

УДК 004.4'23

А. В. Шевченко [A. V. Shevchenko]

О. С. Мезенцева [O. S. Mezentseva]

Д. В. Мезенцев [D. V. Mezentsev]

**АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ
РОБОТИЗИРОВАННЫМИ СИСТЕМАМИ****ARCHITECTURE OF INFORMATION SYSTEM MANAGEMENT
OF ROBOTIC SYSTEMS**

ФГАОУ ВО «Северо-Кавказский федеральный университет», г. Ставрополь, Россия,
E-mail: luckyleo769@mail.ru

Аннотация. *Статья содержит анализ функционала существующих информационных систем управления роботизированными системами. Анализ архитектур Robotics Operation System, Player, Marie, Orocos и Microsoft Robotics Developer Studio показал, что ни одна из существующих систем не способна в полной мере удовлетворить всем требованиям, возникающим при разработке программного обеспечения робототехнических систем.*

Материалы и методы. *В статье рассмотрены программные решения и функционал разработанной информационной системы управления RoboStudio, архитектура которой основана на изученных системах и удовлетворяет комплексу предъявляемых требований: модульности, аппаратной независимости, избыточности, наличию графического интерфейса и эмуляции робототехнических систем. Для управления информационной системы был разработан язык rScript, интерпретируемый в реальном времени работы системы, посредством которого происходит управление всеми подключенными модулями и оборудованием, благодаря чему информационная система предоставляет большие возможности по управлению и программированию робототехнических систем.*

Результаты. *Графический пользовательский интерфейс позволяет сконцентрироваться на непосредственном решении задач, упрощает образовательный процесс, а эмулятор робототехнических систем позволяет отладить и апробировать разрабатываемые сценарии удаленно, без риска повреждения дорогостоящего оборудования.*

Экспериментальные исследования функционала системы проводились с использованием полноразмерного антропоморфного робота AR-601E и стенда «Мехатроника» производства НПО «Андроидная техника».

Заключение. *Архитектура и особенности программной реализации информационной системы управления роботизированными системами RoboStudio, позволяющей осуществлять полный и единовременный контроль за всеми электронными компонентами роботизированной системы, что повышает безопасность использования дорогостоящего оборудования, а также дает возможность гибкой настройки системы для работы с различным роботизированным оборудованием без изменений в исходном коде.*

Ключевые слова: информационная система, робототехника, ROS, система управления, андроидный робот.

Abstract. *The article contains an analysis of the functionality of existing information management systems of robotic systems. Analysis of the architectures of Robotics Operation System, Player, Marie, Orocos and Microsoft Robotics Developer Studio showed that none of the existing systems can fully meet all the requirements arising in the development of software for robotic systems.*

Materials and methods. *The article discusses software solutions and the functionality of the developed RoboStudio information management system, whose architecture is based on the systems studied and satisfies a set of requirements: modularity, hardware independence, redundancy, the presence of a graphical interface and emulation of robotic systems. To control the information system, rScript was developed, interpreted in real time of the system operation, through which all the connected modules and equipment are controlled, thanks to which the information system provides great opportunities for managing and programming robotic systems.*

Results. *The graphical user interface allows you to concentrate on solving problems directly, simplifies the educational process, and the emulator of robotic systems allows you to debug and test the developed scenarios remotely, without the risk of damaging expensive equipment.*

Experimental studies of the system functionality were carried out using the full-size anthropomorphic robot AR-601E and the Mechatronics stand produced by the Android Technological Scientific and Production Association.

Conclusion. *Architecture and features of the software implementation of the information management system of robotic systems RoboStudio, which allows for full and one-time control of all electronic components of the robotic system, which increases the safety of expensive equipment, as well as allows flexible configuration of the system to work with a variety of robotic equipment without changes in the source code.*

Key words: information system, robotics, ROS, control system, Android robot.

Введение. В настоящее время еще не сформировано каких-либо четких стандартов программирования робототехнических систем. Каждый производитель создает свою программную и аппаратную архитектуру исходя из собственных представлений об оптимизации процесса разработки. Крупные производители часто заказывают программные решения для своих робототехнических систем у сторонних разработчиков, мелкие создают программы своими силами. В итоге, программное обеспечение ориентировано только на конкретную платформу (чаще на конкретную модификацию робототехнической системы). Это приводит к тому, что зачастую при даже незначительной модернизации существующей роботизированной системы, приходится полностью перерабатывать программное обеспечение. Отсутствие универсальных программных продуктов влечет за собой огромные затраты времени и средств разработчиков, и нередко приводит к сворачиванию перспективных проектов.

