

Разработка средств визуализации и отладки управляющих программ для электроавтоматики, интегрированных в систему ЧПУ

Visualization and debugging system integrated in CNC for electric automation programs

Статья посвящена исследованию и анализу средств для программирования логических контроллеров, а также разработке универсального инструментария для создания, визуализации и отладки управляющих программ для электроавтоматики, интегрированного в систему ЧПУ AxiOMA Ctrl.

The article is devoted to research and analysis tools for programming logic controllers, and the development of generic tools for the creation, visualization and debugging of programs for electric automation, integrated into the control system AxiOMA Ctrl.

Ключевые слова: программно-реализованный контроллер, Soft PLC, система ЧПУ, среда программирования, ПЛК, электроавтоматика станка.

Key words: virtual controller, Soft PLC, numerical control system, metalworking, PLC, machine tool electric components.

Программируемые логические контроллеры (ПЛК) на сегодняшний день являются базовыми элементами систем промышленной автоматизации. На их основе построены все АСУ ТП, системы мониторинга, контроля функционирования, телеметрии, обеспечения безопасности и многие другие. Также, огромное значение современные ПЛК занимают в системах ЧПУ.

Мировые производители систем ЧПУ, такие как Siemens, Bosch Rexroth, Heidenhain, в большинстве своем узко ориентированы на конкретную аппаратуру, разработанную в их компаниях. Появляется задача в выборе средства для программирования электроавтоматики от различных производителей.

На сегодняшний день, существуют универсальные средства для программирования логических контроллеров, такие как CoDeSys, ISaGRAF и др. Пакет CoDeSys является одним из наиболее известных универсальных инструментов стандарта МЭК 61131 программирования для ПЛК и промышленных компьютеров. Его используют многие компании-изготовители аппаратных средств индустрии автоматизации. CoDeSys включает богатый набор средств отладки и сопровождения. Кроме среды программирования, в состав комплекса CoDeSys входят: SP RTE (эмуляция ПЛК на ПК), Soft Motion (набор средств управления движением – до многомерной интерполяции современных систем ЧПУ), ARTI (обеспечивает символьный доступ к переменным в ПЛК) и ENI Server (позволяет работать с одним проектом не-

скольким пользователям). CoDeSys обладает достаточно широкими графическими возможностями, которые по функциональности не уступают простейшим SCADA-системам.

Для решения поставленных задач данный пакет является слишком громоздким и имеет следующие недостатки: высокая цена, сложность в обслуживании, высокая специализация сотрудников.

Таким образом, просматривается необходимость в создании универсального средства по созданию, визуализации и отладки управляющих программ для электроавтоматики. Данное средство должно являться простым и универсальным инструментарием для разработки управляющих программ для программируемых логических контроллеров, адаптированным под работу с аппаратной частью от различных производителей. Можно выделить следующие основные требования для разрабатываемого инструментария: аппаратная независимость, отсутствие платной лицензии, универсальность и простота работы, стандартизированный язык программирования, наличие режима эмуляции, возможность использования в учебных целях.

При разработке инструментария за прототип был взят вышеупомянутый CoDeSys, как один из наиболее известных универсальных инструментов программирования ПЛК и промышленных компьютеров. Графический редактор управляющих программ для электроавтоматики имеет широкие возможности для создания и редактирования программ для ПЛК на

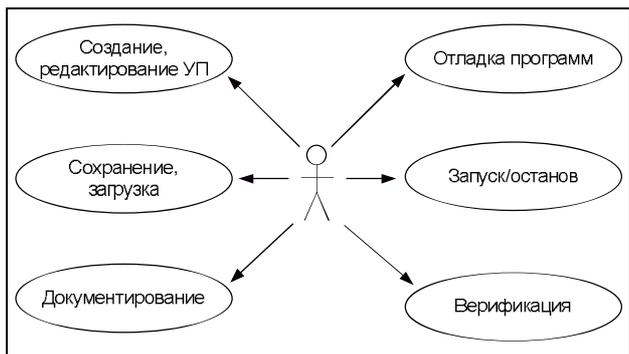


Рис. 1. Диаграмма прецедентов инструментария для разработки и отладки управляющих программ электроавтоматики

языке Functional Block Diagram (FBD). Диаграмма прецедентов среды разработки управляющих программ представлена на Рисунок 1.

