

## ОПЫТ ИСПОЛЬЗОВАНИЯ FlexCtrl SCADA ДЛЯ АВТОМАТИЗАЦИИ УСКОРИТЕЛЕЙ

*А. С. Никифоров<sup>1</sup>, В. В. Алейников, К. П. Сычёв,  
И. В. Борина, А. А. Рукавишников*

Объединенный институт ядерных исследований, Дубна

В статье описывается программная часть автоматизации ускорителей на базе лицензионных программных средств (FlexCtrl SCADA, Photon Application Builder, Cogent DataHub) и разработанной дополнительно библиотеки классов, которые вместе представляют интегрированную среду для создания системы автоматизации, которая по функциональным возможностям относится к визуальному программированию. Количество свойств компонентов этой среды и количество самих компонентов может быть увеличено, так как среда разработки характеризуется своей открытостью.

Software of particle accelerator control system is described in this paper. Software is based on licensed software (FlexCtrl SCADA, Photon Application Builder, Cogent DataHub) and additionally developed class library. This software is integrated programming environment and according to its functionality serves as visual programming. Properties number of IDE components as well as components number can be increased due to the fact that IDE is characterized by its openness.

PACS: 29.20.-c

### ВВЕДЕНИЕ

В настоящее время для автоматизации промышленных процессов используются многочисленные программные пакеты SCADA (Supervisory Control And Data Acquisition). Работу ускорительного комплекса можно в некотором смысле рассматривать как промышленный процесс для производства ускоренных заряженных частиц. Поэтому есть смысл использовать SCADA для задачи автоматизации работы ускорителя. Автоматизированная система управления (АСУ) ускорителем должна соответствовать следующему набору требований:

- 1) высокая надежность;
- 2) большое количество информационных каналов с данными (порядка 10 тыс.);
- 3) минимальное количество разработчиков проекта;
- 4) сжатые сроки создания проекта;
- 5) дружественный интерфейс оператора;
- 6) высокая информативность для службы технической поддержки ускорителя;
- 7) поддержка разных типов пользователей;
- 8) обеспечение доступа к АСУ через Internet.

В свете этих требований особенно важно сделать правильный выбор не только SCADA, но и операционной системы.

---

<sup>1</sup>E-mail: nikiforov12345@mail.ru

## ОПЕРАЦИОННАЯ СИСТЕМА И АВТОМАТИЗАЦИЯ УСКОРИТЕЛЕЙ

Современные ускорители включают в себя множество подсистем, которые взаимодействуют между собой, обеспечивая получение пучка заряженных частиц для физических и экспериментальных установок. К подсистемам ускорителя можно отнести: вакуумную систему, систему водоохлаждения, управление магнитами и т. д. Компьютеры осуществляют обработку многих тысяч информационных каналов из подсистем ускорителя, отображение состояния ускорителя на мониторах и получение команд от операторов. Создание систем автоматизации ускорителя начинается с выбора операционной системы и дальнейшего выбора инструментальных программных средств, необходимых для разработки АСУ. К операционным системам, используемым в системах автоматизации, предъявляются определенные требования, которые относят их к операционным системам реального времени (ОСРВ). ОСРВ [1] используются в областях с повышенными требованиями надежности, отказоустойчивости, а именно:

1) ОСРВ должны обеспечивать высокую степень отказоустойчивости системы, чтобы при отказе какой-либо части программного обеспечения (ПО) другая часть продолжала нормально функционировать. ОСРВ должна гарантировать невозможность общего отказа системы;

2) ОСРВ должны удовлетворять жестким требованиям по качеству ПО, что подразумевает соответствие различным отраслевым, национальным и международным стандартам. Особенностью требований к ОСРВ является то, что ПО должно иметь доказанное качество;

3) требования по надежности: вероятность сбоя в ПО должна быть очень низкой;

4) требования по безопасности и секретности данных: в системе должны быть предусмотрены средства защиты наиболее важной информации.

Одним из стандартов, формально определяющих требования к ОСРВ, является POSIX («Portable Operating System interface for unIX»). Это переносимый интерфейс операционных систем на уровне исходных текстов (<http://www.pasc.org>), первое описание которого было опубликовано в 1986 г. Основная спецификация разработана как IEEE 1003.1 и одобрена как международный стандарт ISO/IEC 9945-1:1990. С точки зрения ОСРВ наибольший интерес представляют три стандарта: 1003.1a (OS Definition), 1003.1b (Realtime Extensions) и 1003.1c (Threads).