Некоторые разработчики программного обеспечения для роботизированных систем частично решают проблему универсального программного обеспечения с помощью бесплатной информационной системы ROS, которая так же имеет свои ограничения и не всегда способна удовлетворить все требования производителей.

Материалы и методы. Оборудование

Исследования проводились с использованием двух роботов производства НПО «Андроидная техника»: стенда «Мехатроника» и полноразмерного антропоморфного робота AR-601E (рисунок 1). Первый представляет механический манипулятор в виде руки, имеющий 10 степеней свободы. Второй достаточно близко повторяет анатомию человеческого тела, имея по 13 степеней свободы на каждой руке, по 6 – на ногах, 1 – в тазе, 3 – в шее.

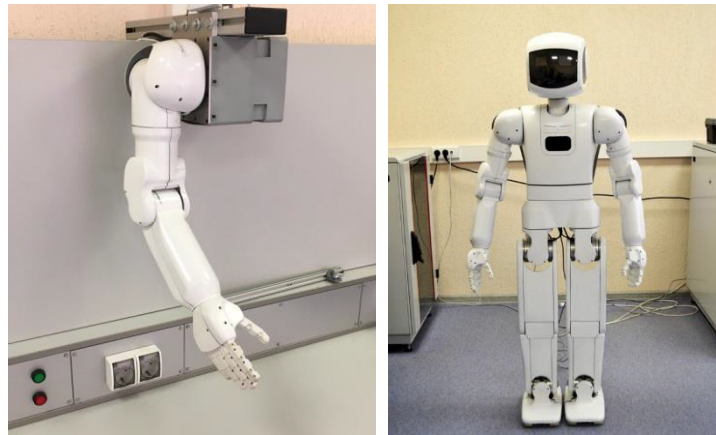


Рис. 1. Стенд «Мехатроника» (слева) и полноразмерный антропоморфный робот AR-601E (справа).

С технической точки зрения роботы сильно похожи: управление происходит путем передачи команд материнской плате робота по каналу связи LAN и приему показаний с сенсоров. Но по причине использования разных моделей контроллеров моторов, используются две различные информационные системы на базе ROS.

Программное обеспечение

В стандартную комплектацию к роботам НПО «Андроидная техника» входит информационная система управления на базе ROS (Robotics Operation System).

Изначально ROS разрабатывалась для проекта STAIR. Данный проект включал в себя создание и программирование двух роботов STAIR-1 и STAIR-2 (рис. 2) [15].

Для решения задач проекта STAIR к ROS предъявлялись следующие требования:

1) модульная архитектура – так как проект предусматривал разделение задач на управление мелкой моторикой, распознавание образов, перемещение в пространстве, монолитная архитектура значительно увеличила бы срок выполнения проекта и снизила эффективность разработки (увеличение количества ошибок, увеличение количества экспериментов);

2) параллельные математические вычисления – для распознавания образов и решения обратной задачи кинематики, вычислительной мощности бортовых компьютеров роботов STAIR-1 и STAIR-2 не хватало, поэтому требовалось переносить наиболее сложные вычисления на другие, более мощные ЭВМ посредством протокола LAN;

3) аппаратная независимость – роботы STAIR-1 и STAIR-2 имели различные аппаратные архитектуры, поэтому требовалось, чтобы ROS создавала единый программный интерфейс для возможности исполнения одних и тех же программ действий на обоих роботах без изменения кода программ.

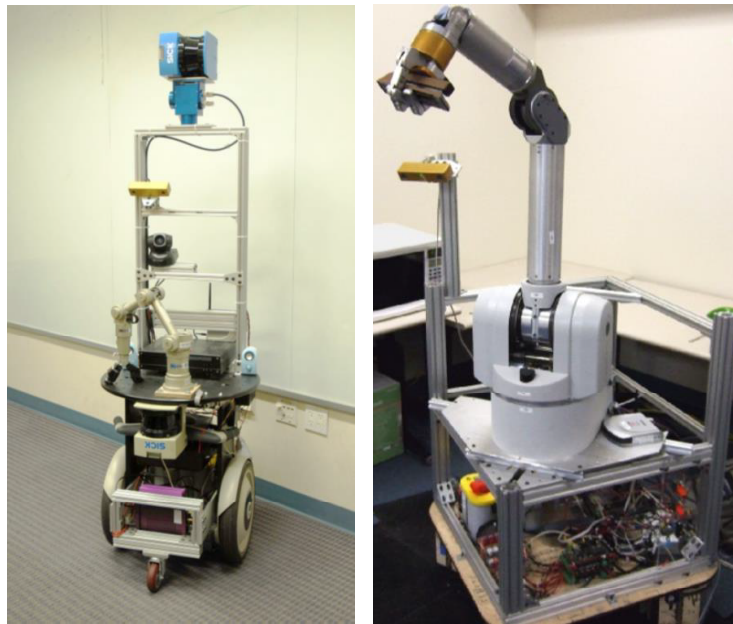


Рис. 2. Роботы STAIR-1 (слева) и STAIR-2 (справа) [15]