Разработка описываемого инструментария для программирования и отладки управляющих программ ПЛК велась с использованием технологий .NET,

XML. При разработке инструментария сохранялось следование основам объектно-ориентированного проектирования, таким, как компонентная архитектура, модульность, а также использование концепции MVC (Model View Controller) – разбиения программного продукта на отдельные модули, которые отвечают за хранение, визуализацию и управление данными. Совокупность использования данных решений позволяет добиться высокой гибкости разрабатываемого ПО, а также независимости модуля хранения данных и модуля визуализации.

Интерфейс среды разработки управляющих программ представленный на рисунке 2 позволяет производить редактирование как визуальной части (создание/удаление/перемещение функциональных блоков, масштабирование, регулировка приоритета видимости, копирование объектов, и др.), так и настраивать все параметры функциональных блоков при помощи панели настроек.

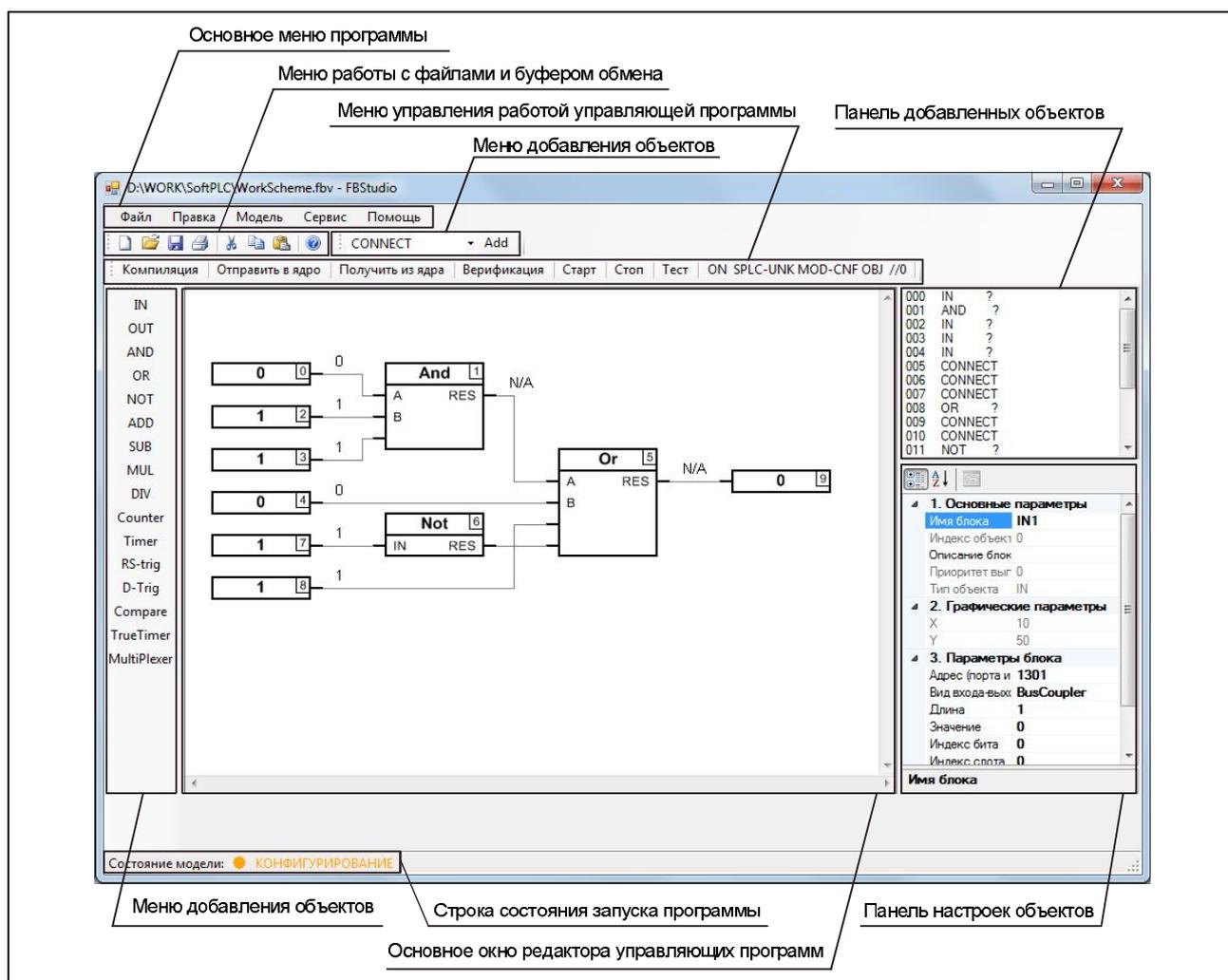


Рис. 2. Интерфейс основного экрана редактора программ для электроавтоматики

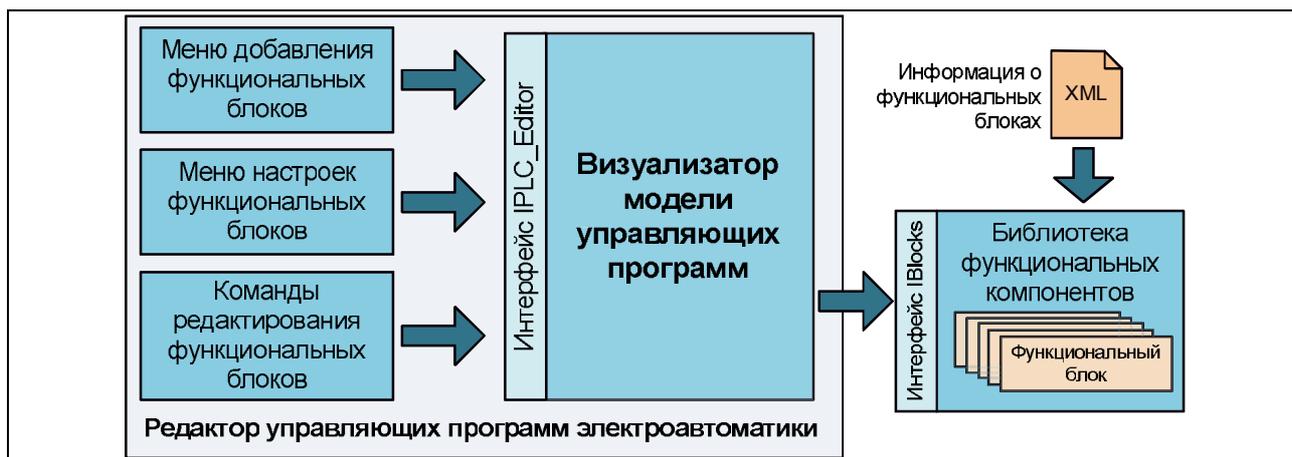


Рис. 3. Основные модули редактора управляющих программ электроавтоматики

Программирование управляющей программы для ПЛК сводится к соединению готовых компонентов линиями связи. Полученные схемы четко отражают взаимосвязь входов и выходов всех функциональных блоков, используя технологию инкапсуляции алгоритмов обработки данных. Также, имеется возможность сохранять и загружать разработанные модели в специализированный файл с расширением «fbv», а также производить экспорт в графические форматы (bmp, jpg) и печать программы. Перечисленный функционал в полной мере дает широкие возможности для подробной визуализации и документирования разработанных управляющих программ.

