

Разработка средств визуализации и отладки управляющих программ для электроавтоматики, интегрированных в систему ЧПУ¹

П.А. Никитченко,
инж. 2 кат., pnikishchkin@gmail.com
МГТУ «Станкин», г. Москва

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 of the tools for programming logic controllers, and to the development of generic tools for the creation, visualization and debugging of control programs for electrics, that are integrated into the CNC system AxiOMA Ctrl.

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 многое другое. Также, современные ПЛК имеют огромное значение в системах числового программного управления.

Мировые производители систем ЧПУ, такие как Siemens, Bosch Rexroth, Heidenhain, в большинстве своем узко ориентированы на конкретную аппаратуру, разработанную в их компаниях; поэтому выбор универсального средства для программирования электроавтоматики от различных производителей становится актуальной задачей, требующей решения. 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 программирования для ПЛК и промышленных компьютеров. Его используют более ста известных компаний-изготовителей 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, включая многомерную интерполяцию современных систем ЧПУ), ARTI (обеспечивает символьный доступ к переменным в ПЛК) и ENI Server (позволяет работать с одним проектом несколькими пользователями). Также CoDeSys обладает достаточно широкими графическими возможностями, которые по функциональности не уступают простейшим SCADA-системам.

Однако для решения поставленных задач данный пакет является слишком громоздким и имеет следующие недостатки:

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

При разработке описываемого инструментария за прототип был взят вышеупомянутый CoDeSys, как один из наиболее известных универсальных инструментов программирования для ПЛК и промышленных компьютеров.

Разработка описываемого средства для программирования и отладки управляющих программ для 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 на отдельные модули частей программы, 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.

Разработанный графический редактор управляющих программ для электроавтоматики имеет широкие возможности для создания и редактирования программ для ПЛК на языке FBD. Диаграмма прецедентов разрабатываемого инструментария показана на рис. 1.

Разработанный инструментарий имеет удобный интерфейс, позволяющий 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. Программирование управляющей программы для ПЛК сводится к «склеиванию» готовых компонентов – функциональных блоков.

¹ Работа выполнена по Госконтракту № 16.120.11.323-МК от 18.02.2011г. на проведение НИР в рамках ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 г.

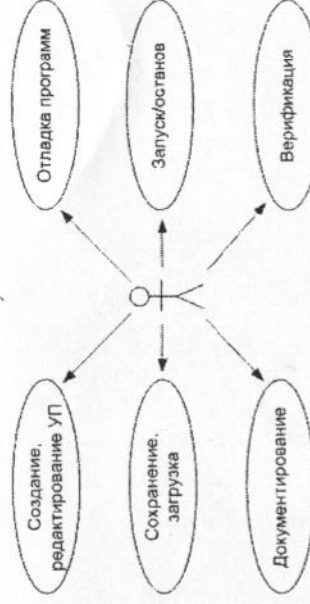


рис. 1 Диаграмма прецедентов инструментария для разработки и отладки управляющих программ электроавтоматики

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 экспортрование в графические форматы (bmp, jpg) и производить печать программы. Перечисленный функционал в полной мере дает широкие возможности для подробной визуализации и документирования разработанных управляющих программ. Интерфейс основного окна программы показан на рис. 2.

