

Методические аспекты проектирования программно-аппаратных систем логического управления технологическим оборудованием

Г.М. Мартинов,
д.т.н., проф.,
Р.А. Нежметдинов,
к.т.н., доц.
r.nezhmetdinov@stankin.ru,
МГУ «СТАНКИН», г. Москва

В работе предложена методика проектирования программно-аппаратных систем логического управления технологическим оборудованием. Рассмотрены отдельные шаги методики проектирования и результаты выполнения каждого из шагов. Приведён пример декомпозиции задач логического управления обрабатывающего центра наклонной компоновки CA-650.

Article proposes a technique for designing logical control systems. The individual steps of the design methodology and the results of each step are considered. The decomposition of the tasks of the logical control part of the machine tool center is presented.

Концепция построения систем логического управления с момента разработки первых устройств на базе релейных схем автоматики и до недавнего времени оставалась неизменной. Однако, на сегодняшний момент появились новые формы организации контроллеров (программно реализованные контроллеры, РАС системы), новые типы аппаратных вычислительных платформ (мобильные устройства, одноплатные компьютеры) и все большую популярность обретают идеи интеграции производственных ресурсов, при этом отсутствует методологическое решение проблемы построения систем логического управления на базе новых принципов. В этой ситуации каждый производитель контроллеров и систем автоматизации продвигает решения, основанные на собственной линейке продуктов, без учета особенностей функционирования систем управления в целом. Международные стандарты (IEC - International Electrotechnical Commission, ISO - International Organization for Standardization) определяют лишь границы, в рамках которых необходимо проектировать системы логического управления, но не сами базовые принципы.

Методика построения систем логического управления технологическим оборудованием

Процесс проектирования и реализации систем логического управления итеративен, сложен и неоднозначен в выборе методов и средств. В связи с этим возникает необходимость в разработке инструментария построения систем логического управления. В качестве такого инструментария может быть предложена методика, которая определяет фиксированный набор практических шагов, приводящих к требуемому результату. Методика не определяет теоретическое обоснование полученного результата, она только регламентирует действия разработчика. В качестве входных данных методика должна использовать техническое задание на разработку системы логического управления в котором содержатся все технические характеристики и функционал будущей системы. На выходе, после прохождения всех шагов методики имеем систему логического управления, соответствующую заявленным характеристикам. Полная методика построения систем логического управления представлена на рисунке 1, рассмотрим пошагово каждый из её этапов.

Адаптация моделей системы управления под конкретный объект. На начальном этапе производится анализ и адаптация моделей проектирования систем логического управления. В результате моделирования получим ряд основных модулей системы логического управления, определим их функционал и взаимосвязь модулей между собой, что позволит существенно упростить процесс дальнейшей разработки.

Разработка сетевой структуры системы управления. Процесс проектирования и разработки систем логического управления характеризуется: итеративностью, многоуровневостью и многоэтапностью. Для построения системы управления необходимо однозначное и полное описание функционирования проектируемой системы на определенном уровне представления в соответствии с техническим заданием, поставленными целями и требованиями. Для получения качественного и всестороннего описания системы логического управления необходимо определиться с выбором методов и средств формализованного описания системы. В распределенных системах управления и сетях применяется модульный принцип организации управления на основе иерархической многоуровневой схемы, в основе которого лежат понятия процесса, уровня управления, интерфейса и протокола связи. Основываясь на указанных понятиях предлагается на втором шаге разработки использовать инструментарий, который будет определять сетевую структуру системы логического управления. Сетевая структура должна однозначно определять: иерархические уровни из которых состоит система управления; оборудование, датчики и средства диагностики объекта управления; необходимые аппаратные вычислительные ресурсы; аппаратные модули ввода/вывода и протоколы их подключения к ядру логического управления; связи с локальными системами управления и системами управления верхнего уровня.

Выделение специализированных функций и команд. Система логического управления может применяться для автоматизации широкого круга технологического оборудования или технологических процессов, каждый из которых имеет специфические особенности и функционал. Для осуществления поддержки полного набора функций объекта управления необходимо определить набор специфических для данного объекта команд. Указанные команды должны быть систематизированы в таблицу и по ним должны быть определены условия их активации и деактивации.