С точки зрения архитектуры программного обеспечения, весь функционал разработанного инструментария делится на отдельные модули. Схема взаимодействия основных модулей продукта представлена на Рисунок 3.

Модули выполнены независимо друг от друга и взаимодействуют между собой через специально разработанные программные интерфейсы, что позволяет следовать основным парадигмам компонентного подхода и модульной архитектуры. Основным модулем является визуализатор модели управляющих программ, на вход которого подаются команды на добавление, редактирование функциональных блоков и настройку модели. Разработанный инструментарий содержит в себе необходимый набор элементов на языке FBD, с помощью которого можно решать задачи автоматизации производственных процессов.

Описанный набор функциональных блоков хранятся в библиотеке функциональных компонентов, которая содержит в себе информацию обо всех объектах, которыми можно оперировать при создании управляющей программы для ПЛК. Программный интерфейс данной библиотеки содержит методы, по-

зволяющие получать объекты функциональных блоков по их названию или индексу. Каждый объект функционального блока состоит из графической и модельной частей. Графическая часть содержит поля и методы для визуализации функционального блока, а модельная часть хранит в себе информацию, необходимую для реализации логики работы блока. Описание функциональных блоков содержится в XML-файле, что дает возможность удобного редактирования имеющихся компонентов и добавления новых. Данная схема содержит информацию о типе блока, его имени, а также обо всех параметрах его входов и выходов. XSD схема описания функционального блока представлена на Рисунок 4.