Из более чем сотни ОСРВ, разработанных в разное время, широкое распространение получили лишь около двадцати [2]. В настоящее время есть всего четыре сертифицированные ОСРВ:

VxWorks 6.4 (<http://www.windriver.com/products/vxworks/>);

Integrity ([http://get.posixcertified.ieee.org/select\\_product.tpl](http://get.posixcertified.ieee.org/select_product.tpl));

NEOS 3.0 (<http://www.mdstec.com/main/eng/?no=297>);

QNX Neutrino RTOS ([www.qnx.com](http://www.qnx.com)).

Данные о сертифицированных ОСРВ были опубликованы в официальном реестре IEEE. Тестирование проводилось по уровню *Conformance*.

Некоторые из ОСРВ построены на основе микроядра (QNX Neutrino, Integrity, VxWorks), а другие содержат монолитное ядро (LynxOS). Какой вариант ядра — монолитное или микроядро — наиболее соответствует ОСРВ? Спор по этому вопросу ведется давно. Надо признать, что окончательного ответа на этот вопрос пока нет. Исходя из того, что все гениальное должно быть просто. Архитектура операционной системы на

основе микроядра выглядит изящной и не громоздкой. Целесообразно для нашей задачи реального времени использовать ОСРВ, мы выбрали для системы автоматизации ускорителей сертифицированную ОСРВ–QNX Neutrino RTOS, в основе которой лежит микроядро, с учетом того, что эта система имеет следующие сертификаты:

**POSIX PSE52 Realtime Controller 1003.13-2003.** Этот сертификат подтверждает то, что у QNX Neutrino состав и функциональность API (интерфейса прикладного программирования), утилит и командного интерпретатора соответствуют требованиям профиля PSE52 (контроллер реального времени) стандарта IEEE 1003.13-2003.

**Common Criteria по уровню EAL4+.** Это сертификат соответствия требованиям стандарта по информационной безопасности ISO/IEC 15408.

**IEC 61508 по Safety Integrity Level 3 (SIL3)** — сертификат соответствия по требованиям функциональной безопасности. IEC 61508 — базовый стандарт для ответственных компьютерных систем, определяющий классическую V-модель разработки критичного программного обеспечения и являющийся основой для отраслевых стандартов.

## SCADA

АСУ ускорителя имеет дело с большим потоком информации, который содержит данные о текущем состоянии ускорителя. Для обработки и хранения подобных потоков информации используются программные пакеты SCADA. В настоящее время создано и развивается большое количество программ SCADA для разных операционных систем. Для задач автоматизации ускорителей выработан ряд требований к программным пакетам SCADA:

- 1) SCADA выполняет обработку данных порядка 10–20 тыс. информационных каналов;
- 2) SCADA синхронизирует доступ к данным для чтения и записи от разных пользовательских программ;
- 3) SCADA допускает подключение драйверов, обслуживающих нестандартное оборудование;
- 4) SCADA посылает сигналы пользовательским программам при изменении переменных базы данных;
- 5) SCADA реализует избыточное резервирование;
- 6) Для создания графического интерфейса в SCADA должен быть набор программных инструментов, которые могут обеспечить быстрое их создание, простое изменение функциональных возможностей элементов в окнах визуализации без привлечения к работе специалистов высокой квалификации. Одновременно должна быть возможность для квалифицированных пользователей увеличивать функциональные возможности имеющихся объектов визуализации, создавать новые объекты визуализации с новыми свойствами;
- 7) SCADA должна обеспечивать объектно-ориентированное взаимодействие с пользовательскими программами.

Известные нам SCADA имеют в своем составе как минимум три структурных блока, это

- база данных на определенное количество переменных;
- язык программирования для обработки переменных из базы данных;
- среда визуального программирования для создания графического интерфейса пользователя.

SCADA предлагает среду визуального программирования, в которой компоненты визуализации имеют *ограниченный* набор свойств. Пользователь SCADA не имеет возможности увеличить количество свойств компонентов визуализации, равно как создавать новые компоненты, которые не предусмотрены разработчиками SCADA. В связи с этими ограничениями мы не используем в SCADA-среде визуального программирования.

