

МОНІТОРИНГ ПАРАМЕТРІВ МІКРОТРОНА М-30

Ю. Плакош, В. Ковтун

Закарпатський державний університет
кафедра інформаційних управляючих систем і технологій
вул. Заньковецької, 89 а, м. Ужгород, 88000
e-mail: yuri.plakosh@gmail.com

Створено імітаційну модель монітора параметрів мікротрона М-30, що дозволяє проводити моніторинг та контроль його блоків та обробку статистичних параметрів. Розроблено схему спряження з функціональними блоками мікротрона М-30 з використанням модулів АЦП WAD-AIK-BUS. Також реалізовано графічний інтерфейс та бізнес-логіку монітора параметрів. Програмна частина містить сукупність графічних панелей, які в реальному часі відображають значення параметрів мікротрона М-30 або функції від декількох параметрів. Використовувалася мова програмування Java (версія JDK 1.6). Також було використано такі технології та розробки як Swing, Java Communications API, JFreeChart, JUnit, Log4j і Spring Framework.

Вступ

Пульт управління ядерно-фізичною установкою ІЕФ НАН України представляє собою складну інтерактивну та багатофакторну систему. Його модернізація потребує розробки та створення моделі панелі управління мікротрона М-30. Реалізація такої моделі на основі персонального комп'ютера дозволяє модернізувати робоче місце оператора установки та автоматизувати збір, накопичення і обробку даних вимірювань. При цьому ефективним методом дослідження таких складних систем є імітаційне моделювання. Особливий інтерес представляє таке дослідження для багатофакторних задач, коли потрібно врахувати наявність кореляційних зв'язків між параметрами системи та стохастичний характер їх функціонування.

Мікротрон М-30 представляє собою циклічний електронний прискорювач зі змінною кратністю прискорення. Він був змонтований в Ужгороді на початку 70-х років з використанням базових вузлів, отриманих з Об'єднаного інституту ядер-

них досліджень (м. Дубна, Московської обл.). Широкий інтервал перебудови енергії (1-30 МеВ) та висока моноенергетичність випромінювання (0,02%) привабливі для численних застосувань М-30 в галузі фундаментальних (дослідження будови ядра та ядерних реакцій, фізики запізнюючих нейтронів, вивчення фотоядерних реакцій, збудження ізомерних станів в області енергій 1-30 МеВ) та прикладних ядерно-фізичних досліджень, зокрема, радіаційній фізиці, ядерній медицині та екології довкілля. На базі мікротрона М-30 створено радіаційний стенд для дослідження радіаційної стійкості матеріалів та приладів в умовах дії потужних пучків нейтронів, гамма-випромінювання та швидких електронів.

Дана робота представляє результати розробки структурної моделі програмного забезпечення пульта управління М-30, підбір апаратних засобів та стратегію конструювання програмних модулів.

Моделювання мікротрона М-30 передбачає встановлення його цільової функції, в ролі якої виступає струм приско-

рених електронів, структурування його елементів, блоків (стан вакууму у прискорювальній камері, катода, магнітного поля, ефективності захоплення НВЧ-поля модулятора, тощо) та встановлення зв'язків між ними.

Програмний комплекс

В роботі представлені загальні вимоги та структура системи управління та моніторингу параметрів прискорювача електронів мікротрона М-30 ІЕФ НАН України, реалізованої на основі персонального комп'ютера. Ця розробка дозволяє модернізувати робоче місце оператора та автоматизувати збір, накопичення і обробку даних вимірювань. Розробка базується на інтерактивній та багатofакторній моделі пульта управління М-30, що враховує кореляційний зв'язок між її параметрами. Для цього створена структурна модель програмного забезпечення пульта управління, підібрано апаратні засоби та запропоновано стратегію конструювання програмних модулів.

Реалізовано графічний інтерфейс та бізнес-логіку, а також схему спряження з функціональними блоками мікротрона М-30 з використанням модулів АЦП WAD-AIK-BUS [1], гальванічно розв'язані вхідні канали яких дозволяють реалізувати систему моніторингу параметрів прискорювача з достатньою достовірністю та надійністю. Програмна частина панелі управління містить сукупність віртуальних панелей, кожна з яких відображає в реальному часі значення параметра М-30 або функції від декількох параметрів. Також існує кілька спеціалізованих панелей, які виконують спеціалізовану обробку параметрів. Кількість датчиків, що обробляє програмний комплекс, залежить від кількості АЦП, що з'єднані в одну шину, і обмежена 127 модулями, кожен з яких має 4 канали для вхідних сигналів.

Програмна частина розробки є універсальною, оскільки вона дозволяє проводити моніторинг параметрів прак-

тично будь-якого пристрою (установки), для якого побудовано схему спряження з його функціональними блоками. Також на основі універсальної програмної частини можна побудувати спеціалізовану обробку параметрів у вигляді окремих панелей та модулів.