Рассмотрим подробнее процесс создания и отладки управляющих программ. На начальном этапе разработки управляющей программы для ПЛК, выполняется добавление функциональных блоков из панели меню в окно модели, используя технологии «Drag & drop». Затем, функциональные блоки связываются между собой и устанавливаются приоритеты выполнения каждого, что определяет последовательность их работы. Также, разработанный инструментарий позволяет редактировать изломы линий связи, позволяя избегать наложений соединений на функциональные блоки, что делает визуализацию модели более наглядной и удобной.

Следующим этапом работы является настройка параметров работы всех блоков. Это можно осуществить двумя способами: установить требуемые параметры в панели настроек блока, или открыть специализированное окно настроек блока. Основные параметры работы с аппаратным обеспечением настраиваются в блоках входа/выхода. Функционал программы позволяет настраивать следующие параметры входов/выходов: адрес порта или памяти, вид входа/выхода (BusCoupler, CommonPlcMemory, Port),

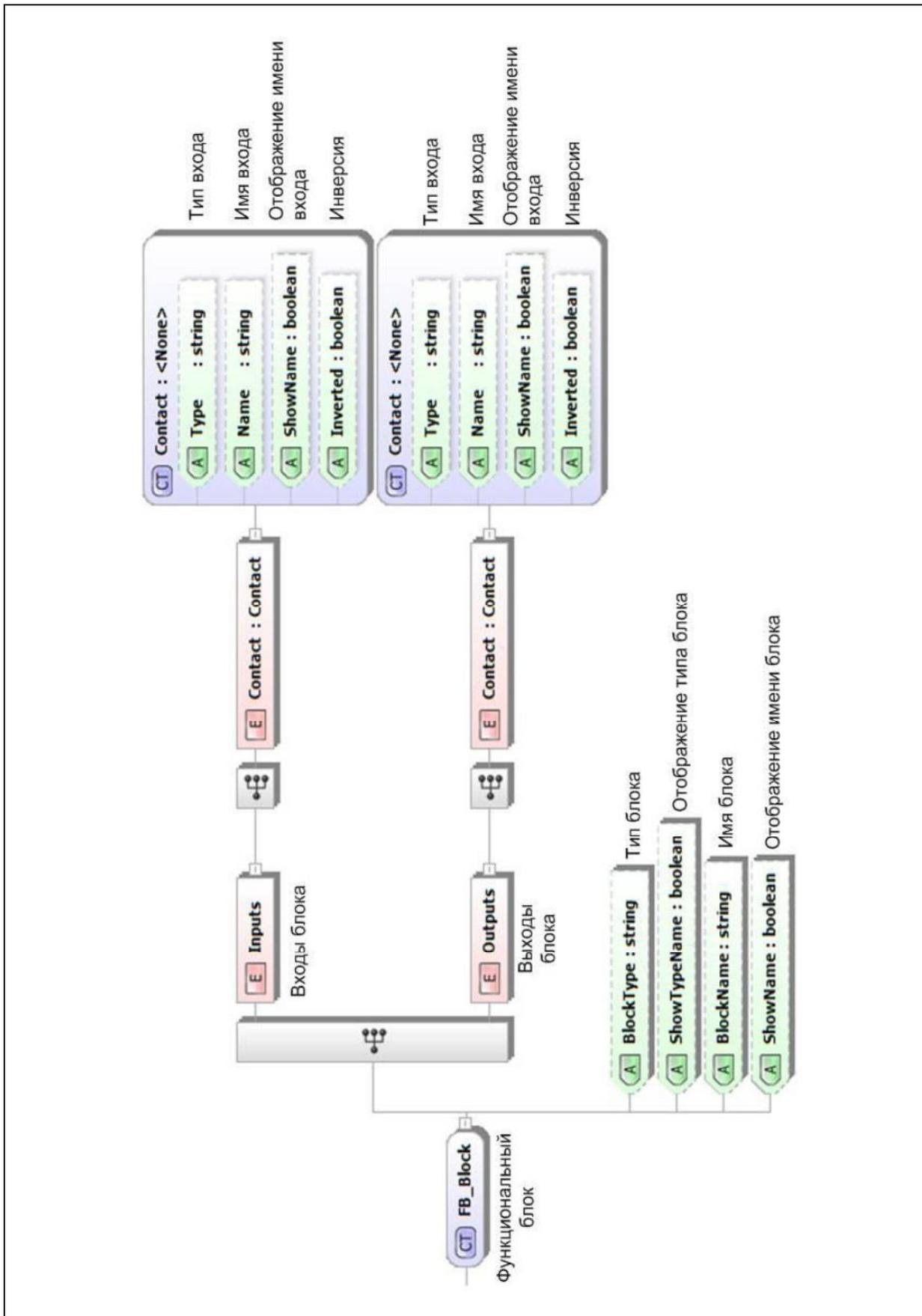


Рис. 4. XSD-схема описания блока в библиотеке функциональных компонентов



Рис. 5. Схема взаимодействия редактора управляющих программ с ядром системы ЧПУ

длина, значение, индекс бита, индекс слота, индекс устройства, тип хранимого значения (Bit, Bool, Analog, Digital).

После настройки всех параметров функциональных блоков и завершения разработки управляющей программы, обязательным требованием является проведение верификации модели на уровне терминала, т.е. проверки на пустые контакты, несвязанные блоки, и т.д. Для этого требуется нажать на кнопку «Верификация» в меню управления работой программы. Без успешного завершения данного действия, невозможно выполнить запуск и отладку модели.

Для тестирования разработанных функциональных схем и работы с аппаратной частью электроавтоматики была произведена интеграция разработанного инструментария с системой ЧПУ AxiOMA Ctrl и разработкой канала передачи данных между терминалом и ядром системы ЧПУ. Таким образом, при полноценной работе с разработанным инструментарием требуются две основные части программы: терминал – вышеописанная графическая оболочка, позволяющая моделировать