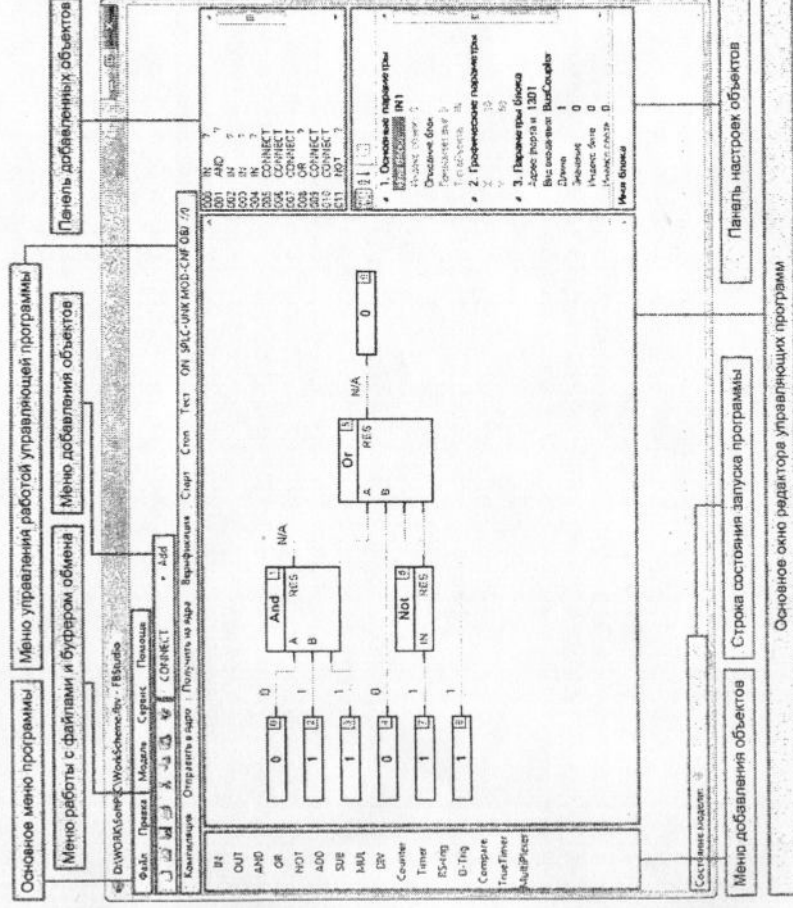


рис. 2 Интерфейс основного экрана редактора программ для электроавтоматики

С точки зрения архитектуры программы, весь функционал разработанного инструментария делится на отдельные модули. Диаграмма основных модулей продукта представлена на рис. 3.

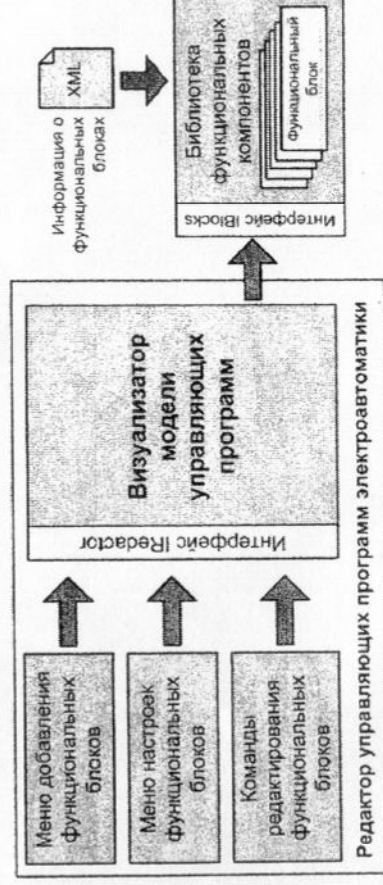


рис. 3 Основные модули редактора управляющих программ электроавтоматики

Как видно из рисунка, все модули выполнены независимо друг от друга и взаимодействуют между собой через специально утвержденные интерфейсы, что позволяет следовать основным парадигмам компонентного подхода и модульной архитектуры. Основным модулем является визуализатор модели управляющих программ, на вход которого подаются основные команды на добавление, редактирование функциональных блоков и настройку модели.

На данный момент разработанный инструментарий позволяет оперировать в управляющей программе следующими функциональными блоками:

1. Входы/выходы:
 - блок входа (IN);
 - блок выхода (OUT);
2. Логические операции:
 - дизъюнкция (AND);
 - конъюнкция (OR);
 - отрицание (NOT);
3. Арифметические операции:
 - сложение (ADD);
 - вычитание (SUB);
 - умножение (MUL);
 - деление (DIV);
4. Триггеры:
 - D-триггер;
 - RS-триггер;
5. Счетные элементы:
 - таймер;
 - счетчик;

Перечисленный набор компонентов содержит все логические элементы, требуемые для создания программ для ПЛК среднего уровня. Функциональные блоки хранятся в библиотеке функциональных компонентов, которая содержит в себе всю информацию обо всех объектах, которыми можно оперировать при создании управляющей программы для ПЛК. Интерфейс данной библиотеки содержит методы, позволяющие получать объекты функциональных блоков по их названию или индексу. Также каждый объект функционального блока состоит из графической и модельной частей. Графическая часть содержит все необходимые поля и методы для визуализации функционального блока, а модельная часть хранит в себе информацию, необходимую для реализации логики работы блока. Описание всех функциональных блоков содержится в XML-файле, что позволяет удобно редактировать имеющиеся компоненты и добавлять новые. Данная схема содержит информацию о типе блока, его имени, а также обо всех параметрах его входов и выходов. Схема описания функционального блока в XML формате представлена на рис. 4.

```
<FBBlock BlockType="..." BlockTypeName="">
  <Inputs>
    <Contact Type="..." Name="..." />
    ...
  </Inputs>
  <Outputs>
    <Contact Type="..." Name="..." />
    ...
  </Outputs>
</FBBlock>
```

рис. 4 XML-схема описания блока в библиотеке функциональных компонентов.

