

За результатами дослідження експериментально доведено можливість реалізації IDE середовища, кросплатформного за видом проектування та браузерного застосунку – за формою виконання, яке підтримує модель наскрізного процесу отримання Firmware. Практична цінність досліджень та розробки полягає в автоматизації процесів створення профілю користувача, генерації сценарію та проведення збірки проекту, завантаження Firmware та відладки вбудованого ПЗ. Виконання IDE у формі браузерного застосунку дозволяє відмовитись від використання незручного інтерпретатора Python, а варіацію автоматизації збірки задавати виключно підключенням оптимального js-контролера. Перенесення всього процесу отримання Firmware на локальний рівень забезпечує кращу захищеність проектів, розвантажує мережу та підвищує ефективність розробки. Інтеграція webDAV-клієнта у браузерний IDE забезпечує можливість управління версіями та кооперативні методи розробки ПЗ за GIT-доступом.

Література:

1. T. Semiawan Analysis of the Effectiveness and Efficiency of Software Development Tools / T. Semiawan, R. Alifi, H. Hayati: веб-сайт. URL: https://www.researchgate.net/publication/357216868_Analysis_of_the_Effectiveness_and_Efficiency_of_Software_Development_Tools (дата звернення: 12.05. 2023).
2. Efficient Embedded System Development: A Workbench for an Integrated Methodology/ N. Hili, Ch. Fabre, S. Dupuy-Chessa: веб-сайт. URL: <https://hal.science/hal-00671966/document> (дата звернення: 12.05. 2023).

УДК 004.4

ПРОГРАМНА ІНЖЕНЕРІЯ НМІ-РОЗШИРЕНЬ ВБУДОВАНИХ СИСТЕМ

Ярмілко А. В., Даценко Д. В.

Черкаський національний університет імені Богдана Хмельницького

Abstract. A generalized requirements list for developing embedded systems' NMI extensions has been compiled. An innovative GUI framework model has been proposed, segmented corresponding to the characteristics of the GUI libraries.

Програмна інженерія застосунків персональних комп'ютерів (PC), включаючи їх ультра-мобільний аналог (UMPC), має відмінності від розробки ПЗ вбудованих систем як за підходами аналізу вимог, проектування, конструювання, відлагодження, тестування, менеджменту конфігурування, супроводу, так і за використанням інструментальних засобів. Виокремлення із галузі інформаційних технологій спеціалізації «програмна інженерія вбудованих систем» дозволяє краще врахувати специфіку підходів, підвищити ефективність отримання такого виду програмного забезпечення, а отже є актуальною задачею досліджень. Особливість поточної стадії дослідження

полягає у звуженні його предмету до НМІ-розширень вбудованих систем на базі WiFi-платформ, апаратно-програмного концепту використання контролера швидкісної SPI-шини [1] та методів підвищення гарантоздатності та адаптивності систем критичної інфраструктури [2].

В ході досліджень узагальнено перелік вимог до розробки НМІ-розширень вбудованих систем:

- досягнення комплексності ведення та надання ітераційності процесу проектування НМІ в парадигмі UX/UI-дизайну;
- винесення процесу створення НМІ-образу в окреме виконання із застосуванням ефективних засобів редагування та візуалізації НМІ-моделі;
- забезпечення нечутливості до порядку та повноти створення UI-вікон в межах НМІ-моделі та їх наповнення віджетами;
- досягнення резистентності до помилок як у НМІ-моделі, так і в НМІ-образі.
- забезпечення некритичності до відображення віджетів при частковому або цілковитому незаповненні їх структур атрибутів та станів, зв'язування віджетів із змінними бізнес-моделі та обробниками системних подій;
- підтримка динамічного перевантаження НМІ-образу.

За результатами аналізу GUI-бібліотек μ GUI, MakiseGUI, LittlevGL та μ GFX встановлено, що вони не можуть задовольнити зазначені вимоги концептуально, тому наявний запит на отримання інноваційного GUI-фреймворку, модель застосування якого представлена на рис. 1 діаграмою компонентів. Модель фрагментовано за трьома рівнями, які розділені за ознаками стадійності використання GUI-бібліотеки НМІ-розширення для вбудованих систем. На верхньому рівні відображено інструментальний набір, який включає програмні компоненти-плагіни розширення середовища розробки IDE Visual Studio Code (VS Code), бібліотеки підрівня GUI HMISys та вбудованого застосунку emAppProj.

