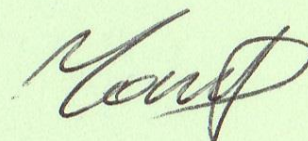


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

**Чопей Ратібор Степанович**



УДК 004.054

**ЗАСОБИ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ  
СПЕЦІАЛІЗОВАНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ВБУДОВАНИХ СИСТЕМ**

*01.05.03 – математичне та програмне забезпечення  
обчислювальних машин і систем*

**Автореферат**  
дисертації на здобуття наукового ступеня  
кандидата технічних наук

Львів – 2019

## Дисертацією є рукопис

Робота виконана у Національному університеті «Львівська політехніка» Міністерства освіти і науки України.

**Науковий керівник:** доктор технічних наук, професор  
**Федасюк Дмитро Васильович,**  
Національний університет «Львівська політехніка»,  
проректор з науково-педагогічної роботи, професор  
кафедри програмного забезпечення.

**Офіційні опоненти:** доктор технічних наук, професор  
**Дивак Микола Петрович,**  
Тернопільський національний економічний університет,  
декан факультету комп'ютерних інформаційних  
технологій;

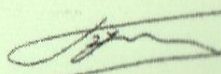
кандидат технічних наук, доцент  
**Мельничин Андрій Володимирович,**  
Львівський національний університет імені Івана Франка,  
доцент кафедри теорії оптимальних процесів.

Захист дисертації відбудеться 4 квітня 2019 р. о 14<sup>00</sup> год. на засіданні спеціалізованої вченої ради Д 35.052.05 у Національному університеті «Львівська політехніка» (79013, Львів, вул. С. Бандери, 12, ауд. 226 головного корпусу).

З дисертацією можна ознайомитись у науковій бібліотеці Національного університету «Львівська політехніка» (79013, м. Львів, вул. Професорська, 1).

Автореферат розісланий “2” березня 2019 р.

Учений секретар спеціалізованої  
вченої ради, д.т.н., професор



Р.А. Бунь

## ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

**Актуальність теми.** Зростання продуктивності мікроконтролерів, що зумовлено застосуванням 32-бітних процесорів з архітектурою ARM спричинило розширення сфер застосування вбудованих систем, у тому числі уможливило їх використання для систем реального часу. Такі системи широко використовуються в різних галузях життєдіяльності людей (автомобільна, авіація, виробництво, медицина, сільське господарство, тощо).

Системи реального часу характеризуються тим, що в них правильність одержаних результатів залежить не тільки від логічної коректності програми, але й від проміжку часу, за який вони отримані. Залежно від наслідків, зумовлених порушенням часових вимог, системи реального часу прийнято поділяти на системи “м’якого” та “жорсткого” реального часу. У системі “м’якого” реального часу порушення часових вимог призведе до зниження якості роботи системи. Водночас, у системах “жорсткого” реального часу порушення часових вимог є рівнозначним відмові системи, а відтак, може, призвести до катастрофічних наслідків. Саме до таких систем висуваються більш жорсткі вимоги щодо їхньої надійності та безпечності. Забезпечення цих вимог є неможливим без застосування засобів автоматизованого тестування програмного коду. При цьому для систем жорсткого реального часу тестування тривалості виконання програмного коду набуває особливої актуальності.

Ще у 1990-х роках низка науковців (J. Engblom, A. Colin, G. Bernat, R. Chapman, C. Ferdinand, C. Healy, P. Atanassov та ін.) працювала над питаннями розроблення й дослідження моделей та методів аналізу тривалості виконання програмного коду вбудованих систем, що забезпечують низький рівень похибки отриманих результатів. У зв’язку із зростанням складності архітектури мікроконтролерів, а також ускладненням та розширенням функціоналу вбудованих систем зростає потреба в розробленні нових методів вимірювання тривалості виконання програмного коду, які не передбачають створення моделей апаратного забезпечення.

Одним із важливих упущень наявних методів є те, що жоден із них не дає змогу враховувати вплив зовнішніх і внутрішніх чинників на вбудовану систему та зміну тривалості виконання програмного коду під дією цих впливів. Сукупність наведених обставин породила актуальне наукове завдання удосконалення наявних методів тестування тривалості виконання програмного коду вбудованих систем реального часу і створення відповідного програмного засобу для автоматизації процесу їх тестування. Актуальність цього наукового завдання, його практичне значення й зумовили вибір теми, мети, завдань і структури дисертаційного дослідження.

**Зв’язок роботи з науковими програмами, планами, темами.** Дисертація відповідає науковому напрямку кафедри програмного забезпечення Національного університету «Львівська політехніка» “Програмне і математичне забезпечення автоматизованих систем”. Дисертація виконана в межах науково-дослідних робіт фінансованих Міністерством освіти і науки України та госпдоговірних науково-дослідних робіт, які виконувалися в Національному університеті «Львівська

політехніка»: «Розроблення математичного забезпечення для програмного засобу аналізу функціональної безпечності та надійності програмно-апаратних систем відповідального призначення» (№ держреєстрації 0117U004458, 2018); госпдоговірні науково-дослідні роботи №0538, 0565 та 0581, що виконувалися впродовж 2016–2018 р. за тематикою «Автоматизоване тестування вбудованого програмного забезпечення». У перелічених науково-дослідних роботах автор розробив методи тестування тривалості виконання програмного коду вбудованих систем реального часу і створив відповідний програмний засіб для автоматизації процесу їх тестування.

**Мета та завдання дослідження.** Метою дисертаційної роботи є розроблення методів та алгоритмів тестування тривалості виконання програмного коду вбудованих систем із урахуванням поведінки периферійних пристроїв, які забезпечують заданий рівень надійності під час дії на них зовнішніх та внутрішніх чинників.

Для досягнення зазначеної мети поставлені і вирішені такі завдання:

1) проаналізувати структурні та функціональні особливості процесу тестування мікроконтролерних вбудованих систем реального часу, що дасть змогу сформулювати принципи, яких потрібно дотримуватися при розробленні програмного засобу для їх тестування;

2) проаналізувати наявні методи й засоби аналізу тривалості виконання програмного коду, встановити їх переваги та недоліки, що дасть змогу визначити напрямки їх удосконалення для підвищення ефективності;

3) удосконалити метод статичного аналізу тривалості виконання програмного коду для мікроконтролерних вбудованих систем, який враховує їхню внутрішню архітектуру, що дасть змогу врахувати втрати швидкості надсилання даних між внутрішніми модулями та при надсилання даних через інтерфейси зв'язку;

4) розвинути метод динамічного тестування тривалості виконання програмного коду вбудованих систем, який враховує особливості мікроконтролерів з архітектурою ARM, що дасть змогу використовувати його для більшості сучасних аналогічних вбудованих систем;

5) розробити метод формування плану тестування програмних функцій вбудованої системи, який враховує поведінку периферійних пристроїв вбудованої системи, що забезпечить підвищення ефективності її тестування;

6) розробити метод прогнозування тривалості виконання програмного коду, який ґрунтується на аналізі впливу зовнішніх та внутрішніх чинників на периферійні пристрої вбудованої системи, що дасть змогу підвищити точність отриманих результатів;

7) розробити, реалізувати та верифікувати програмний засіб для тестування тривалості виконання програмного коду на підставі розроблених методів, моделей та алгоритмів, що дасть змогу автоматизувати процес тестування спеціалізованого програмного забезпечення (ПЗ) вбудованих систем реального часу.

**Об'єктом дослідження** є автоматизоване тестування тривалості виконання програмного коду вбудованих систем реального часу.

**Предметом дослідження** є методи та алгоритми тестування тривалості виконання програмного коду вбудованих систем реального часу з урахуванням

поведінки периферійних пристроїв, а також впливу на них зовнішніх та внутрішніх чинників.

**Методи дослідження.** Для вирішення поставлених у дисертаційній роботі завдань використано такі методи дослідження: методи натурального експерименту та теорії алгоритмів – для методів динамічного аналізу тривалості виконання програмного коду; теорію ймовірності та математичну статистику – для побудови моделі функціонування програмного забезпечення, що враховують поведінку периферійних пристроїв; теорія системного аналізу та теорія математичного моделювання – для оцінювання впливу зовнішніх та внутрішніх чинників на тривалість виконання програмного коду; теорію алгоритмів, принципи аспектно-та об'єктно-орієнтованого програмування – для розроблення програмних засобів.

**Наукова новизна одержаних результатів.** Основні результати роботи, які визначають її наукову новизну та виносяться на захист:

1. Вперше розроблено метод прогнозування тривалості виконання програмного коду, який враховує вплив зовнішніх і внутрішніх чинників на периферійні пристрої вбудованої системи реального часу, що дає змогу забезпечити заданий рівень їхньої надійності та безпечності.

2. Удосконалено метод статичного аналізу тривалості виконання програмного коду для мікроконтролерних вбудованих систем, який, на відміну від наявних, враховує швидкість обміну даними між внутрішніми модулями мікроконтролера, що дає змогу підвищити точність отриманих результатів.