Нами была выбрана FlexCtrl SCADA фирмы BitCtrl (<http://www.bitctrl.de>). В настоящее время FlexCtrl SCADA используется в системах автоматизации ряда ускорителей в России и за рубежом: IC100, U400, U400M, DC60, CYLAB (<http://flerovlab.jinr.ru/flnr/main.html>).

### ПРОГРАММНЫЕ ИНСТРУМЕНТЫ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Для создания АСУ ускорителя мы используем следующие программные инструменты и программное обеспечение:

- операционную систему QNX Neutrino RTOS;
- систему разработки проектов и создания базы данных в FlexCtrl SCADA;
- компилятор языка ST для программ, которые выполняют обработку переменных, находящихся в базе данных FlexCtrl SCADA;
- OPC-серверы фирмы Cogent;
- среду визуального программирования Photon Application Builder (PhAB);
- компилятор языков программирования С и С++.

На языках программирования С и С++ нами создано специализированное программное обеспечение (СПО), которое объединяет в единое целое ресурсы вышеперечисленных программных инструментов и программного обеспечения (рис. 1) Это программное обеспечение совместно с программным обеспечением Cogent DataHub создает несколько OPC-серверов, которые содержат данные из базы данных FlexCtrl SCADA, и далее экспортирует их с разными правами доступа для разных пользователей локальной сети лаборатории и пользователей Интернет. Через эти OPC-серверы FlexCtrl SCADA, может взаимодействовать с другими программами SCADA, которые работают в MS Windows, например, LabView. СПО в своем составе имеет набор программных классов на языке С++, которые дают возможность использовать ресурсы PhAB в качестве среды

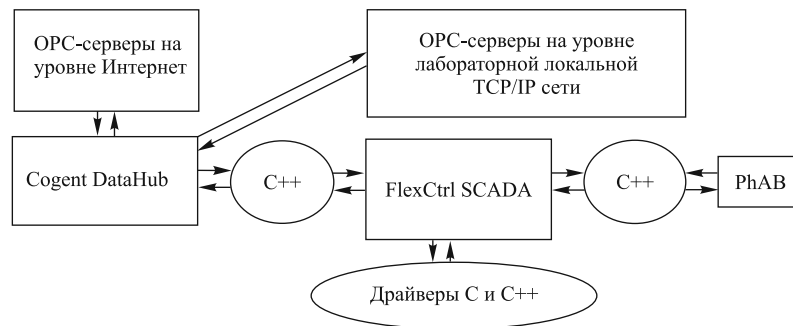


Рис. 1. Структура используемого программного обеспечения

визуального программирования для создания программ визуализации, взаимодействующих с базой данных FlexCtrl SCADA. Также с этой базой взаимодействуют программные драйверы, которые написаны на С и С++ для работы с аппаратурой на низком уровне.

Объединяемые программные инструменты и СПО дают возможность строить систему автоматизации ускорителя объектно-ориентированными методами. База данных FlexCtrl SCADA позволяет создавать пользовательские типы данных для описания определенных типов устройств, например шибера (рис. 2). Далее в SCADA создается набор переменных этого нового типа, количество которых равно количеству соответствующих устройств. Здесь же в SCADA задается алгоритм для работы с новым типом данных, который,