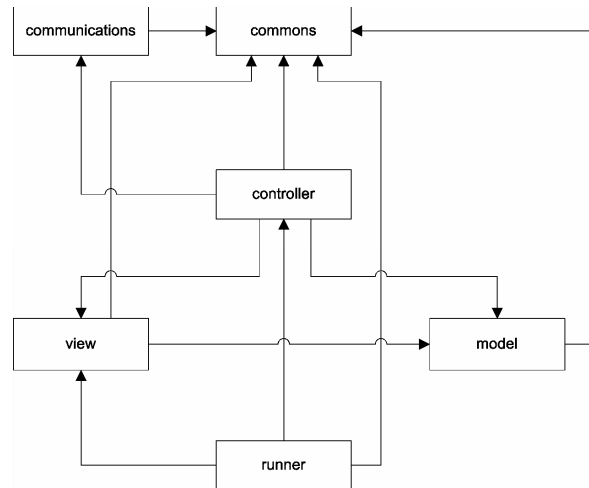


Рис.1. Структурна схема залежностей між модулями програмного комплексу

Програмна система складається з таких модулів (Рис.1):

- **commons.** Цей модуль містить код, що є загальним для всіх модулів.
- **communications.** Цей модуль реалізує частину протоколу ModBus, що дозволяє взаємодіяти з мікротроном М-30 через модулі АЦП WAD-AIK-BUS.
- **controller.** Реалізує бізнес-логіку – правила зчитування і обробки значень параметрів мікротрона М-30 та їх виведення.
- **model.** Містить класи обміну даними між модулями програмної частини.
- **view.** Цей модуль включає код, що дозволяє представити інформацію користувачеві у потрібній формі: графічна (у вигляді діаграми), звукова (подача звукового сигналу на динаміки ПК у разі виходу значення хоча б одного параметра за допустимі межі), текстова (запис

значень кожного з параметрів у текстовий файл).

- `runner`. Здійснює запуск системи в одному з двох режимів: тестовому (коли дані з АЦП не зчитуються, а беруться від генератора випадкових чисел) і реальному.

В даному проекті за основу розділення модулів брався шаблон `model-view-controller (MVC)` (Рис.2). Цей шаблон застосовується для відділення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли проводитися без зміни інтерфейсу користувача. Контролер інтерпретує введені дані та інформує модель і вигляд про необхідність відповідної реакції. Окрім даного використовувались також шаблони адаптер (`adapter`), спостерігач (`observer`), стратегія (`strategy`), інверсія контролю (`inversion of control`), фасад (`facade`). [2]

При розробці використовувалася мова програмування Java (версія `JDK 1.6`) [3]. Також було використано такі технології та розробки як `Swing`, `Java Communications API`, `JFreeChart`, `JUnit`, `Log4j` і `Spring`.

Програми на Java транслуються в байт-код, що виконується на віртуальній Java-машині (`JVM`) – програмі, що обробляє байт-код і передає інструкції устаткуванню, як інтерпретатор, але з тією відмінністю, що байт-код, на відміну від тексту, обробляється значно швидше.

Перевага подібного способу виконання програм – в повній незалежності байт-коду від ОС і устаткування, що дозволяє виконувати Java програми на будь-якому пристрої, який підтримує віртуальну машину. Іншою важливою особливістю технології Java є гнучка система безпеки, завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне пере-

ривання. Це дозволяє користувачам завантажувати програми, написані на Java, на їх комп'ютери (або інші пристрої, наприклад, мобільні телефони) з невідомих джерел, при цьому не побоюючись зараження вірусами, пропажі цінної інформації тощо.

Часто до недоліків цього підходу відносять те, що виконання байт-коду віртуальною машиною може знижувати продуктивність програм і алгоритмів, реалізованих на мові Java. Дане твердження справедливе для перших версій віртуальної машини Java, проте останнім часом воно практично втратило актуальність. Цьому сприяв ряд удосконалень: застосування технології `JITs (Just-In-Time compiler)`, що дозволяє переводити байт-код в машинний код під час виконання програми з можливістю збереження версій класу в машинному коді, широке використання `native-коду` в стандартних бібліотеках, а також апаратні засоби, що забезпечують прискорену обробку байт-коду (наприклад, технологія `Jazelle`, підтримувана деякими процесорами фірми `ARM`).

Останнім релізом є версія 1.6, в якій було вироблено поліпшення системи безпеки, поліпшення підтримки `XML` і скриптової мови `Mozilla Rhino`, додані деякі нові можливості в створенні графічних інтерфейсів.