3. Отримав подальший розвиток метод формування плану тестування програмних функцій вбудованої системи, в якому, на відміну від відомого, враховано поведінку її периферійних пристроїв, що дає змогу оцінити важливість тестування кожної програмної функції, і в такий спосіб підвищити ефективність процесу автоматизованого тестування.

4. Отримав подальший розвиток метод динамічного тестування програмного забезпечення вбудованих систем з використанням інтерфейсу відлагодження, в якому, на відміну від відомого, враховано особливості мікроконтролерів з архітектурою ARM, що дає змогу виконувати тестування тривалості виконання програмних функцій вбудованої системи без внесення змін до її програмного чи апаратного забезпечення.

**Практичне значення одержаних результатів** полягає в тому, що:

1. Розроблено архітектуру програмного засобу для автоматизованого тестування тривалості виконання програмного коду, використання якої дає змогу підвищити ефективність процесу тестування програмного забезпечення вбудованих систем.

2. Розроблено програмне та інформаційне забезпечення програмного засобу для автоматизованого тестування програмного забезпечення вбудованих систем, яке дає змогу виконувати тестування та прогнозування тривалості виконання програмного коду з урахування впливу зовнішніх і внутрішніх чинників на вбудовану систему.

Розроблений програмний засіб EXTT на основі розроблених методів впроваджено в процес тестування вбудованих систем компанії Dinamica Generale S.p.A (м. Поджо Руско, Італія), що підтверджено відповідним актом. Розроблений

метод формування плану тестування, що враховує поведінку периферійних пристроїв впроваджено в практику тестування вбудованих систем компанії ПП НВП «Спаринг-Віст Центр» (м. Львів). Також, результати дисертаційного дослідження використовуються у навчальному процесі Національного університету «Львівська політехніка» для дисциплін «Програмування мікроконтролерів». Використання та впровадження результатів дисертаційної роботи підтверджено відповідними актами.

*Результати дисертаційної роботи використані:*

- у держбюджетній науково-дослідній роботі “Розроблення математичного забезпечення для програмного засобу аналізу функціональної безпечності та надійності програмно-апаратних систем відповідального призначення” (номер держреєстрації №0117U004458);

- у трьох госпдоговірних науково-дослідних роботах №0538, 0565 та 0581, що виконувалися впродовж 2016 – 2018 р. за тематикою “Автоматизоване тестування вбудованого програмного забезпечення”.

**Особистий внесок здобувача.** Усі наукові результати дисертаційної роботи отримані здобувачем особисто. У друкованих працях, написаних у співавторстві, здобувачеві належать: [2, 11, 12] – аналіз та формування напряму наукових досліджень; [3] – метод аналізу тривалості виконання програмного коду з урахуванням архітектури мікроконтролера; [7, 9, 10] – архітектура засобу тестування тривалості виконання програмного коду та модель процесу віддаленого тестування; [5, 8] – модель для оцінювання тривалості виконання програмного коду з урахуванням поведінки периферійних пристроїв вбудованої системи; [1, 4, 6] – метод аналізу тривалості виконання програмного коду з урахуванням впливу зовнішніх і внутрішніх чинників на апаратне забезпечення вбудованих систем та методу прогнозування тривалості виконання програмного коду.

**Апробація результатів дисертації.** Основні положення і результати роботи доповідалися та обговорювалися на 6 міжнародних наукових конференціях: XIV та XV Міжнародних науково-практичних конференціях “Математичне і програмне забезпечення інтелектуальних систем” «MPZIS» (Дніпро, 2016, 2017); XIV International Conference the Experience of Designing and Application of CAD Systems in Microelectronics «CADSM» (Поляна, 2017); IX International Academic Conference of Young Scientists «Computer Science and Engineering CSE-2017» (Львів, 2017); XIV International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering «TCSET-2018» (Славське, 2018); Міжнародній науково-практичній конференції «Електротехнічні та комп’ютерні системи: Теорія та практика ЕЛТЕКС-2018» (Одеса, 2018).

Результати дисертаційної роботи неодноразово обговорювалися на наукових семінарах кафедри програмного забезпечення Національного університету «Львівська політехніка».

**Публікації.** За результатами досліджень, які викладено в дисертаційній роботі, опубліковано 12 наукових праць, серед яких 3 статті в наукових фахових виданнях України, 1 стаття у науковому виданні України, що внесене до міжнародної наукометричної бази Scopus; 2 статті в наукових періодичних

виданнях інших держав; 6 публікацій у збірниках праць міжнародних конференцій, серед яких 2 внесено в міжнародну наукометричну базу Scopus.

**Структура та обсяг роботи.** Дисертація складається із вступу, п'яти розділів, висновків, списку використаних джерел із 110 найменувань та 2 додатків. Загальний обсяг роботи становить 154 сторінки, з них 112 основного тексту, 45 рисунків та 35 таблиць.

## ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** проведено обґрунтування актуальності тематики дисертаційного дослідження, сформульовано мету та завдання роботи, визначено об'єкт і предмет дослідження, відображено наукову новизну та практичне значення отриманих результатів, представлено зв'язок роботи з науковими програмами та темами. Також охарактеризовано особисту участь здобувача у публікаціях, написаних у співавторстві, та апробацію результатів дисертації.

У першому розділі дисертації – **“Стан проблеми тестування спеціалізованого програмного забезпечення вбудованих систем”** – проведено аналіз конструктивних та функціональних особливостей вбудованих систем та наведено їх особливості в порівнянні із системами, що розробляються на базі комп'ютерів. Представлено необхідність проведення аналізу тривалості виконання програмного коду вбудованих системах реального часу.

Дослідження наявних методів аналізу тривалості виконання програмного коду показало, що використання статичних методів аналізу тривалості виконання програмного коду мають ряд недоліків, що обмежує їх застосування, а саме: необхідність розроблення моделей апаратного забезпечення для кожної вбудованої системи і є найбільш трудомістким етапом, що зумовлено розвитком апаратного забезпечення, а також відсутністю даних про їхні часові характеристики. Внесення помилок при розробленні моделей апаратного забезпечення призводить до зниження адекватності отриманих результатів.

З іншого боку, методи динамічного аналізу тривалості виконання програмного коду мають потенційну перевагу над статичними методами позаяк передбачають проведення відповідних вимірювань безпосередньо на вбудованій системі, а тому ці методи не потребують створення моделей апаратного забезпечення. Саме тому, їх частіше використовують на практиці. До недоліків методів динамічного аналізу можна віднести складність пошуку вхідних даних, які призводять до найбільш тривалого виконання програми. Встановлено, що наявні методи статичного та динамічного аналізу тривалості виконання програмного коду не враховують вплив часу відгуку периферійних пристроїв вбудованої системи та вплив зовнішніх і внутрішніх чинників на тривалість виконання програмного коду.

Таким чином, розвиток наявних моделей, методів та засобів автоматизованого тестування тривалості виконання програмного коду дасть змогу підвищити точність одержаних результатів та ефективність процесу тестування і в такий спосіб, прискорити процес тестування вбудованої системи, що знизить собівартість розроблення вбудованої системи.

У другому розділі дисертації – “**Методи динамічного аналізу тривалості виконання програмного коду**” – сформульовано основні принципи, які необхідно враховувати під час розроблення методів та засобів тестування тривалості виконання програмного коду вбудованих систем. На підставі цих принципів розроблено методи динамічного аналізу. Ці методи, як і відомі методи динамічного аналізу, виконуються з допомогою апаратного трасування через інтерфейс відлагодження, проте на відміну від відомих враховують особливості мікроконтролерів з архітектурою ARM. Саме це дало змогу проводити тестування вбудованої системи без впливу на хід виконання її програми, а відтак і на тривалість її виконання.

Розроблений метод динамічного аналізу передбачає вимірювання тривалості виконання програмного коду за різних наборів вхідних значень, для пошуку найбільшої тривалості виконання. На підставі розробленого методу динамічного аналізу тривалості виконання програмного коду розроблено архітектуру програмного модуля (рис. 1) та алгоритм, що його реалізує (рис. 2). Описаний метод складається з таких основних кроків:

**Крок 1.** Пошук початкової адреси програмної функції, за якою вона розміщується у пам’яті програм вбудованої системи. На цьому кроці проводиться синтаксичний аналіз тар-файлу, що генерується у процесі компіляції проекту. Цей файл містить імена всіх функцій, які увійшли до збірки, а також імена глобальних змінних та адреси, за якими вони розміщені в оперативній пам’яті.

**Крок 2.** Пошук кінцевої адреси функції. Цей крок передбачає проведення синтаксичного аналізу listing-файлу, що генерується у процесі компіляції. Listing-файл містить програмний код мовою C та його інтерпретацію асемблерним кодом. Шляхом статичного аналізу listing-файлу визначаємо кінцеву адресу функції.