Компоненти згруповані за функціональною ознакою. Так, агрегація ToolChain та SDK Firmware обумовлена тим, що розробка ПЗ WiFi-модулів можлива лише за їх спільного використання. Рівень їх залежності відповідає процедурній та послідовній згуртованості. Такі підходи виправдані і для набору плагінів верстки веб-сторінок, хоча ступінь їх залежності низький і може бути класифікований як випадкова згуртованість. Програмний пакет GUI HMISys, як правило, вкладений у пакет emAppProj. Ця залежність на діаграмі умовно не показана, хоча вони логічно згуртовані. Причина полягає у їх функціональній згуртованості відносно середовища розробки.

Компонента HTTP-Client слугує оболонкою контейнера HTML2HMI як веб-застосунка, що надає функціонально-послідовної агрегації компонентам Convertor, HMI_iFrame та GetValue, призначенням яких є відображення моделі та перетворення її у НМІ-образ. Нашарування браузера над операційною

системою, як і базування останньої на платформі PC чи UMPC, умовно не показані в силу очевидної їх зв'язаності.

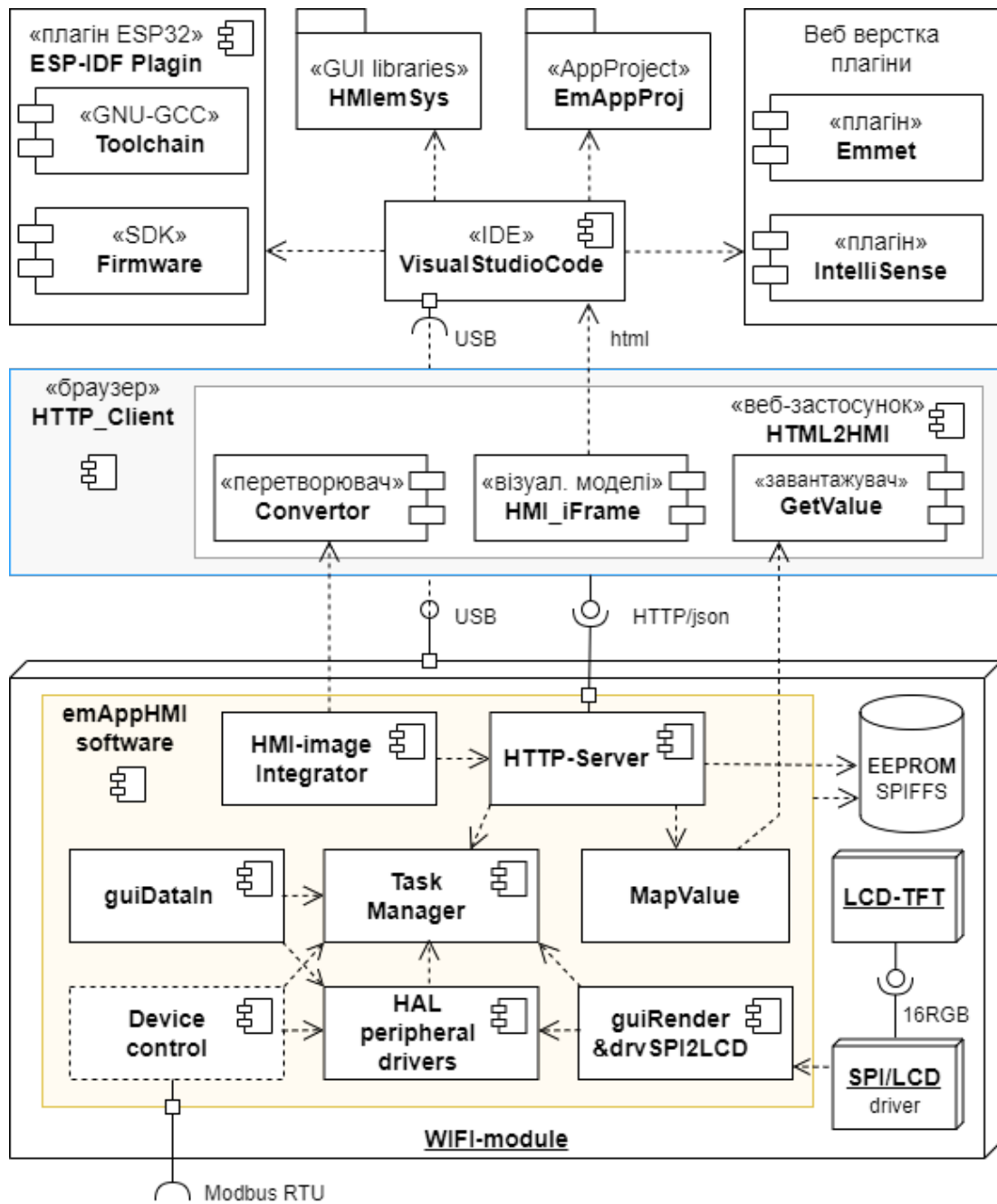


Рисунок 1 – Проектна модель розробки HMI-розширень вбудованих систем

Третій фрагмент діаграми відображає відношення між компонентом emAppHMI software, апаратною платформою WiFi-модуль, апаратним драйвером SPI/LCD дисплеєм та бортовим накопичувачем, які функціонально згуртовані в межах реалізації цільового призначення. Компоненти групи emAppHMI software суміщають взаємну логічну зв'язаність та процедурно-послідовну згуртованість. Такий значний рівень компонентної когезії, як прояв високого рівня згуртованості та зчеплення, засвідчує належну якість архітектурного дизайну проекту.

В результаті досліджень програмної інженерії НМІ-розширень вбудованих систем на базі WiFi-платформ узагальнено вимоги, отримано проектну модель та фреймворк як більш досконалу форму реалізації GUI-бібліотеки. Цей фреймворк здатен забезпечити реалізацію усього вище зазначеного переліку вимог. Включення до складу вбудованої програми emAppHMI