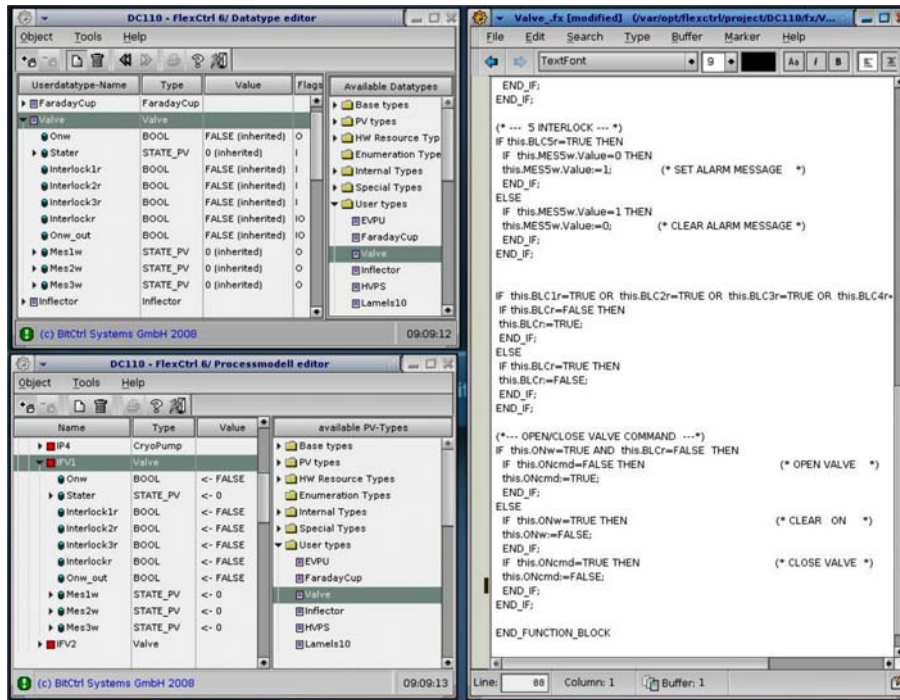


Рис. 2. Система разработки проектов в FlexCtrl SCADA

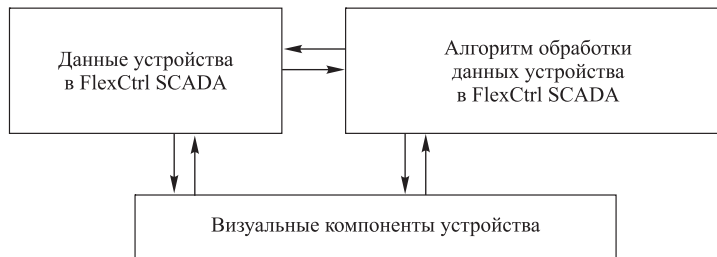


Рис. 3. Взаимодействие компонентов ПО

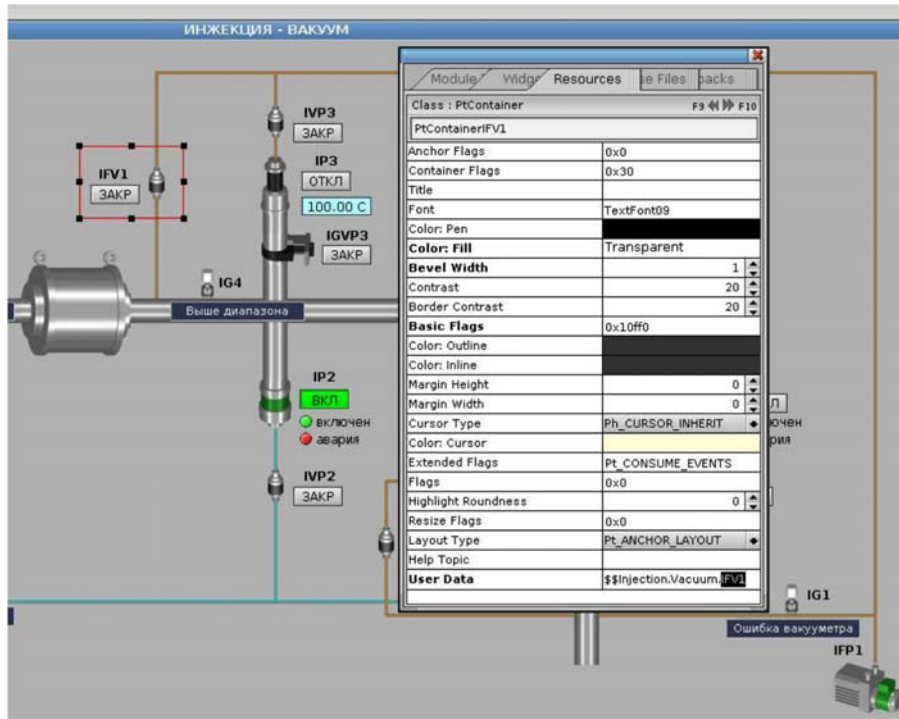


Рис. 4. Среда визуального программирования Photon Application Builder

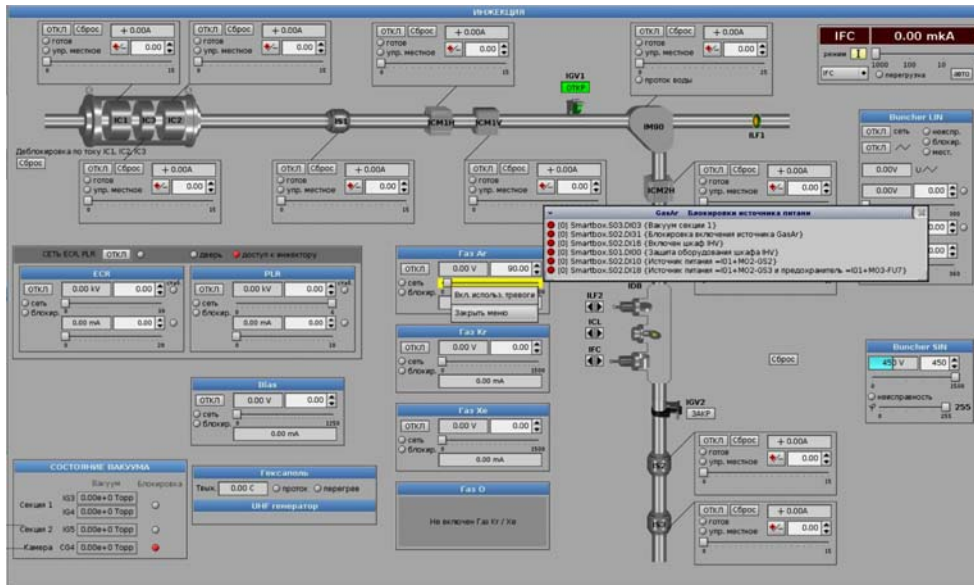


Рис. 5. Окно программы визуализации вакуумной системы ускорителя

в частности, управляет блокировками устройства. В задачи SCADA входит контроль поступления новых данных от устройства и исполнение соответствующего алгоритма обработки этих данных.

Классы СПО поддерживают, с одной стороны, связь с компонентами визуализации устройства, которые создаются средствами PhAB, а с другой — связь с данными устройства, которые находятся в базе FlexCtrl SCADA (рис. 3). Основная задача классов СПО — поддерживать соответствие компонентов визуализации устройств и базы данных FlexCtrl SCADA. Использование этих классов позволяет тиражировать однотипные устройства простым копированием в среде разработки PhAB (рис. 4). В этой среде разработки однотипные устройства различаются только именами. Классы СПО реализуют ряд дополнительных функциональных возможностей, которые требуются для операторов ускорителя и для службы технической поддержки. Например, для любого устройства в любой момент можно открыть окно с информацией о блокировках устройства (рис. 5). Если поступает сигнал тревоги от какого-либо устройства, то