**Крок 3.** Запуск режиму відлагодження проекту в середовищі Keil  $\mu$ Vision та початок виконання програми.

**Крок 4.** Встановлення точок зупинки за адресами, що відповідають початку та закінченню функції.

**Крок 5.** Встановлення значень глобальних змінних, що забезпечують перехід до виконання функції, яка підлягає тестуванню.

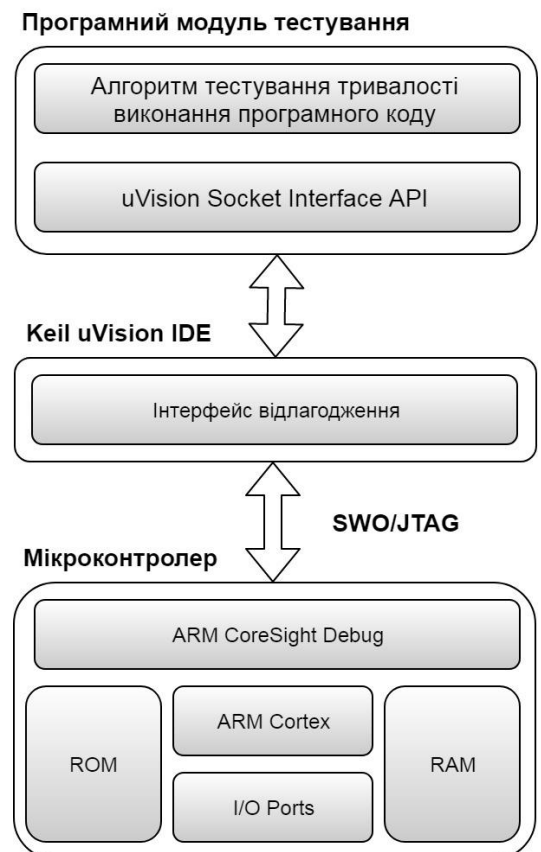


Рис. 1 – Архітектура програмного модуля для тестування тривалості виконання програмного коду



Крок 6. Запуск таймера в обробнику переривання, виклик якого зумовлений зупинкою виконання програми мікроконтролера у встановленій точці.

Крок 7. Надсилання команди продовження виконання програми.

Крок 8. При повторному виклику обробника переривання, зупиняємо вимірювання тривалості виконання програмної функції та відображаємо отримані результати вимірювання.

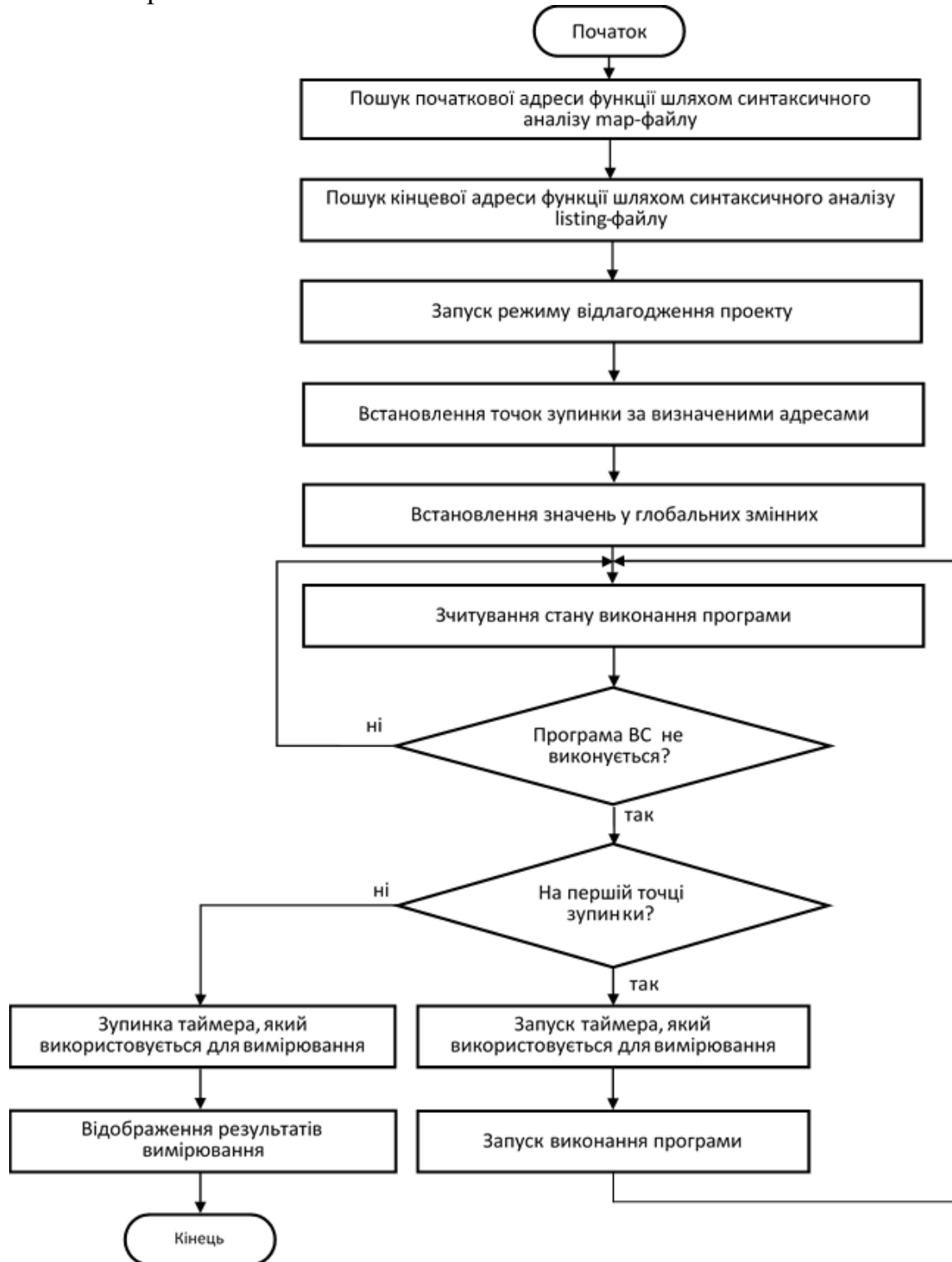
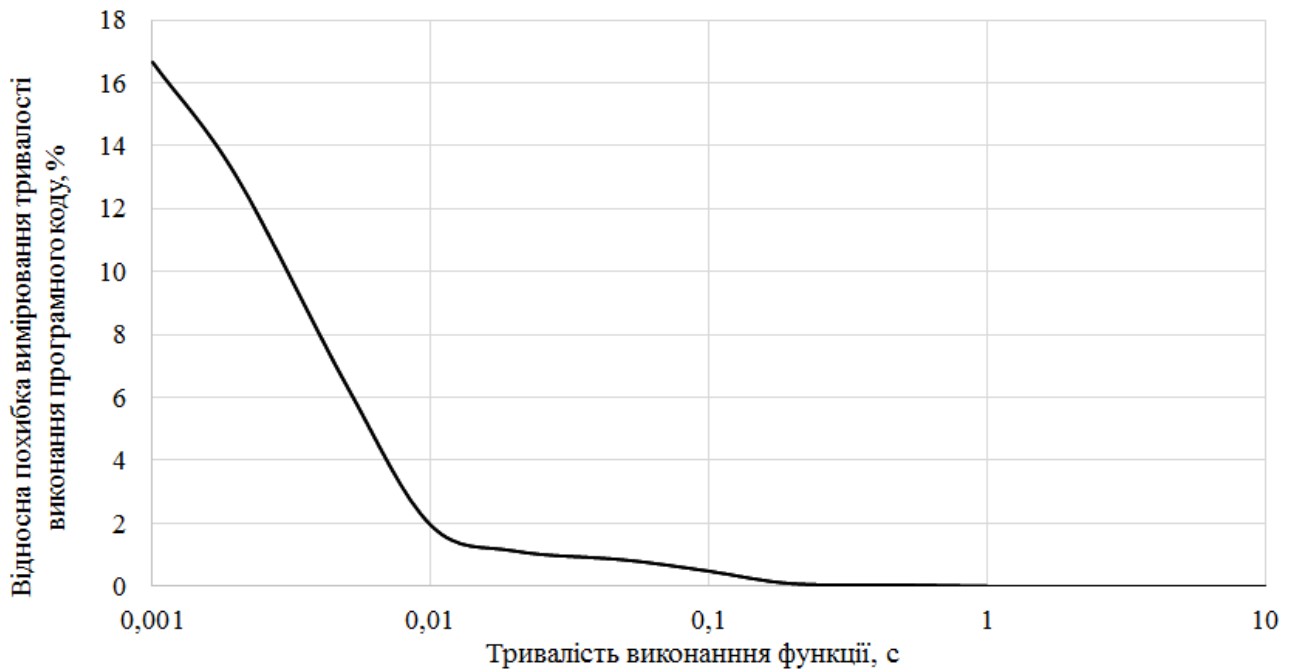