управляющие программы для электроавтоматики независимо от любой аппаратуры, а также ядро, в котором выполняются необходимые расчеты, реализуется логика работы построенной модели, а также производится привязка к аппаратуре. Связующим звеном между данными модулями является клиентская часть, позволяющая осуществлять связь между терминалом инструментария и ядром системы ЧПУ через специализированные интерфейсы. На Рисунок 5 показана схема взаимодействия редактора управляющих программ с ядром системы ЧПУ.

При запуске разработанной управляющей программы для электроавтоматики происходит передача специфицированного пакета данных из терминала

редактора в ядро системы ЧПУ. Данный пакет содержит в себе всю необходимую информацию о команде, которую необходимо выполнить, версии редактора, а также всех его функциональных блоков. Структура передаваемого пакета данных при отправке в ядро объекта управляющей программы представлена на Рисунок 6.

При отправке команды на запуск разработанная управляющая программа начинает работать в ядре системы ЧПУ. Ядро, в свою очередь, производит отправку текущего состояния работы программы, которое будет отображаться в строке состояния редактора. Также, при включенном режиме отладки программы, ядро системы ЧПУ с заданной частотой производит отправку в терминальную часть редактора информацию о блоках, значения в которых изменились. Таким образом, имеется возможность в режиме реального времени производить отладку управляющей программы и следить за актуальными значениями входов/выходов разработанной модели.

В ходе работы был проведен анализ имеющихся средств программирования и отладки рабочих программ для логических контроллеров, который показал, что для задач интеграции программного обеспечения в состав систем ЧПУ необходим специализированный аппаратно независимый инструментарий, который был разработан и интегрирован в состав системы ЧПУ AxiOMA Ctrl. С помощью созданного средства имеется возможность решать множество задач по программированию логических контроллеров на языке FBD стандарта МЭК 61131, независимо от используемой аппаратуры. Также, имеются возможность использования данного средства в учебных целях на специально оборудованных стендах программирования электроавтоматики.