оператор получает графическую подсказку, на какую панель надо перейти, и на этой панели проблемное устройство выделяется мигающей подсветкой. Набор дополнительных функциональных возможностей классов СПО реализовывался с учетом пожеланий от пользователей АСУ.

Разработанное СПО многократно уменьшило рутинную работу для разработчиков систем автоматизации ускорителей, которая связана с прописыванием связей между переменными базы данных SCADA и графическими компонентами программы визуализации.

## ЗАКЛЮЧЕНИЕ

Создаваемые нами АСУ ускорителей имеют в среднем 5–15 тысяч информационных каналов. Разработанное специализированное программное обеспечение совместно с программным обеспечением сторонних производителей (QNX Neutrino RTOS, FlexCtrl SCADA, Photon Application Builder, Cogent DataHub) многократно уменьшило рутинную работу для разработчиков систем автоматизации ускорителей. Используемые нами программные и аппаратные средства не являются узкоспециализированными и могут с успехом применяться как для научной, так и для промышленной автоматизации.

## СПИСОК ЛИТЕРАТУРЫ

1. *Золотарёв С. В.* Взгляд на мир операционных систем реального времени в 2006 году. РТСофт. МКА 1/2006.
2. A Selection Methodology for the RTOS Market Philip Melanson, Siamak Tafazoli Canadian Space Agency Space Technologies Branch, Software and Ground Segment. Presented at Data Systems in Aerospace (DASIA 2003) Conf., Prague, Czech Republic, June 4, 2003.