Рис. 2 – Блок-схема реалізації алгоритму методу динамічного тестування тривалості виконання програмного коду мікроконтролерних вбудованих систем

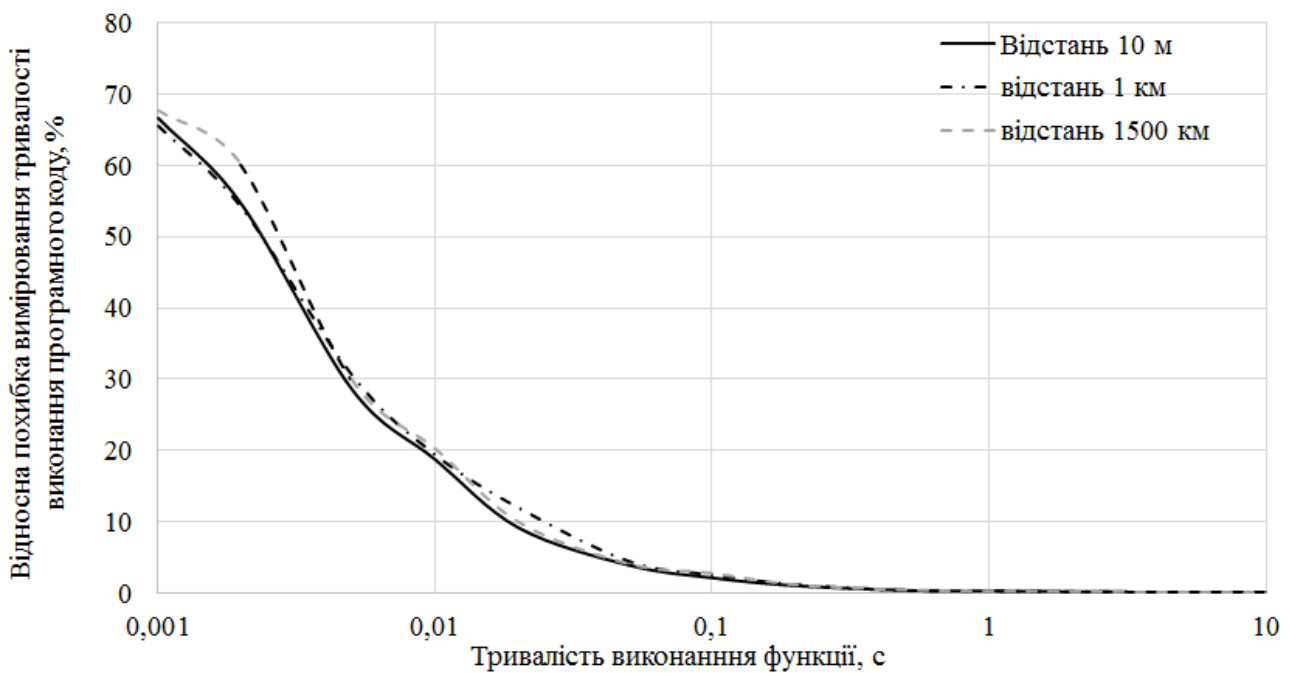
Описаний метод дає змогу автоматизувати процес тестування тривалості виконання програмного коду, що підвищує ефективність тестування та дає змогу знизити часові витрати на тестування вбудованих систем у цілому.

Проведено дослідження залежності відносної похибки вимірювання тривалості виконання програмного коду від тривалості виконання функції (рис. 3).

Отримані результати підтверджують доцільність використання розроблених методів для аналізу тривалості виконання програмного коду мікроконтролерних вбудованих систем жорсткого реального часу.



(а)



(б)

Рис. 3 – Залежності відносної похибки вимірювання тривалості виконання програмного коду від тривалості виконання функції: (а) – метод вимірювання тривалості виконання програмного коду; (б) – метод віддаленого тестування тривалості виконання програмного коду

У третьому розділі дисертації – **“Оцінювання тривалості виконання програмного коду з урахуванням часової поведінки периферійних пристроїв”** – для врахування поведінки периферійних пристроїв удосконалено модель функціонування ПЗ. У моделі розглянуто процес зміни часу відгуку периферійних пристроїв та відповідно зміни тривалості виконання програмних функцій.

Для розробленої моделі функціонування ПЗ представлено розвинутий метод формування плану тестування вбудованих систем, який враховує поведінку периферійних пристроїв. Застосування розвинутого методу формування плану тестування дає змогу підвищити ефективність тестування програмного коду вбудованих систем, а саме, забезпечує зменшення часових витрат на тестування.

Розроблено алгоритм побудови графу потоку керування за текстом програми мовою C, що дає змогу отримати матрицю інцидентності для будь-якої функції. Це дає змогу більш ефективно аналізувати текст програми, а саме: визначати кількість гілок у кожній функції, умови переходу в них та визначати кількість виконань циклу.

Удосконалено метод синтаксичного аналізу тривалості виконання програмного коду за рахунок урахування втрати швидкості при обміні даними між внутрішніми модулями мікроконтролерів, а також при надсиланні даних через інтерфейси зв'язку (UART, I<sup>2</sup>C, SPI та інші). У ролі вхідних даних цей метод використовує текст програми, а також файли, що створюються середовищем розробки при компіляції проекту. На основі цих вхідних файлів шляхом синтаксичного аналізу визначаємо налаштування мікроконтролера та його тактової частоти. Після цього встановлюємо взаємозв'язки між шинами синхронізації та внутрішніми модулями. Наступним етапом методу є визначення налаштувань інтерфейсів, що використовуються для зв'язку з зовнішніми периферійними пристроями. На підставі отриманих даних розраховуємо тривалість виконання програмного коду за формулою:

$$t_f = \sum_{i=1}^m (n_i \cdot t_i + tr_i), \quad (1)$$

де:  $m$  – кількість асемблерних інструкцій у функції;  $n_i$  – кількість тактів, за яку виконується асемблерна інструкція;  $t_i$  – тривалість виконання однієї обчислювальної операції, або тривалість надсилання одного байту інформації через інтерфейс зв'язку, що розраховується за наведеною нижче формулою;  $tr_i$  – тривалість читання інструкції з пам'яті програм.

Тривалість надсилання одного байту даних через інтерфейс зв'язку визначаємо за формулою:

$$t_{intf} = t_{APB} + t_{bintf} = \frac{M}{f_{APB}} + \frac{N}{f_{intf}}, \quad (2)$$

де:  $t_{bntf}$  – тривалість надсилання одного байту через інтерфейс зв'язку, що включає надсилання  $N$  додаткових бітів;  $M$  – кількість інформаційних бітів у

повідомленні,  $N$  – кількість додаткових бітів, що визначається конфігурацією інтерфейсу;  $f_{intf}$  – робоча частота інтерфейсу зв'язку.

На основі розробленого методу статичного аналізу тривалості виконання програмного коду проводиться дослідження залежності зміни часу відгуку на тривалість її виконання. Для цього розділяємо всі програмні інструкції на дві групи: тривалість виконання яких не залежить від периферійних пристроїв та тривалість виконання яких залежить від периферійних пристроїв. Використовуючи метод статичного аналізу розраховуємо тривалість виконання фрагментів програми, що не залежать від периферійних пристроїв.

На другому етапі досліджуємо діапазон зміни тривалості виконання програмного коду таким чином:

- з документації на периферійні пристрої визначаємо мінімальну та максимальну тривалість для кожної операції;
- на підставі даних отриманих із документації розраховуємо мінімальну та максимальну тривалість, що необхідна для виконання певного фрагменту програми;
- припускаємо, що тривалість виконання програмної інструкції, яка залежить від апаратного забезпечення, змінюється згідно з нормальним законом розподілу; тому для оцінки мінімальних та максимальних значень випадкової складової часу виконання функції застосовується метод Монте-Карло; для забезпечення бажаної точності отриманих результатів використовуємо формулу  $1/\sqrt{N}$ , де  $N$  – це кількість виконаних числових експериментів;
- для згенерованих значень затримок розраховуємо значення середньої тривалості виконання функції та значення її дисперсії за формулою:

$$D\xi \approx \frac{1}{N-1} \left[ \sum_{j=1}^N \xi^2 - \frac{1}{N} \left( \sum_{j=1}^N \xi_j \right)^2 \right]. \quad (3)$$

- розраховуємо суму та різницю середньої тривалості виконання функції та дисперсії випадкової величини.

На третьому етапі розраховуємо ваговий коефіцієнт потоку, що викликає цю функцію, а також розраховуємо вагу функції:

$$w_{coef} = (priority + 4) \cdot \left( \frac{P_{creation}}{periodicity} \right), \quad (4)$$

де:  $priority$  – пріоритет потоку, у якому викликається досліджувана функція;  $periodicity$  – період повторного виклику потоку, у якому викликається досліджувана функція;  $P_{creation}$  – ймовірність створення потоку, у якому викликається досліджувана функція;

$$w_{function} = \frac{T_{max} - T_{min}}{2} \cdot w_{coef}, \quad (5)$$

де  $T_{max}$ ,  $T_{min}$  – мінімальна та максимальна тривалість виконання функції, що розрахована на попередніх етапах.