Рис. 6. Обобщенная структура пакета данных при передаче в ядро функционального блока управляющей программы

Библиографический список

1. **Мартинов Г. М., Мартинова Л. И.** - Современные тенденции в области числового программного управления станочными комплексами // СТИН. 2010. №7
2. **Григорьев С.Н., Андреев А.Г., Мартинов Г.М.** Перспективы развития кроссплатформенных компьютерных систем числового программного управления высокотехнологичного оборудования // Автоматизация в промышленности, 2011. №5, с. 3-8.
3. **Мартинов Г.М., Козак Н.В., Нежметдинов Р.А., Пушков Р.Л.** Принцип построения распределенной системы ЧПУ с открытой модульной архитектурой // Вестник МГТУ "Станкин". 2010. №4(12).
4. **Мартинов Г.М., Нежметдинов Р.А., Козак Н.В., Пушков Р.Л.** Прикладные решения в области управления электроавтоматикой станков с ЧПУ класса PCNC // Промышленные АСУ и контроллеры. 2011. № 4.
5. **Нежметдинов Р.А., Шемелин В.К.**, Повышение качества архитектурных решений систем ЧПУ на основе программно реализованного контроллера типа SoftPLC // Автоматизация и современные технологии. 2008. № 6.
6. **Мартинова Л.И., Козак Н.В., Нежметдинов Р.А., Пушков Р.Л.** Реализация открытости управления электроавтоматикой станков в системе ЧПУ класса PCNC // Приборы и системы. Управление, контроль, диагностика. 2011. № 02.
7. **Мартинов Г.М., Мартинова Л.И., Козак Н.В., Нежметдинов Р.А., Пушков Р.Л.** Принципы построения распределенной системы ЧПУ технологическими машинами с использованием открытой модульной архитектуры // Справочник. Инженерный журнал с приложением. 2011. № 12.
8. **Нежметдинов Р.А., Соколов С.В., Обухов А.И., Григорьев А.С.**, Расширение функциональных возможностей систем ЧПУ для управления механо-лазерной обработкой // Автоматизация в промышленности. 2011. № 5.
9. **Мартинова Л.И., Козак Н.В., Нежметдинов Р.А., Пушков Р.Л., Обухов А.И.** Практические аспекты применения отечественной многофункциональной системы ЧПУ "АксиОМА Контроль" // Автоматизация в промышленности. 2012. № 5.
10. **Мартинова Л.И., Мартинов Г.М.** Организация межмодульного взаимодействия в распределенных системах ЧПУ. Модели и алгоритмы реализации // Мехатроника, автоматизация, управление. 2010. №11.
11. **Мартинов Г.М., Мартинова Л.И.** Практические аспекты реализации сплайн-контуров в системах ЧПУ класса PCNC // Информационные технологии в проектировании и производстве. 2012. №3.
12. **Мартинов Г.М., Нежметдинов Р.А., Соколов С.В.** Способ построения инструментария систем мониторинга и настройки параметров мехатронного технологического оборудования на основе специализированных программных средств // Мехатроника, автоматизация, управление. 2012, №7.
13. **Мартинов Г.М., Любимов А.Б., Бондаренко А.И., Сорокоумов А.Е., Ковалев И.А.** Подход к построению мультипротокольной системы ЧПУ // Автоматизация в промышленности. 2012. №5.
14. **Мартинов Г. М., Козак Н. В.** Декомпозиция и синтез программных компонентов электроавтоматики // Приборы и системы. Управление, контроль, диагностика. 2006. №12.

Мартинов Георги Мартинов – доктор технических наук, профессор, зав. кафедрой «Компьютерные Системы Управления» МГТУ «Станкин».

Тел.: (499) 972-94-40.

Нежметдинов Рамиль Амирович – кандидат технических наук, доцент кафедры «Компьютерные Системы Управления» МГТУ «СТАНКИН»,

Tel: (499) 972-94-40, E-mail: neramil@ncsystems.ru

Никишечкин Петр Анатольевич, аспирант кафедры «Компьютерные Системы Управления» МГТУ «СТАНКИН»,

Tel: (499) 972-94-40, E-mail: petrnikishechkin@gmail.com

Nikishechkin Petr Anatolievich, postgraduate student, department of “Computer-architecture Control System” of MSTU “STANKIN”,

Tel: (499) 972-94-40, E-mail: petrnikishechkin@gmail.com

Martinov Georgi Martinov – doctor Sc. of Engineering, head of the sub-department of “Computer-architecture Control System” of MSTU “STANKIN”.

Tel: (499) 972-94-40, E-mail: martinov@ncsystems.ru

Nezhmetdinov Ramil Amirovich - candidate Sc. of Engineering, department of “Computer-architecture Control System” of MSTU “STANKIN”

Tel: (499) 972-94-40, E-mail: neramil@ncsystems.ru