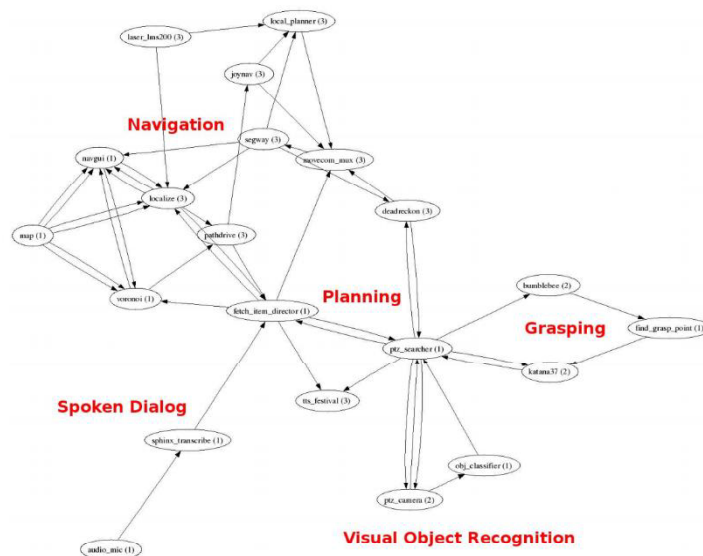


Рис. 3. Блок схема ROS для проекта STAIR [15]

В связи с поставленными требованиями, условиями работы и целями проекта STAIR, Robotics Operation System имеет ряд особенностей:

- поддержка операционных систем только на базе ядра Linux;
- отсутствие шаблонных базовых функций, используемых при программировании практически любой роботизированной системы (например, решения прямой и обратной задач кинематики);
- отсутствие графического интерфейса пользователя;
- организация связи модулей ROS посредством логической локальной сети на каждом компьютере и физической между несколькими компьютерами (рис. 4);

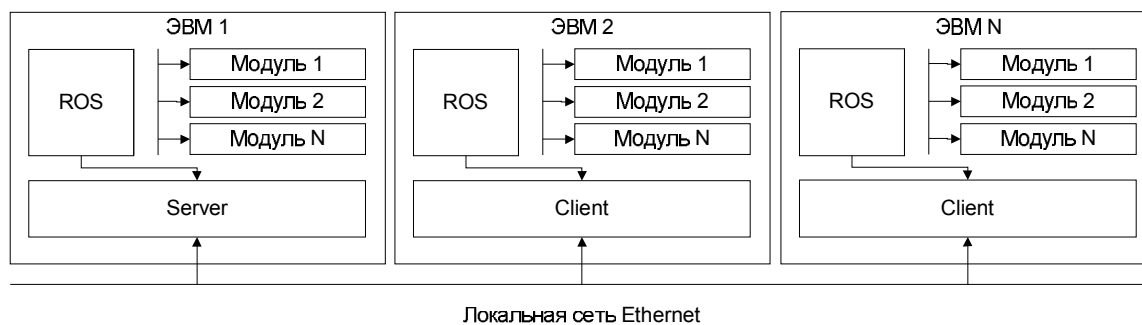


Рис. 4. Структура связей между модулями системы ROS

Данные особенности делают использование ROS в лаборатории робототехнических систем недостаточно эффективным, поскольку отсутствие графического интерфейса усложняет тестирование и отладку новых программ и делает невозможным образовательный процесс; отсутствие в ОС Linux полноценной поддержки оборудования серьезно ограничивает вычислительную мощность исполняющей ЭВМ; крайне малая эффективность использования локальной сети Ethernet в решении задач распараллеливания математических вычислений, где требуется высокая скорость передачи информации между вычислительными устройствами, не позволяет распараллеливать серьезные математические вычисления; отсутствие базового функционала усложняет разработку высокоуровневых модулей, требует дублирования программного кода, что приводит к большим потерям вычислительной мощности;

Альтернативное программное обеспечение

Так как ROS содержит ряд недостатков и особенностей, не позволяющих в полной мере использовать имеющееся оборудование, были проанализированы существующие информационные системы для программирования и управления роботизированными системами.