Алгоритм, що реалізує метод формування плану тестування з урахуванням поведінки периферійних пристроїв, представлено на рис. 4.

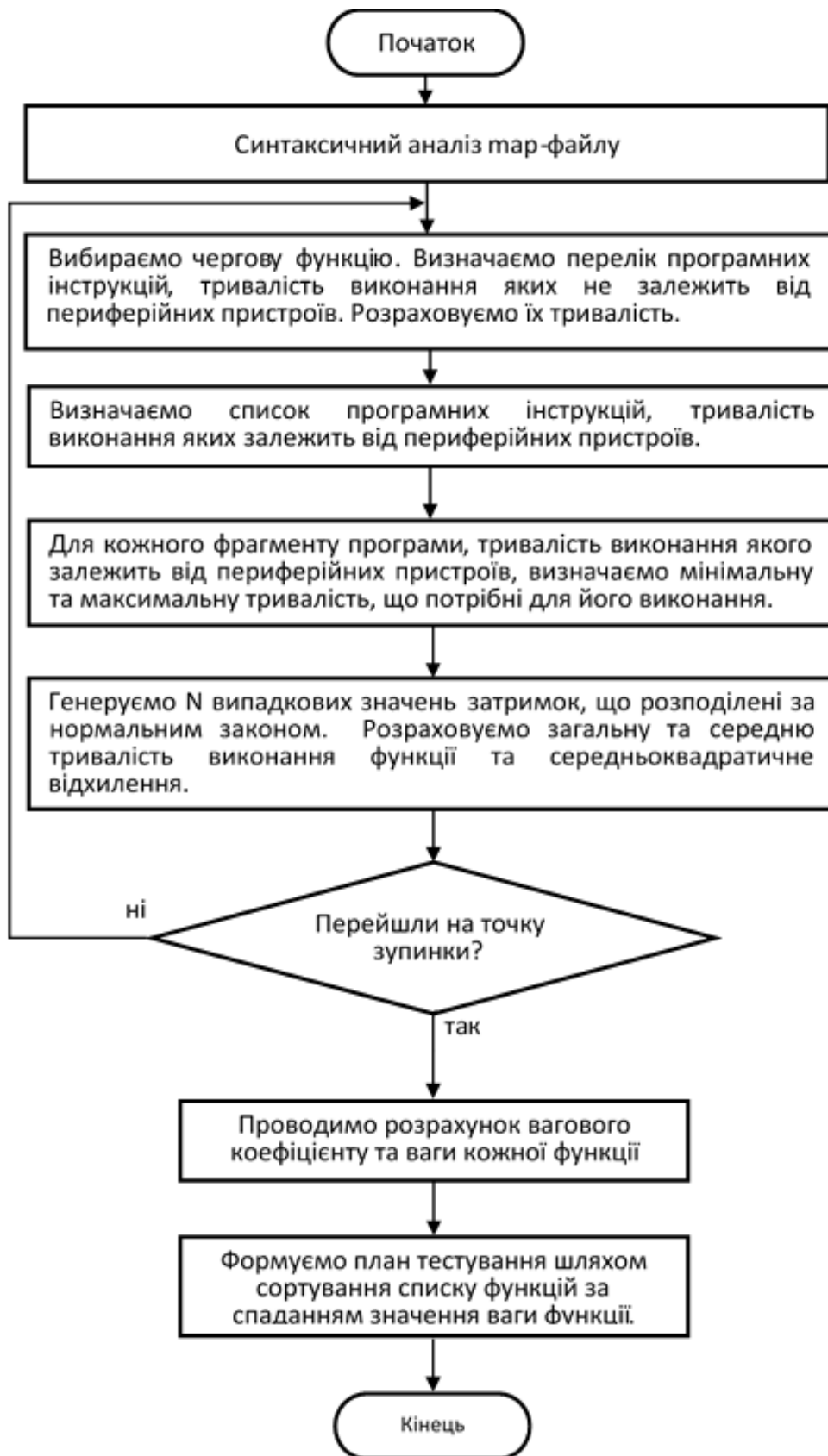


Рис. 4 – Блок-схема алгоритму формування плану тестування на підставі моделі функціонування ПЗ з урахуванням поведінки периферійних пристроїв

У четвертому розділі дисертації – “Прогнозування тривалості виконання програмного коду з урахуванням впливу зовнішніх та внутрішніх чинників” – для врахування впливу зовнішніх та внутрішніх чинників на апаратне забезпечення вбудованих систем запропоновано використовувати граф потоку керування у якості моделі функціонування ПЗ (рис. 5).

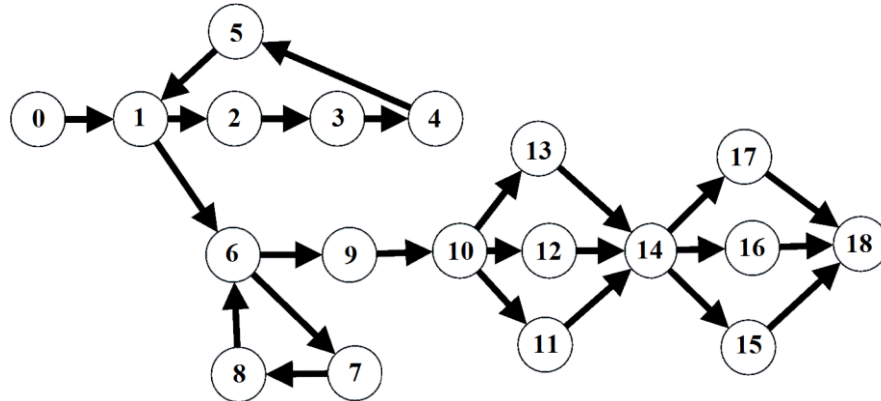


Рис. 5 – Приклад графу потоку керування, що представляє певну програмну функцію

Згідно з такою моделлю тривалість виконання певної гілки програми прийнято розраховувати, як:

$$\tau_b = \sum_{i=1}^n \tau_i, \quad (6)$$

де:  $\tau_i$  – тривалість виконання інструкції програмного коду, що подана у вигляді вершини графу;  $n$  – загальна кількість вершин у гілці програми.

Своєю чергою, тривалість виконання програмного коду з урахування впливу зовнішнього чи внутрішнього чинника на апаратне забезпечення вбудованої системи може бути обчислена за формулою:

$$\tau_b = \sum_{j=1}^k \tau_j + \sum_{i=1}^m \tau_i(param), \quad (7)$$

де:  $\tau_j$  – тривалість виконання програмної інструкції, що характеризується постійним значенням тривалості виконання;  $k$  – кількість програмних інструкцій із константною тривалістю виконання;  $\tau_i$  – тривалість виконання програмної інструкції, яка залежить від часу відгуку периферійного пристрою та змінюється під дією на нього певного чинника  $param$ ;  $m$  – кількість програмних інструкцій тривалість виконання яких залежить від часу відгуку периферійного пристрою.

Запропонований метод аналізу тривалості виконання програмного коду з урахуванням впливу зовнішніх і внутрішніх чинників на апаратне забезпечення вбудованих систем, який складається з таких основних етапів:

1. Синтаксичний аналіз тексту програми, що у свою чергу має ряд підзадач:
  - визначення фрагментів програми, тривалість виконання яких залежить від часу відгуку периферійних пристроїв;
  - визначення фрагментів програми, тривалість виконання яких не залежить від часу відгуку периферійних пристроїв, та здійснення розрахунку тривалості їх виконання.

2. Пошук та аналіз інформації про залежність часу відгуку периферійного пристрою від зовнішніх або внутрішніх чинників.
3. Вибір математичної моделі апроксимації.
4. Прогнозування тривалості виконання програмного коду.

Використовуючи розроблене програмного забезпечення, проведено прогнозування тривалості виконання програмного коду з урахуванням старіння компонентів (рис. 6).

