом із даними про місцезнаходження в локальній SQLite реляційній базі даних.

Для зберігання даних була розроблена програмна інфраструктура, а також використано ASP.NET Core і Angular. Angular використовуємо для відображення керування і виведення зображення з камери [13].

Для оптимізації зв'язку між сервісами також можна використовувати заміну загальноприйнятих форматів обміну даними (XML, JSON) модифікованою сторонньою реалізацією, щоб зекономити час для передачі по протоколу HTTP.

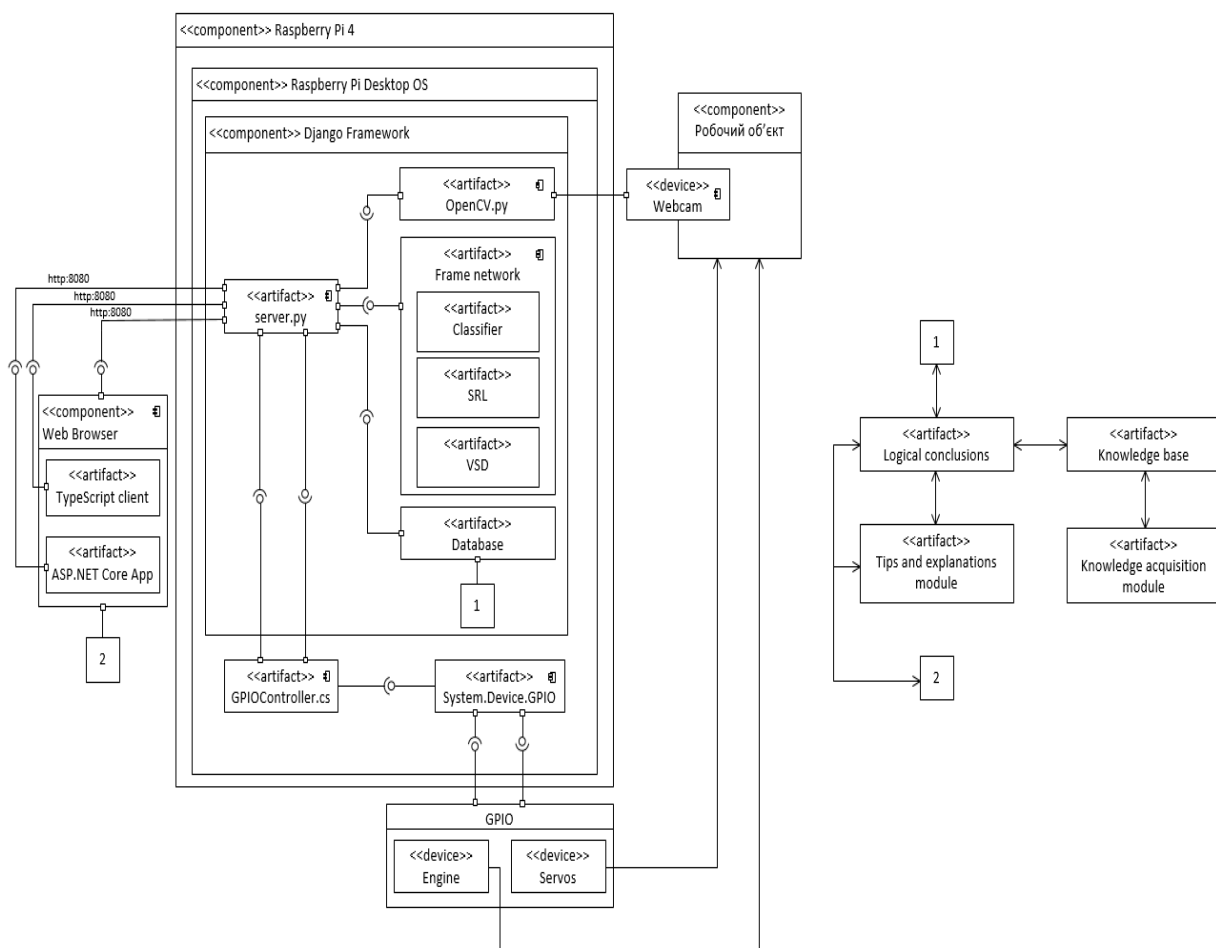


Рисунок 3 – Модифікована діаграма компонентів ПАІН

Для забезпечення роботи системи штучного інтелекту з використанням загальновідомих підходів [17] потрібні такі елементи у ПАІН:

1. Підсистема представлення знань у конкретній предметній області і управління ними. Для реалізації цих функцій використовується механізм, який називається базою знань.

2. Механізм, який на підставі знань, наявних у базі знань, здатний робити логічні висновки. Цей механізм називається механізмом логічних висновків.

3. Реалізація інтерфейсу для правильної передачі відповідей користувачеві. В іншому випадку користуватися такою системою надто незручно. Механізмом, що реалізує цю функцію, є користувальницький інтерфейс.