Player – сервер распределенных устройств [21]. Проект реализуется в Южном Калифорнийском университете. Так же как и ROS является Middleware-ПО, то есть посредником между программой-клиентом, установленным на робототехническую систему, средой разработки на компьютере разработчика [1–3, 14, 19, 21]. Соответственно, Player так же не решает основных проблем, связанных с использованием системы ROS.

Marie – (англ. Mobile and Autonomous Robotics Integration Environment) среда интеграции мобильных и автономных робототехнических систем была разработана Лабораторией мобильных робототехники и интеллектуальных систем Университета Шербрука, Канада [4–7]. Состоит из 3 программных «слоев» [6]: «Приложение», «Компоненты» и «Ядро». Слои «Приложение» и «Компоненты» предоставляют минимум инструментов для разработки программ управления, а «Ядро» содержит инструменты, отвечающие за обмен данными между слоями и подключенным оборудованием, управления производится центральным программным модулем «Медиатором» (рис. 5). Marie не удовлетворяет требованиям к информационной системе из-за отсутствия встроенных инструментов моделирования и эмуляции робототехнических систем.

Orocos – инструментальный реального времени (RTT), для разработки программ управления робототехническими системами на языке C++. Представляет собой библиотеку инструментов для работы с робототехническими системами [8, 9, 11, 12, 16–18]. Среди недостатков можно выделить: отсутствие графического пользовательского интерфейса, системы эмуляции и отладки.

Microsoft Robotics Developer Studio (MRDS). Платформа MRDS включает в себя язык визуального программирования Visual Programming Language (VPL) и имитационную визуальную 3D-среду. Язык визуального программирования Visual Programming Language (VPL) предлагается в качестве средства описания алгоритмов поведения роботов для начинающих программистов, язык C# – для профессиональных [10, 13, 20]. Основным недостатком MRDS при использовании является отсутствие контроля за электропитанием оборудования и отказоустойчивости, т.е. при возникновении каких-либо ошибок программы, информационная система неспособна предотвратить возможное повреждение дорогостоящего оборудования.

Также известны коммерческие программные продукты, предназначенные для разработки программного обеспечения робототехнических систем (GostaiUrbi, Evolution Robotics ER1 и др.), а также закрытые разработки военных и силовых ведомств, анализ которых не являлся целью проведенного исследования.

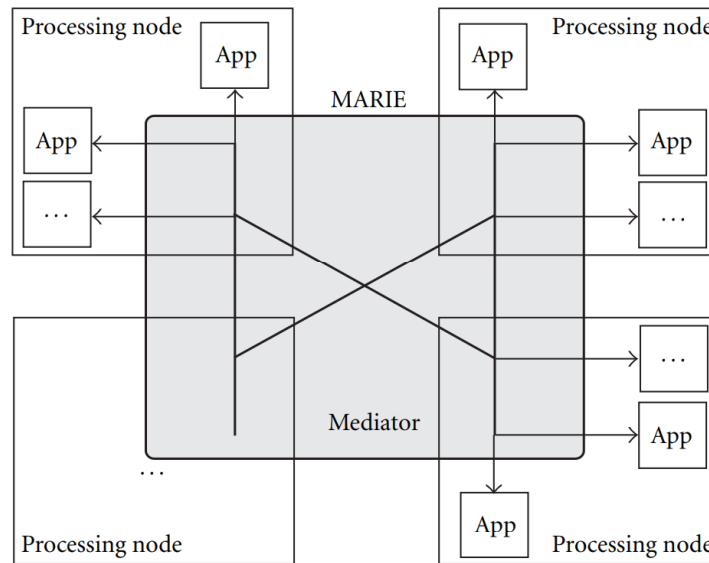


Рис. 5. Пример построения связей между Медиатором и компонентами системы Marie [5]

Анализ информационных систем показал, что все существующие решения не в полной мере удовлетворяют комплексу функциональных требований, необходимых для работы с робототехническим оборудованием: модульности, аппаратной независимости, избыточности, наличия графического интерфейса и эмуляции робототехнических систем. Альтернативным решением является разработанная информационная система управления роботизированными системами RoboStudio, Экспериментальные исследования функционала системы проводились с использованием полноразмерного антропоморфного робота AR-601Еи стенда «Мехатроника».

Достоинства разработанной информационной системы управления роботизированными системами RoboStudio

Модульность – как и ROS, разработанная информационная система имеет возможность быстрого расширения функционала и разделения задач между модулями системы. Связь между модулями поддерживается прямым вызовом методов средствами Win32 и платформы .NET, а не через локальную сеть.

Аппаратная независимость – реализована возможность управления любыми типами роботизированных систем.

Графический интерфейс – для повышения удобства работы и образовательного процесса управление информационной системой производится с помощью графического интерфейса пользователя.