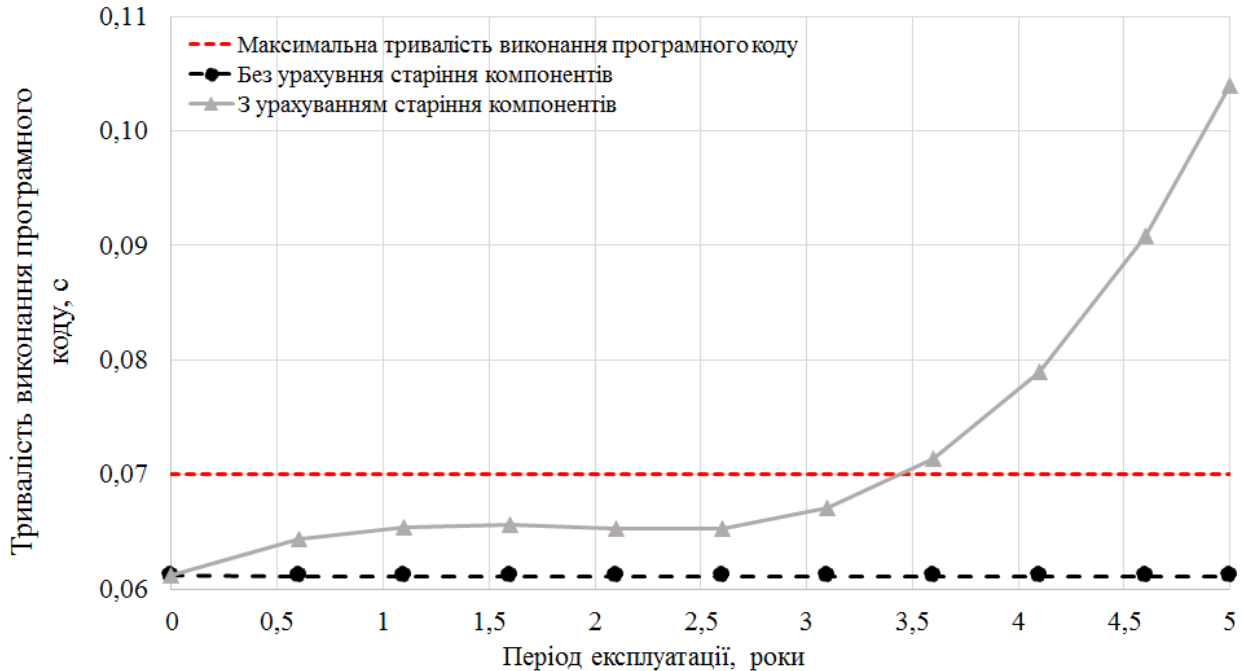


Рис. 6 – Тривалість виконання програмного коду з урахування та без урахування старіння компонентів

У п'ятому розділі дисертації – “**Архітектура програмного засобу для автоматизованого тестування тривалості виконання програмного коду та її практичне застосування**” – подано опис та архітектуру (рис. 7) розробленого ПЗ для автоматизованого тестування тривалості виконання програмного коду.

У розробленому ПЗ реалізовано такі основні можливості:

- 1) синтаксичний аналіз тексту програми вбудованих систем;
- 2) формування плану тестування з урахуванням поведінки периферійних пристроїв вбудованих системи;
- 3) формування плану тестування без урахування поведінки периферійних пристроїв вбудованих системи;
- 4) тестування тривалості виконання програмного коду вбудованих систем;
- 5) прогнозування тривалості виконання програмного коду під час дії зовнішніх та внутрішніх чинників на компоненти вбудованих систем;
- 6) візуалізація та збереження результатів аналізу.

Для розроблення програмного забезпечення використано такі технології та засоби: мову програмування C#, платформу .NET, технологію WPF (Windows Presentation Foundation) для реалізації графічного інтерфейсу, бібліотеки Math.Net, NGraphics, SkiaSharp та  $\mu$ Vision Socket API для роботи з математичними виразами та графами, а також їх графічним представленням.

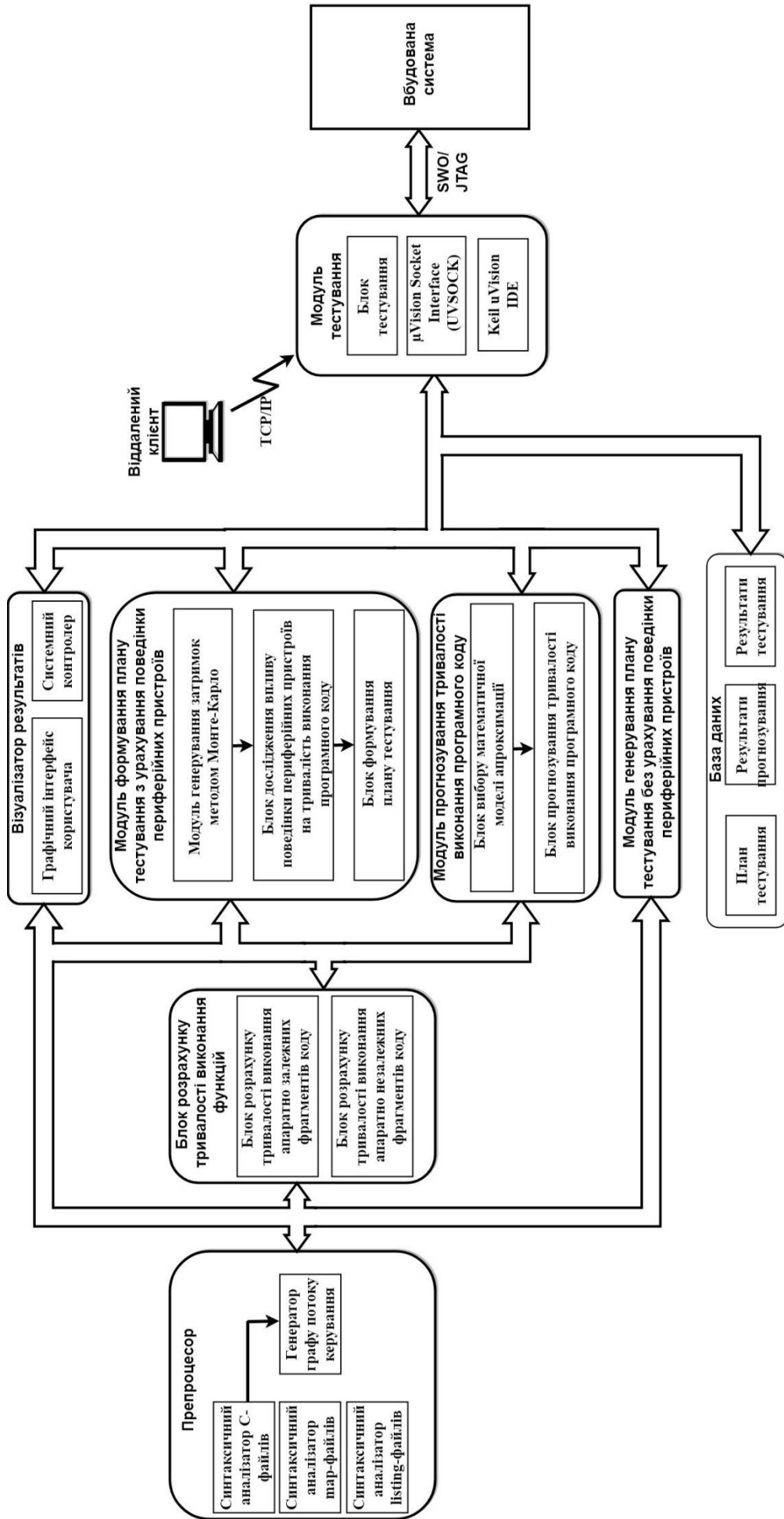


Рис. 7 – Архітектура програмного забезпечення «EXTT»



Для порівняння розробленого ПЗ EХТТ та RapіTime проведено тестування п'яти вбудованих систем Італійської компанії Dinamica Generale, а саме: Tomato Sniper, Smart Control, Pump Logger, Field Scale та DG8000-IC. Порівняння отриманих результатів проводилось за двома ключовими критеріями: точність отриманих результатів та час витрачений на тестування (рис. 8).