Рассмотрим подробнее процесс создания управляющих программ и ее отладки. На первом этапе разработки управляющей программы для ПЛК выполняется добавление требуемых для разрабатываемой модели компонентов из панели меню функциональных блоков, в окно модели, используя технологию «Drag & drop». Затем, функциональные блоки требуется связать между собой, а также определить и установить приоритеты выполнения каждого из них для реализации корректной последовательности их работы. Соединение блоков между собой осуществляется путем наведения курсора мыши на один из входов/выходов функционального блока, и ведения курсора с нажатой левой клавишей мыши до входа/выхода блока, с которым требуется произвести связь. Также разработанный инструментарий позволяет редактировать изломы линий связи, позволяя избежать их наложения на функциональные блоки, что делает визуализацию модели более наглядной и удобной.

Следующим этапом работы является настройка параметров работы всех блоков. Для каждого блока настройки индивидуальны. Настройку блока можно произвести двумя способами: установить требуемые параметры в панели настроек блока, или открыть специализированное окно настроек блока, кликнув два раза по блоку, настройкой которого требуется изменить. Основные параметры требуется настраивать в блоках входа/выхода. Функционал продукта позволяет настраивать следующие параметры входов/выходов:

- адрес порта или памяти;
- вид входа/выхода (BusCoupler, CommonPlcMemory, Port);
- 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 пустье 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 Ctl и разработка 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 приводится 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 рис. 5 показана схема взаимодействия редактора управляющих программ с ядром системы ЧПУ.

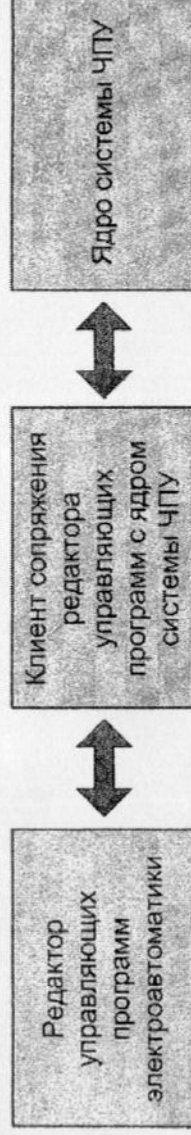


рис. 5 Схема взаимодействия редактора управляющих программ с ядром системы ЧПУ

При запуске разработанной управляющей программы для электроавтоматики происходит передача специфированного 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 рис. 6.

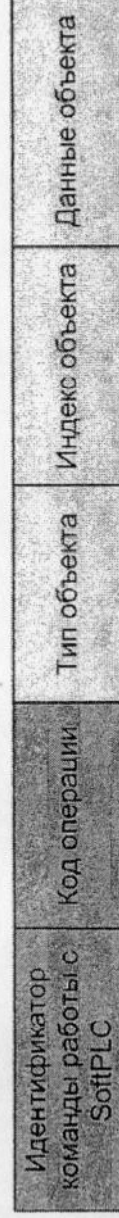


рис. 6 Обобщенная структура пакета данных при передаче в ядро функционального блока управляющей программы

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 результате проделанной работы был проведен анализ имеющихся средств по программированию логических контроллеров, а также решена задача по разработке гибкого, аппаратно-независимого инструментария для создания и отладки управляющих программ для электроавтоматики, интегрированного в систему ЧПУ AxiOMA Ctl. С помощью разработанного средства имеется возможность решать множество задач по программированию логических контроллеров на стандартизированном языке, независимо от используемой аппаратуры. Также имеются перспективы использования данного средства в учебных целях на специально оборудованных стендах обучения программированию электроавтоматики.

Литература

1. 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).
2. Грязнов И.Е., Кривошеин А.А., Иванов А.В. Средства программирования современных контроллеров. Известия ВолгГТУ.
3. Петров, И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / И. В. Петров; под ред. проф. В. П. Дьяконова. - М.: СОЛОН-Пресс, 2004. - 256 с.
4. Руководство пользователя по программированию ПЛК в CoDeSys V2.3. - Смоленск: ПК "Пролюг", 2004. - 423 с.
5. Христенсен, Дж. Х. Знакомство со стандартом на языке программирования PLC: IEC 1131-3 (МЭК 1131-3) Электронный ресурс] / Дж. Х. Христенсен. - [2004]. - Режим доступа: <http://www.asuip.ru>.
6. Сосонкин В.Л., Мартинов Г.М. Программирование систем числового программного управления. - М. Логос, 2008. - 344 с.
7. Сосонкин В.Л., Мартинов Г.М. Системы числового программного управления: учеб. пособие. - М. Логос, 2005. - 296 с.