Избыточность – информационная система имеет весь набор базовых инструментов для работы с роботизированными системами: среда разработки, отладки и тестирования, эмуляция роботизированной системы, решение основных задач кинематики и динамики.

Архитектура

Информационная система RoboStudio состоит из следующих модулей (рисунок 6):

- ядро – модуль управления всеми подключаемыми программными частями информационной системы;
- драйвер – динамически заменяемый элемент, соединяющий информационную систему с подключенным роботизированным оборудованием. Компилирует команды исходя из особенностей архитектуры оборудования. Приводит показания датчиков к единому формату. Кроме функций обмена данными с оборудованием, содержит подробную информацию о роботизированной системе для построения ее виртуальной в эмуляторе;
- интерпретатор сценариев – исполняет программы действий, пересылая команды в единый программный интерфейс для дальнейшего их выполнения надстройками, эмулятором или подключенным оборудованием;
- модуль математического моделирования – программный модуль решающий прямую и обратную задачи кинематики;

- эмулятор – программный модуль для апробации и тестирования сценариев на виртуальной модели робота;
- надстройки – подключаемые программные модули, расширяющие функционал информационной системы.

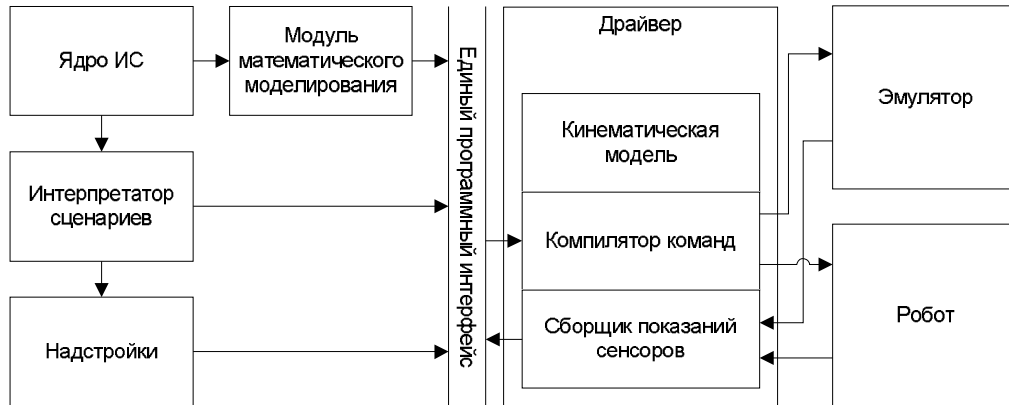


Рис. 6. Архитектура информационной системы управления

Результаты. Практические решения

Каждый модуль системы Robo Studio, кроме ядра, является динамически подключаемой библиотекой Windows (DLL), содержащий определенный набор команд, отвечающих за работу этого модуля. Вызов команд происходит из интерпретатора. При необходимости любой модуль, кроме ядра, может быть заменен на пользовательскую версию соответствующего программного элемента. Благодаря прямой связи между модулями, без использования локальной сети, происходит экономия вычислительной мощности в пользу решения наиболее важных задач.

При запуске системы, интерпретатор выполняет сборку установленной конфигурации модулей и их списки команд в коммутатор (рис. 7), из которого происходит непосредственный вызов той или иной команды. Пользовательские команды перед выполнением проходят стадию трансляции с языка rScript на цифровые значения, соответствующие номерам команд в коммутаторе в то время, как ядро системы отдает команды непосредственно в коммутатор.

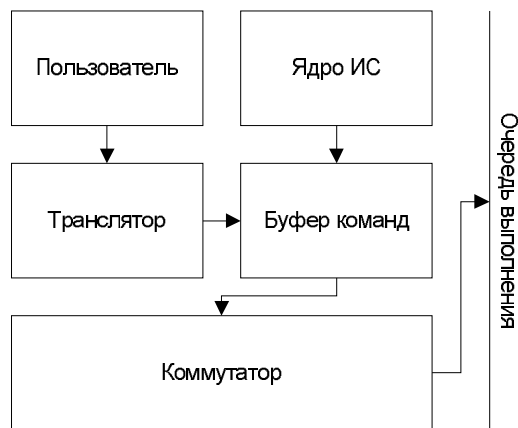


Рис. 7. Схема выполнения команд пользователя и ядра ИС

На рис. 8 проиллюстрирован алгоритм всего цикла выполнения высокоуровневой команды GoFromTo(A, B) – перемещение из точки А в точку В:

Команда поступает в интерпретатор, где переводится в соответствующий ей номер в коммутаторе.