4. Підсистема підтримки бази знань і доповнення її при необхідності. Механізмом, що реалізує ці функції, є модуль придбання знань.

5. Система, що не тільки здатна надавати висновок, а й представляти різні коментарі, що додаються до цього висновку, і пояснювати його мотиви. В іншому випадку користувачеві буде важко зрозуміти висновок. Таке розуміння необхідне, якщо висновок використовується для консультації або надання допомоги при вирішенні будь-яких питань. Система називається модулем порад і пояснень.

На основі діаграми компонентів (рис. 2) було розроблено модифіковану діаграму компонентів (рис. 3). Вона містить інфраструктурні елементи для забезпечення роботи бази знань у механізмі штучного інтелекту. Для реалізації було обрано Keras API. Keras – це провідна бібліотека Python з відкритим кодом, написана для побудови нейронних мереж і проєктів машинного навчання.

5. Висновки

У статті побудовано модифіковану діаграму компонентів і на її основі розроблено робочий прототип ПАІН. Особливістю даного рішення є те, що система використовує фреймворк мережу для автоматизації управління і додатково передає зображення на екран веб-браузера.

У результаті розроблення прототипу змодельовано діаграму розгортання. Ця діаграма (рис. 1) демонструє типову схему апаратної інфраструктури ПАІН, запропоновані контури управління, виконавчі механізми. Таку схему можна використовувати як прототип стандартів аналогічних рішень.

Також результатом розробки є діаграма компонентів (рис. 2). Вона описує програмну інфраструктуру, використані бібліотеки, інтерфейси і зйомні модулі. На практиці досліджено програмну інфраструктуру під час розробки даного апарата. Саме ця інформація є основою базового елемента програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту.

У модифікованій діаграмі компонентів (рис. 3) намічені шляхи створення програмної і апаратної інфраструктури для підключення бази знань у складі програмно-апаратної інфраструктури наземної платформи з елементами штучного інтелекту.

СПИСОК ДЖЕРЕЛ

1. Armbian OS. URL: <https://www.armbian.com/download/>.
2. Microcomputer. URL: <https://www.sciencedirect.com/topics/computer-science/microcomputer/>.
3. Кадило А. Проблемність визначення свідомості та штучний інтелект. Львів: Національний університет «Львівська політехніка», 2014. 16 с.
4. Лепський А.С., Броневиц А.Г. Математичні методи штучного інтелекту: курс лекцій для студентів. Таганрог: Прикладна математика, 2009. 178 с.
5. Акулич М. Искусственный интеллект и маркетинг. М.: Издательские решения, 2018. URL: <http://litmir.biz/rd/268495>.

6. Demand For These Autonomous Delivery Robots Is Skyrocketing During This Pandemic. URL: <https://www.forbes.com/sites/bernardmarr/2020/05/29/demand-for-these-autonomous-delivery-robots-is-skyrocketing-during-this-pandemic/?sh=7db5d1647f3c/>.
7. Best Raspberry Pi Operating Systems for Various Purposes. URL: <https://itsfoss.com/raspberry-pi-os/>.
8. The Raspberry Pi Foundation. URL: <https://www.raspberrypi.org/about/>.
9. Raspberry Pi 4 Model B – міні-комп'ютер. URL: https://micro-pi.ru/raspberry-pi-4-model-b-rpi-4-b-bcm2711/#_Raspberry_Pi_4_Model_B-2/
10. OpenCV-Python Tutorials Video analysis. URL: https://docs.opencv.org/master/dd/d43/tutorial_py_video_display.html.
11. Керівництво з ASP.NET Core. URL: <https://metanit.com/sharp/aspnet5/>
12. Katin P. Development of variant of software architecture implementation for low-power general purpose microcontrollers by finite state machines. *Eureka: Physics and Engineering*. 2017. N 3. P. 49–55. URL: <https://eu-jr.eu/engineering/article/view/361/343/361-1019-1-PB.pdf>.
13. Katin P., Bessmertnyy R. Use of high-performance microcontroller for improving economic efficiency of jam production. *Standartyzatsiya. Sertyfikatsiya. Yakist*. 2019. N 3 (115). P. 69–77.
14. Katin P., Chmelov V., Shemaev V. Development of Typical 'State' Software Patterns for CortexM Microcontrollers in Real Time. *Eastern-European Journal of Enterprise Technologies*. 2020. N 3 (9(105)). P. 29–38. URL: <https://ssrn.com/abstract=3719729>.
15. Spinke V. An object-oriented implementation of concurrent and hierarchical state machines. *Information and Software Technology*. 2013. N 55 (10). P. 1726–1740.
16. Nyman, S. (2012). IAR Application Note # 6811-003. Implementing a State Machine. Available at: http://supp.iar.com/FilesPublic/SUPPORT/000370/AppNote_6811-003_State_Machine.pdf.
17. Уєно Х., Ісідзука М. Представлення і використання знань. Москва: Мир, 1987. 226 с.

Стаття надійшла до редакції 05.02.2021