software компонент НМІ-imageIntegrater дозволяє проводити завантаження НМІ-образу динамічно як на моменті старту програми, так і у процесі її виконання. Надання ізоморфності віконного інтерфейсу дозволяє досягти більш повної адаптованості вбудованої системи до зміни умов функціонування. Така властивість має виняткову цінність для засобів критичної інфраструктури.

Література:

1. A. Yarmilko, "High-Speed SPI Bus Host Controller for Embedded Systems," 2022 IEEE 41st International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 2022, pp. 662-666, doi: 10.1109/ELNANO54667. 2022.9927055: веб-сайт. URL: <https://ieeexplore.ieee.org/document/9927055> (дата звернення: 10.05.2023)
2. A. Yarmilko, D. Datsenko 'Increase of the dependability and adaptability of critical infrastructure' IIoT systems. Міжнародна конференція INUDECO: веб-сайт. URL: https://drive.google.com/drive/folders/1HLLS_R7kaD6anJ8nufYavN7rItSynf8?usp=sharing (дата звернення: 10.05.2023)

УДК 004.454

ДРАЙВЕРИ ЕКСПАНДЕРА SPI IIOT-СИСТЕМ

Ярмілко А. В., Олександренко М. П.

Черкаський національний університет імені Богдана Хмельницького

Abstract. The proposed hardware and software drivers' structure is aimed at implementing IIoT systems on a WiFi module platform. The driver enables a more comprehensive utilization of the WiFi module computational power by expanding the SPI bus.

Часто у склад вбудованих систем інтегровано WiFi-модуль задля зручності комунікації на рівні WirelessLAN або віддаленого доступу через мережевий маршрутизатор. Використання сучасних WiFi-модулів виключно для інформаційного обміну «точка доступу – станція» не ефективно, постільки така задача потребує всього 10..15% від обчислювальної потужності. Надлишок, понад 80% ресурсу, може залучатись на вирішення задач роботи із твердотільними накопичувачами даних, забезпечення людино-машинної взаємодії та завдань утилітарного змісту (наприклад, управління 3D-друком). На практиці більш повне використання цього ресурсу ускладнено обмеженням