Коммутатор вызывает метод, соответствующий номеру в модуле симуляции ходьбы, где происходит деление всего цикла выполнения задачи на итерации с учетом полученных параметров.

На каждой итерации выполняются решения прямой и обратной задачи кинематики, для нахождения положений сочленений в текущий момент цикла ходьбы.

Из полученных данных формируется команда для роботизированной системы, которая отправляется в ядро ИС на проверку безопасности.

Если команда безопасна (не приведет к излому суставов, зависанию системы и пр.), она отправляется в драйвер, где происходит ее трансляция в машинный код и непосредственное выполнение.

Трансляция в машинный код роботизированной системы

Управление роботами НПО «Андроидная техника» происходит с помощью двух массивов размером по 1472 байт каждый. Первый служит для отправки команд (массив команд) на главный контроллер робота, второй содержит показания сенсоров (массив показаний). Обмен данными происходит в отдельном потоке непрерывно и независимо от того, заданы новые команды или нет. В массиве команд для каждого мотора выделено по 16 байт данных, в которых хранится угол поворота мотора, жесткость привода, и другие параметры сервоприводов (табл. 1).

Эти параметры уникальны и могут различаться в зависимости от используемых моторов, контроллеров и других компонентов электронной цепи робота. Поэтому в разработанной ИС предусмотрен специальный программный модуль – драйвер, который транслирует команды информационной системы в требуемый формат, учитывая аппаратные особенности подключенного оборудования.

Работа информационной системы

Информационная система RoboStudio является средой разработки программ для любых видов роботизированных систем. Благодаря использованию интерпретатора языка сценариев появляется возможность разрабатывать программы управления практически любой сложности и направленности без перекомпиляции основного кода информационной системы.

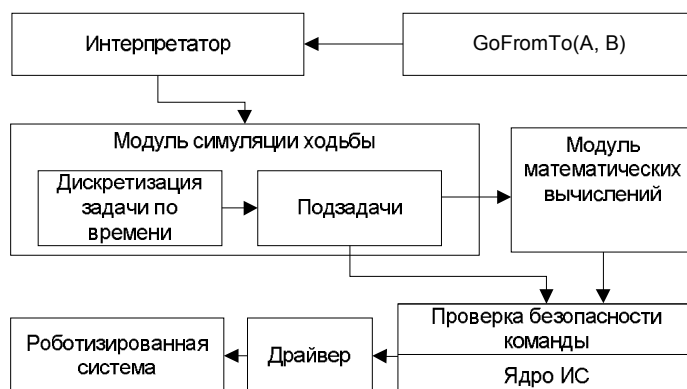


Рис. 8. Схема выполнения функции перемещения робота из точки А в точку В

Эмулятор позволяет производить отладку вновь разработанных сценариев на виртуальной модели робота, тем самым позволяя избегать риски при выполнении ошибочных команд

Модульность системы и подключение драйверов оборудования позволяют расширять функционал информационной системы во время ее работы и организовывать управление практически любыми роботизированными системами.

Таблица 1

Формат команды сервопривода робота AR-601E

Смещение	Размер в байтах	Назначение
0x00	1	Адрес
0x01	1	Состояние мотора (реверс, остановка, трассировка, релаксация)
0x02	2	Текущий угол поворота
0x04	2	ILIM
0x06	2	Позиция по умолчанию
0x08	2	Жесткость муфты
0x0A	2	Демпфер
0x0C	2	Минимальный угол поворота
0x0E	2	Максимальный угол поворота

Пользовательский интерфейс информационной системы (рис. 9) состоит из панели эмулятора, редактора исходного кода, командой консоли и панели состояния, на которой отображается информация об электропитании и показаниях всех датчиков роботизированной системы.