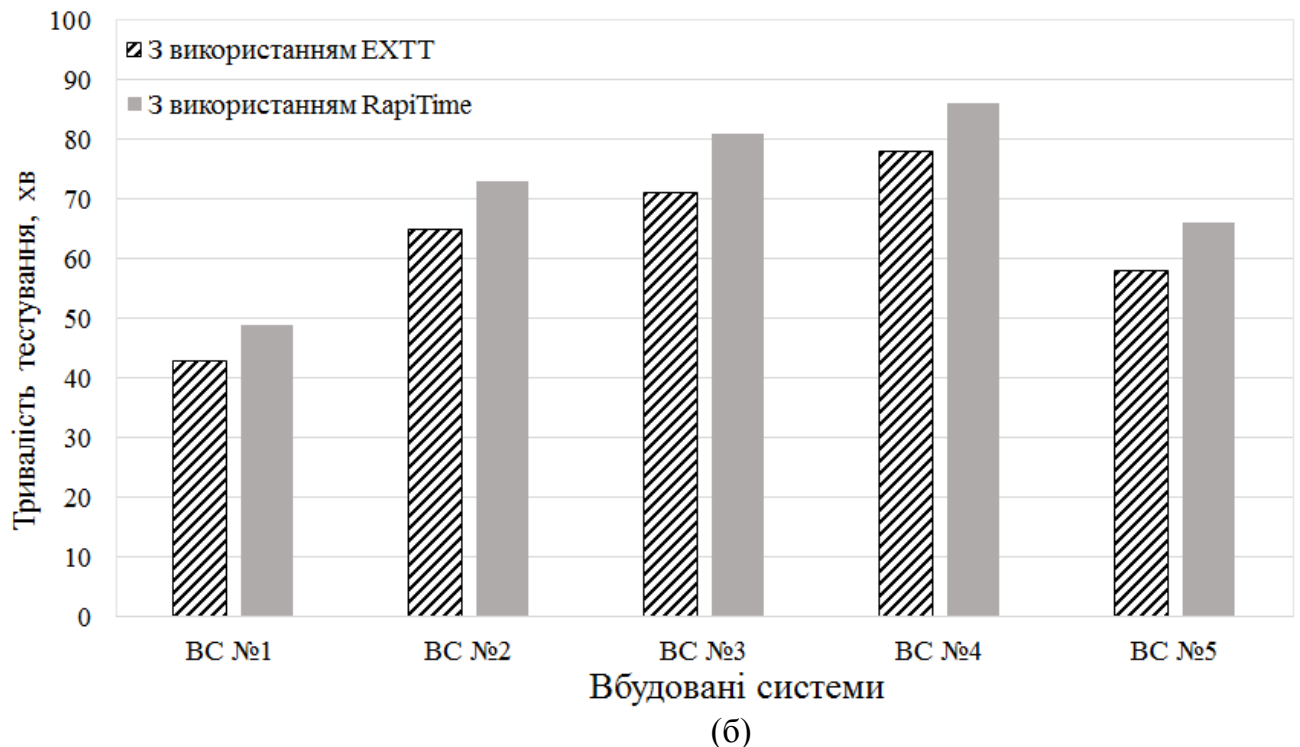
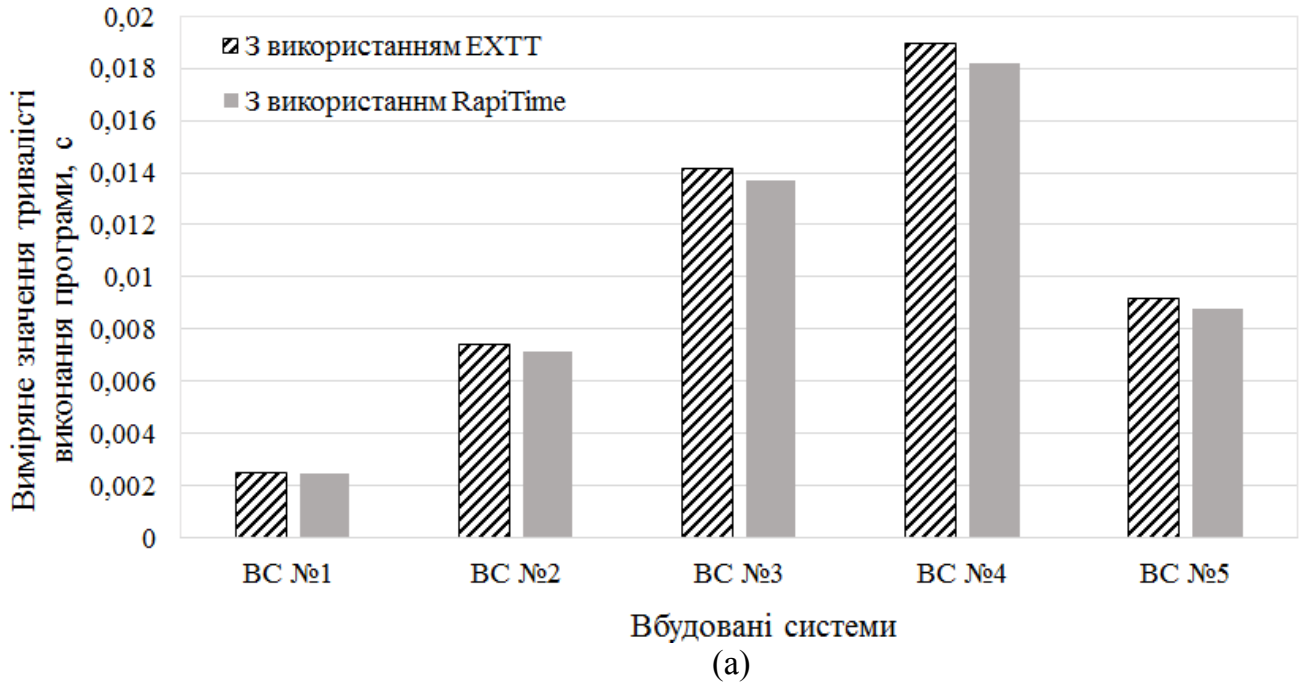


Рис. 8 – Результати порівняння розробленого ПЗ EХТТ та RapіTime:  
(а) – порівняння точності отриманих результатів; (б) – порівняння часу тестування

Проведені дослідження показали, що точність отриманих результатів під час використанні ПЗ EХТТ підвищилася на 3,8 %. Крім цього, ПЗ EХТТ дає змогу зменшити час витрачений на тестування тривалості виконання програмного коду

внаслідок застосування методу генерування плану тестування, який враховує поведінку периферійних пристроїв.

Можливість проведення прогнозування тривалості виконання програмного коду при дії на вбудовану систему зовнішніх і внутрішніх чинників значно підвищує рівень адекватності отриманих результатів та дає змогу розширити напрями застосування розробленого ПЗ.

## ВИСНОВКИ

У дисертаційній роботі розв'язано актуальне наукове завдання – розроблення методів тестування тривалості виконання програмного коду вбудованих систем реального часу і створення відповідного програмного засобу для автоматизації процесу їх тестування. Отримано такі основні наукові та практичні результати:

1. На підставі аналізу проблеми тестування програмного забезпечення мікроконтролерних вбудованих систем реального часу обґрунтовано необхідність удосконалення наявних методів та засобів тестування тривалості виконання програмного коду.

2. Удосконалено метод статичного аналізу тривалості виконання програмного коду для мікроконтролерних вбудованих систем, а саме запропоновано та обґрунтовано метод аналізу втрат швидкості надсилання даних через зовнішні інтерфейси зв'язку та між внутрішніми модулями мікроконтролера. Це стало можливим завдяки статичному аналізу низькорівневих функцій, які виконують налаштування мікроконтролера під час його ввімкнення.

3. Отримав подальший розвиток метод динамічного тестування тривалості виконання програмного коду, який, на відміну від наявних, забезпечує повний контроль над вбудованою системою без внесення змін у її програмне чи апаратне забезпечення, що дає можливість підвищити точність отриманих результатів на 3-5%.

4. Отримав подальший розвиток метод формування плану тестування програмних функцій вбудованої системи, який, на відміну від наявних, враховує поведінку периферійних пристроїв, що дає можливість більш ефективно тестувати вбудовані системи. Експериментально встановлено, що використання плану тестування, який враховує поведінку периферійних пристроїв, забезпечує зменшення витрат часу на тестування на 11-15%.

5. Розроблено метод прогнозування тривалості виконання програмного коду, який, на відміну від наявних, враховує вплив зовнішніх та внутрішніх чинників на апаратне забезпечення вбудованих систем та тривалість виконання їхнього програмного коду.

6. Розроблено програмний засіб для тестування тривалості виконання програмного коду вбудованих систем реального часу, що дає змогу автоматизувати процес тестування вбудованих систем, а також здійснювати розрахунок періодичності технічного обслуговування вбудованих систем, що забезпечить заданий рівень надійності та безпечності їх експлуатації.

7. Результати дисертаційної роботи впроваджено в процес тестування вбудованих систем у компанії Dinamica Generale S.p.A, ПП НВПІІ “Спаринг-Віст Центр” та в навчальний процес Національного університету «Львівська політехніка».

## СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

*Статті у наукових фахових виданнях України*

1. Федасюк Д.В., Чопей Р.С. Алгоритм побудови графу потоку керування за текстом програми мовою С // *Радіоелектроніка, інформатика, управління: науковий журнал*. Запоріжжя: Запорізький національний технічний університет, 2018. № 2. С. 154-161. (Thomson Reuters Web of Science; Google Scholar; eLIBRARY.RU; Index Copernicus).

2. Чопей Р.С., Федасюк Д.В. Огляд підходів до аналізу тривалості виконання програмного коду // *Електротехнічні та комп'ютерні системи: науково-технічний журнал*. Одеса: Одеський національний політехнічний університет, 2017. № 26 (102). С. 68-78. (eLIBRARY.RU; Index Copernicus).

3. Чопей Р.С., Федасюк Д.В. Метод аналізу тривалості виконання програмного коду з урахуванням архітектури мікроконтролера // *Вісник Вінницького політехнічного інституту*. Вінниця, 2018. № 2 (137). С. 74-79. (Index Copernicus; РИНЦ).