Основні можливості:

1. Розширені можливості обробки виняткових ситуацій.
2. Багатий набір засобів фільтрації введення/виведення.
3. Набір стандартних колекцій, таких як масив, список, стек і т. п.
4. Наявність простих засобів створення мережових додатків (зокрема з використанням протоколу `RMI`).
5. Наявність класів, що дозволяють виконувати `HTTP-запити` і обробляти відповіді.
6. Вбудовані в мову засоби створення багатопотокових додатків.
7. Уніфікований доступ до баз даних на основі `JDBC` і `SQLJ`.

8. Підтримка шаблонів (починаючи з версії 1.5).

Переваги мови Java:

1. Підтримка принципу безпеки типів.
2. Компіляція вихідного коду в байт-код.
3. Стандартизація вводу/виводу, графіки, геометрії, діалогу, доступу до баз даних тощо.

4. Можливість переносу і запуску програми на іншій платформі без перекомпіляції.

5. Наявність прибиральника сміття.
6. Повна об'єктно-орієнтованість без підтримки процедурного програмування.
7. Наявність інформації про тип даних під час виконання програми.

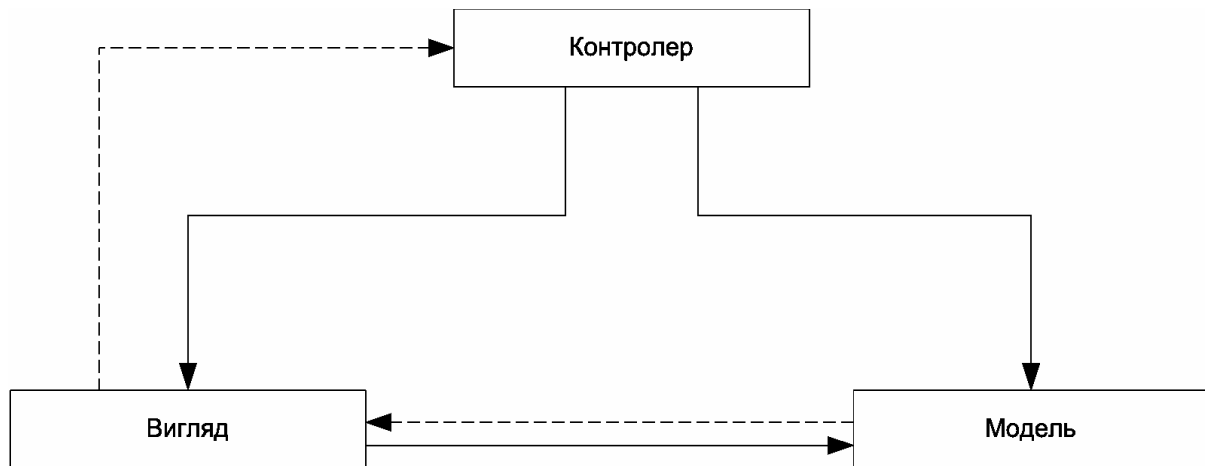


Рис. 2. Шаблон моделі (MVC).

Висновки

До переваг запропонованої розробки можна віднести: універсальність, автоматичне накопичення результатів вимірювань, можливість виводу значень функції від декількох параметрів, можливість розширення функціональності, автомати-

чна обробка результатів вимірювань, компактність, ергономічність, наявність персональних налаштувань оператора, звукова та візуальна сигналізація при виході параметра за допустимі межі.

Обговорюються можливості оптимізації управління М-30.

Література

1. Серия модулей УСО WAD-...-BUS, WAD-...-USB: Техническое описание WAD-AIK-BUS (USB) (Четырехканальный модуль аналогового ввода с поканальной гальванической развязкой входных каналов и интерфейсом USB или RS-485, предназначенный для построения распределенных систем сбора данных). – К., 2006. – 24 с.

2. Erich G., Helm R., Johnson R., and Vlissides J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.
3. Хорстманн К. С., Корнелл Г. Библиотека профессионала. Java 2. Том 1. Основы.: Пер. с англ. – М. : Издательский дом «Вильямс», 2004. – 848 с. : ил. – Парал. тит. англ.

4. Техническое описание модуля WAD-RS232/USB/RS485-BUS (Преобразователя интерфейсов с гальванической ра-

звязкой, предназначенного для построения распределенных систем сбора данных). – К., 2006. – 6 с.

MICROTRON M-30s' PARAMETERS MONITOR

Y. Plakosh, V. Kovtun

Transcarpathian State University
Information Managing Systems and Technologies Department
Zankovetskoyi Str., 89 a, Uzhgorod, 88000

The results of creation of the simulation model of the M-30 Microtron control panel, enabling in situ control of its blocks and statistical parameters processing, are presented. Graphic interface prototype, business logic as well as the interface circuit with the M-30 functional units using ADC WAD-AIK-BUS modules are implemented. The control panel software contains a set of virtual panels, each of them displaying the values of an M-30 parameter or a function of several parameters in real time. Java programming language (JDK 1.6) was used for the development as well as Swing, Java Communications API, JFreeChart, JUnit, Log4j, and Spring.