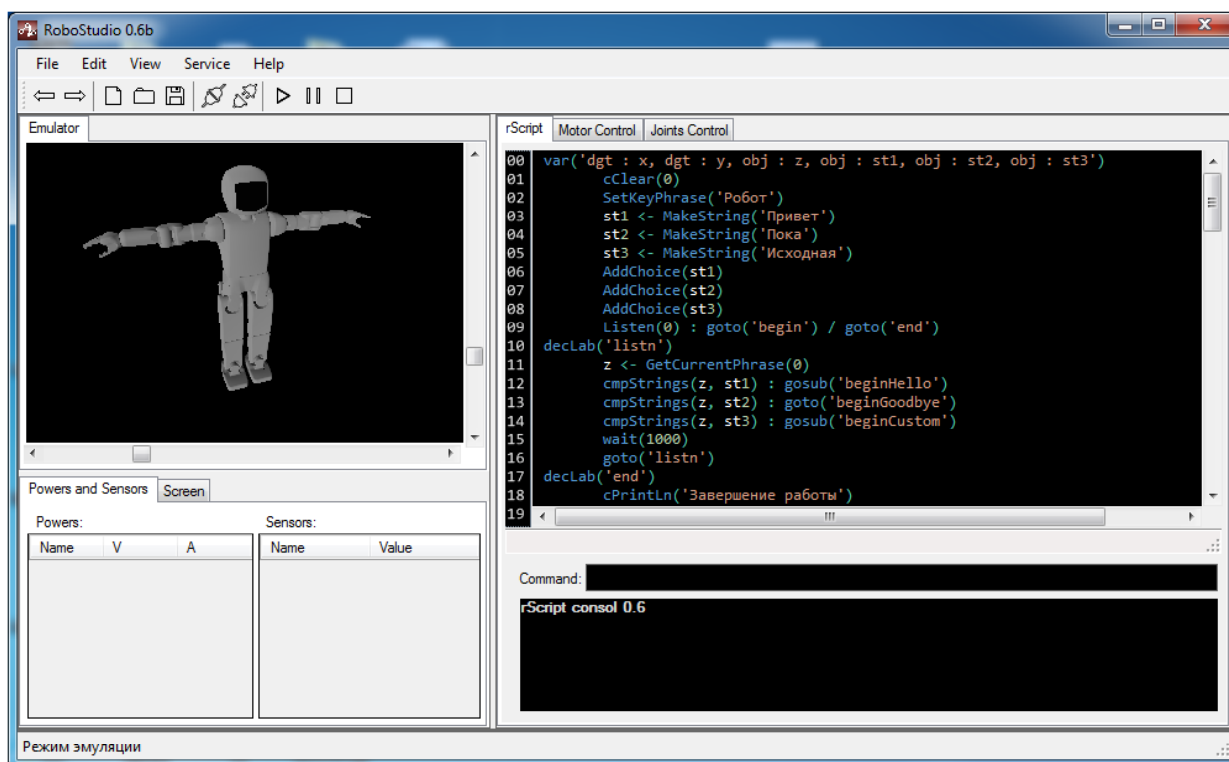


Рис. 9. Вид главного окна информационной системы

Стандартный функционал позволяет избавиться от лишнего программного обеспечения, экономя вычислительную мощность ЭВМ для решения более важных задач, позволяет гибко настроить информационную систему под практически любое оборудование без изменений в исходном коде, осуществлять полный и единовременный контроль за всеми электронными компонентами роботизированной системы, что повышает безопасность использования дорогостоящего оборудования.

Заключение. Именно наличием подобных программных продуктов как RoboStudio, объясняются большие успехи зарубежных разработчиков. Но, к сожалению, такие системы являются засекреченными и, естественно, не распространяются. Таким образом, разработанная система позволит развивать программу импортозамещения в сфере производства робототехники.

Дальнейшее развитие информационной системы RoboStudio

В ROS увеличение производительности осуществляется путем распараллеливания математических задач на несколько ЭВМ. Если на 2007 год (первая версия ROS) это было самым оптимальным решением, то сегодня ЭВМ дают намного больше возможностей параллелизма. Благодаря этому, в будущем планируется перенос части математических вычислений в системе RoboStudio на графический ускоритель.

ЛИТЕРАТУРА / REFERENCES

1. B. Gerkey, R. Vaughan, and A. Howard, "Howard, the player/stage project: tools for multi-robot and distributed sensor systems," in Proceedings of the 11th International Conference on Advanced Robotics (ICAR '03), Coimbra, Portugal, 2003.
2. B. P. Gerkey and M. J. Mataric, "Sold!: auction methods for multi-robot coordination," in Proceedings of the IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems, 2001.
3. B. P. Gerkey, R. T. Vaughan, K. Stoy, A. Howard, G. S. Sukhatme, and M. J. Mataric, "Most valuable player: a robot device server for distributed control," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1226–1231, Wailea, Hawaii, USA, November 2001.