*Стаття в науковому періодичному виданні України, що включено до міжнародної наукометричної бази Scopus*

4. Fedasyuk D., Marusenkova T., Chohey R. A method of predicting the maintenance period of embedded systems for preventing breach of their time requirements // *International Journal of Computing, Ukraine*. 2018. Vol. 17, Is. 2. P. 94-101. (Scopus; Google Scholar; Index Copernicus).

*Статті в наукових періодичних виданнях інших держав*

5. Fedasyuk D., Marusenkova T., Chohey R. A model for estimating firmware execution time taking into account peripheral behavior // *International Journal of Intelligent Systems and Applications, Hong Kong*. 2018. Vol. 10, No. 6. P. 22-29. (Scopus; Google Scholar; Microsoft Academic Search; Stanford University Libraries).

6. Fedasyuk D., Gavrysh V., Marusenkova T., Chohey R. Determining the influence of the external factors on the firmware response time // *Central European Researchers Journal, Slovakia*. 2018. Vol. 4, Is. 1. P. 1-8.

*Статті у збірниках праць міжнародних конференцій, які включено до міжнародних наукометричних баз*

7. Fedasyuk D., Chohey R., Knysh B. Architecture of a tool for automated testing the worst-case execution time of real-time embedded systems' firmware // *IEEE Proceedings of the XIVth International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics" (CADSM'2017)*, February 21–25, 2017. Lviv, 2017. P. 278-282. (Scopus).

8. Chohey R., Fedasyuk D., Marusenkova T. A model for estimating firmware execution time based on peripherals' time behavior // *Proceedings of the XIVth IEEE International Conference "Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering" (TCSET'2018)*, February 20–24, 2018. Lviv, 2018. P. 1179-1183. (Scopus).

*Публікації у матеріалах міжнародних та всеукраїнських конференцій*

9. Chohey R., Knysh B., Fedasyuk D. The model of software execution time remote testing // *Proceedings of the IXth International Conference of Young Scientists "Computer Science and Engineering – 2017" (CSE'2017)*, November 22–24, 2017. Lviv, 2017. P. 398-402.

10. Чопей Р.С., Федасюк Д.В. Метод побудови засобу для автоматизованого тестування тривалості виконання програмного коду вбудованих систем розроблених з використанням KEIL UVISION // Електротехнічні та комп'ютерні системи: Теорія та практика (ЕЛТЕКС – 2018) : Матеріали Міжнародної науково-практичної конференції, 29 травня – 1 червня 2018, Одеса. Одеса, 2018. С. 34-40.

11. Чопей Р.С., Федасюк Д.В. Проблеми тестування вбудованих систем жорсткого реального часу // Математичне і програмне забезпечення інтелектуальних систем (MPZIS-2016): Матеріали XIV Міжнародної науково-практичної конференції, 16-18 листопада 2016, Дніпро. Дніпро, 2016. С. 243-245.

12. Чопей Р.С., Федасюк Д.В. Тестування тривалості виконання програмного коду мікроконтролерних систем реального часу // Математичне і програмне забезпечення інтелектуальних систем (MPZIS-2017) : Матеріали XV Міжнародної науково-практичної конференції, 22–24 листопада 2017, Дніпро. Дніпро, 2017. С. 216-218.

## АНОТАЦІЯ

**Чопей Р.С. Засоби автоматизованого тестування спеціалізованого програмного забезпечення вбудованих систем.** – На правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем. – Національний університет «Львівська політехніка» Міністерства освіти і науки України, Львів, 2019.

У дисертаційній роботі розв'язано актуальне наукове завдання удосконалення наявних методів тестування тривалості виконання програмного коду вбудованих систем реального часу й розроблення відповідного програмного засобу для автоматизації процесу тестування тривалості виконання програмного коду вбудованих систем реального часу.

Розроблене математичне забезпечення включає в себе: метод динамічного аналізу тривалості виконання програмного коду, методу віддаленого аналізу тривалості виконання програмного коду вбудованих систем, метод статичного аналізу тривалості виконання програмного коду для мікроконтролерних вбудованих систем, метод формування плану тестування на підставі моделі функціонування програмного забезпечення з урахуванням поведінки периферійних пристроїв, метод прогнозування тривалості виконання програмного коду з урахуванням впливу зовнішніх та внутрішніх факторів. На підставі цих методів розроблено алгоритми роботи програмних модулів “Аналізу програмного коду”, “Формування плану тестування”, “Тестування тривалості виконання програмного коду” та “Прогнозування тривалості виконання програмного коду”, що були об'єднані в програмний засіб для автоматизованого тестування тривалості виконання програмного коду ЕХТТ.

*Ключові слова:* спеціалізоване програмне забезпечення, автоматизоване тестування, тестування вбудованої системи, вбудована система реального часу, тривалість виконання програмного коду.

**Чопей Р.С. Средства автоматизированного тестирования специализированного программного обеспечения встроенных систем.** – На правах рукописи.

Диссертация на соискание ученой степени кандидата технических наук по специальности 01.05.03 – математическое и программное обеспечение вычислительных машин и систем. – Национальный университет «Львівська політехніка» Министерства образования и науки Украины, Львов, 2019.

В диссертационной работе решено актуальное научное задание усовершенствования существующих методов тестирования продолжительности выполнения программного кода встраиваемых систем реального времени и разработки соответствующего программного средства для автоматизации процесса тестирования продолжительности выполнения программного кода встраиваемых систем реального времени.

Разработанное математическое обеспечение включает в себя: метод динамического анализа продолжительности выполнения программного кода, метода удаленного анализа продолжительности выполнения программного кода встраиваемых систем, метод статического анализа продолжительности выполнения программного кода для микроконтроллерных встраиваемых систем, метод формирования плана тестирования на основе модели функционирования программного обеспечения с учетом поведения периферийных устройств, метод прогнозирования продолжительности выполнения программного кода с учетом влияния внешних и внутренних факторов. На основании этих методов разработаны алгоритмы работы программных модулей: “Анализ программного кода”, “Формирование плана тестирования”, “Тестирование продолжительности выполнения программного кода” и “Прогнозирование продолжительности выполнения программного кода”, которые были объединены в программное средство для автоматизированного тестирования продолжительности выполнения программного кода ЕХТТ.

*Ключевые слова:* специализированное программное обеспечение, автоматизированное тестирование, тестирование встроенной системы, встроенная система реального времени, продолжительность выполнения программного кода.

**Chopey R.S. Means of automated testing of specialized software of embedded systems.** – On the right of manuscript.

Thesis for Ph.D degree on technical sciences in specialty 01.05.03 – “Mathematical and software support of computer machines and systems”. – Lviv Polytechnic National University, Ministry of Education and Science of Ukraine, Lviv, 2019.

The scientific task of improving the existing methods of execution time testing of the real time embedded system software for automation of the process of testing and increasing the efficiency of this process has been solved in the dissertation work.

The study is relevant due to the necessity of modern methods for automated execution time testing against the backdrop of growing requirements for reducing the testing time of embedded systems while ensuring their reliability and safety. Therefore, the method of predicting the software execution time was developed, that takes into account the influence of external and internal factors. It allows to increase the adequacy of the results obtained. The method of dynamic testing of the embedded system software execution time were improved, which, in contrast to the existing ones, does not affect the software execution time. This gives an opportunity to increase the accuracy of the obtained results by 3-5%. Improvement of this method consisted not only in increasing the accuracy of the measurement, but also in adapting this method for its using in embedded systems that based on microcontrollers with the ARM architecture. It is these microcontrollers that are used in 37% of embedded systems currently available. In addition, the coverage of ARM architecture will ensure the relevance of the developed method for the next 5 years, and thus ensure a steady increase in the number of its potential users.

We have developed the method of forming a plan for testing the program functions of the embedded system. This method, in contrast to the existing one, takes into account the behavior of peripherals of embedded systems, which makes it possible to create a test plan that reduces the time spent on testing up to 15% and thus to raise the effectiveness of the testing process. Therefore, the task of increasing the efficiency of testing the execution time of the embedded system software by increasing the accuracy of the obtained results and reducing the time spent on testing process, as well as the increasing of the level of automation the testing process was solved.

The software tool EXTT for automated testing the embedded software execution time was created. EXTT consist of software modules that realized the developed algorithms: "Static firmware analysis", "Creating the test plan", "Test firmware execution time" and "Predicting the firmware execution time". EXTT software tool can be easily integrated into the manufacturing processes of companies involved in the development or testing of embedded systems.

*Keywords:* specialized software, automated testing, testing embedded system, embedded real-time system, execution time analysis.

Підписано до друку 25.02.2019 р.  
Формат 60×84/16. Папір офсетний.  
Друк цифровий. Умовн. друк. арк. 0,9.  
Наклад 100 прим. Зам. № 26

ТзОВ «Растр-7»  
79005, м. Львів, вул. Кн. Романа, 9/1  
тел./факс: (032) 235-52-05  
Свідоцтво суб'єкта видавничої справи:  
ЛВ №22 від 19.11.2002 р.