4. C. Coté, D. Létourneau, C. Raïevsky, Y. Brosseau, and F. Michaud, "Using marie for mobile robot component development and integration," Software Engineering for Experimental Robotics Book Series, vol. 30 of Springer Tracts in Advanced Robotics, Springer, Berlin, Germany, 2007.
5. C. Coté, D. Létourneau, F. Michaud et al., "Code reusability tools for programming mobile robots," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04), pp. 1820–1825, October 2004.
6. C. Coté, Y. Brosseau, D. Létourneau, C. Raïevsky, and F. Michaud, "Robotic software integration using MARIE," International Journal of Advanced Robotic Systems, vol. 3, no. 1, pp. 55–60, 2006.
7. É. Beaudry, Y. Brosseau, C. Ct et al., "Reactive planning in a motivated behavioral architecture," in Proceedings of the National Conference on Artificial Intelligence (AAAI '05), vol. 3, pp. 1242–1247, 2005.
8. H. Bruyninckx, J. De Schutter, T. Lefebvre et al., "Building blocks for slam in autonomous compliant motion," in Proceedings of the International Symposium on Robotics Research (ISRR '03), pp. 432–441, 2003.
9. H. Bruyninckx, P. Soetens, and B. Koninckx, "The real-time motion control core of the Orocos project," in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2766–2771, September 2003.
10. J. Jackson, "Microsoft robotics studio: a technical introduction," IEEE Robotics and Automation Magazine, vol. 14, no. 4, pp. 82–87, 2007.
11. K. Gadeyne, Sequential monte carlo methods for rigorous bayesian modeling of autonomous compliant motion, Ph.D. thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, 2005.
12. K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian hybrid model-state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation," The International Journal of Robotics Research, vol. 24, no. 8, pp. 615–630, 2005.
13. K. Johns and T. Taylor, Professional Microsoft Robotics Developer Studio, Wrox Press, Birmingham, UK, 2008.
14. M. Kranz, R. B. Rusu, A. Maldonado, M. Beetz, and A. Schmidt, "A player/stage system for context-aware intelligent environments," in Proceedings of the System Support for Ubiquitous Computing Workshop, at the 8th Annual Conference on Ubiquitous Computing (Ubicomp '06), Orange, Calif, USA, September 2006.
15. Morgan Quigley, Eric Berger, Andrew Y. Ng (2007), STAIR: Hardware and Software Architecture, AAAI 2007 Robotics Workshop
16. P. Soetens and H. Bruyninckx, "Realtime hybrid task-based control for robots and machine tools," in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 260–265, April 2005.
17. P. Soetens, A software framework for real-time and distributed robot and machine control, Ph.D. thesis, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Heverlee, Belgium, 2006, <http://www.mech.kuleuven.be/dept/resources/docs/soetens.pdf>.
18. P. Soetens, RTT: Real-Time Toolkit, 2010, <http://www.Orocos.org/rtt>.
19. R. T. Vaughan, B. P. Gerkey, and A. Howard, "Howard, on device abstractions for portable, reusable robot code," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03), pp. 2121–2427, Las Vegas, Nev, USA, 2003.
20. S. Morgan, Programming Microsoft Robotics Studio, Microsoft Press, Redmond, Wash, USA, 2008.
21. T. H. Collett, B. A. MacDonald, and B. P. Gerkey, "Player 2.0: toward a practical robot programming framework," in Proceedings of the Australasian Conference on Robotics and Automation (ACRA '05), Sydney, Australia, 2005.

ОБ АВТОРАХ

Шевченко Александр Владимирович, аспирант кафедры информационных систем и технологий, Институт информационных технологий и телекоммуникаций ФГАОУ ВО «Северо-Кавказский федеральный университет», г. Ставрополь, + 7-961-465-48-51, E-mail: luckyleo769@mail.ru

Shevchenko Aleksey Vladimirovich, Graduate student, Chair of Information Systems and Technologies, Institute of Information Technologies and Telecommunications FSAEI HE "North-Caucasus Federal University", Stavropol+ 7-961-465-48-51, E-mail: luckyleo769@mail.ru

Мезентцева Оксана Станиславовна, кандидат физико-математических наук, доцент, профессор кафедры информационных систем и технологий, Институт информационных технологий и телекоммуникаций ФГАОУ ВО «Северо-Кавказский федеральный университет», г. Ставрополь, E-mail: omezentceva@ncfu.ru, (8652) 94-54-75

Mezentseva Oksana Stanislavovna, Candidate of physical and mathematical sciences, docent, Professor of the Department of Information Systems and Technologies, Institute of Information Technologies and Telecommunications FSAEI HE "North-Caucasus Federal University", Stavropol

Мезенцев Дмитрий Викторович, аспирант кафедры информационных систем и технологий, Институт информационных технологий и телекоммуникаций ФГАОУ ВО «Северо-Кавказский федеральный университет», г. Ставрополь, E-mail: dmezentcev@ncfu.ru), + 7-961-466-14-42

Mezentsev Dmitriy Viktorovich, Graduate student, Chair of Information Systems and Technologies, Institute of Information Technologies and Telecommunications FSAEI HE "North-Caucasus Federal University", Stavropol, E-mail: dmezentcev@ncfu.ru), + 7-961-466-14-42

Дата поступления в редакцию 16.07.2018