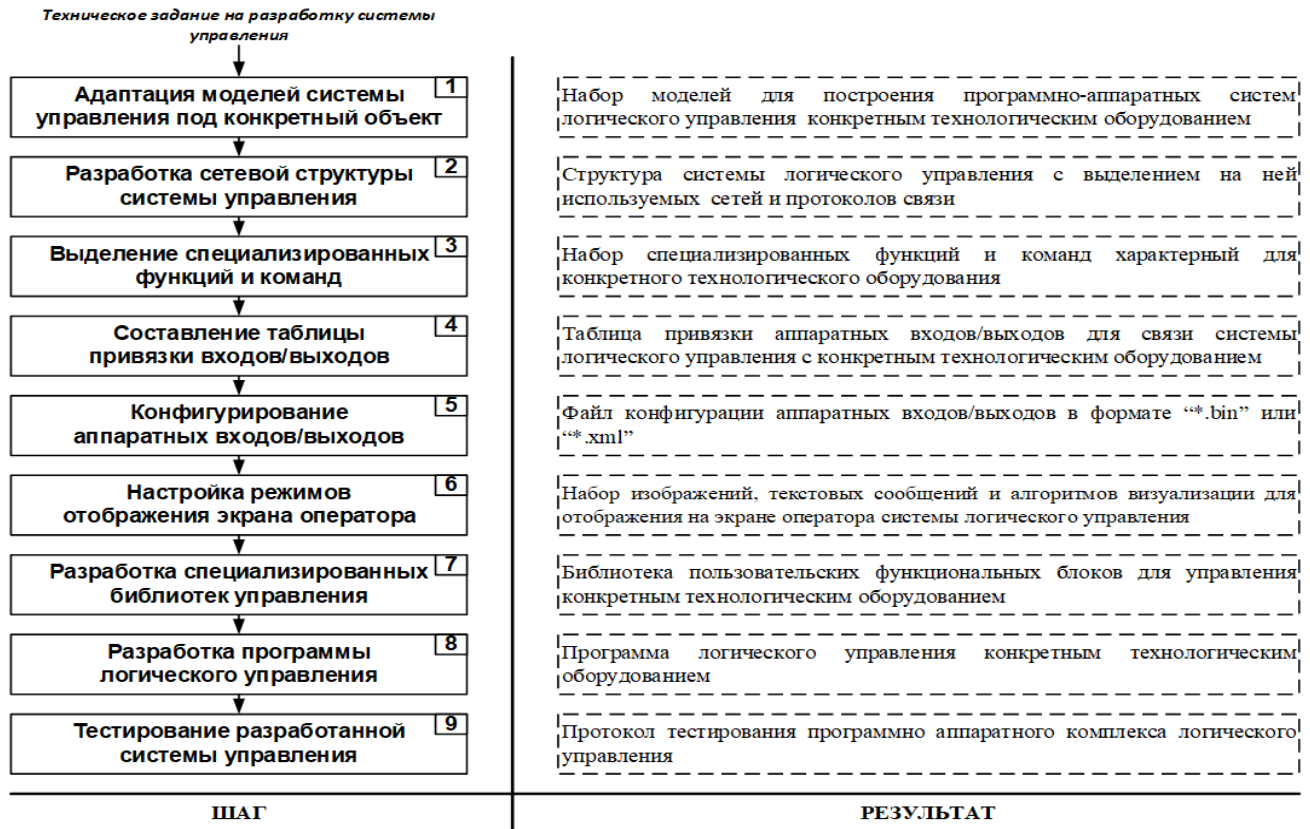


рис. 1. Методика построения системы логического управления

Составление таблицы привязки входов/выходов. Для выполнения следующего шага необходимо провести анализ принципиальной электрической схемы объекта управления с целью выделения узлов объекта и определения количества и типа аппаратных входов и выходов. Результатом анализа должна стать таблица привязки аппаратных входов/выходов, в которая содержит следующие данные:

- количество головных модулей (модулей организации связи);
- количество слотов входов/выходов с их привязкой к конкретному головному модулю;
- тип слотов входов/выходов (аналоговый, дискретный, подключения термосопротивлений и др.);
- адресация слотов входов/выходов.

Конфигурирование аппаратных входов/выходов. При работе с программой логического управления требуется определить зависимость между ячейками разделяемой памяти программно реализованного контроллера и аппаратными входами/выходами. Для конфигурирования аппаратных модулей входов/выходов необходимо знать следующее: структуру системы управления, топологию сети, протоколы коммуникации, а также производителя, модель и количество аппаратных модулей входов/выходов.

Настройка режимов отображения экрана оператора. Для взаимодействия с оператором технологического оборудования в системе логического управления должен быть предусмотрен экран. На сегодняшний момент разнообразие экранов оператора достаточно велико, от семисегментных ячеек способных отображать один символ, до панелей с высоким графическим разрешением и функцией “touchscreen”. Каждый из этих экранов нуждается в настройке режимов отображения, которые задаются программно в ядре системы логического управления. Механизм настройки определяется типом экрана оператора.

Для многоцветных графических панелей оператора должен быть сформирован графический интерфейс пользователя (англ. graphical user interface, GUI), который представляет собой доступные пользователю системные объекты и функции в виде графических компонентов (иконки, текстовые сообщения, меню, кнопки и т.д.). При этом оператор имеет доступ с помощью вспомогательных устройств (клавиатура, мышь и т.д.) ко всем отображаемым объектам экрана.

Обычно GUI экрана оператора – это отдельный программный продукт, со своей архитектурой и программными компонентами на котором есть выделенные области, которые могут быть изменены из программы логического управления. Механизм взаимодействия программы логического управления и GUI экрана оператора определяется разработчиками программного обеспечения графического интерфейса оператора.

Разработка специализированных библиотек управления. В качестве языка программирования систем логического управления в работе рассмотрен язык функциональных блоков, который позволяет разделять программу на подпрограммы, называемые пользовательскими функциональными блоками. Пользовательские функциональные блоки реализованы на базе стандартного набора функциональных блоков, объединенных в единый блок с выделением на нем входов и выходов (рисунок 2). Каждый пользовательский функциональный блок также может содержать в себе специализированные пользовательские функциональные блоки, при этом допускается до семи уровней вложенности.

К преимуществам реализации пользовательских функциональных блоков можно отнести следующие:

- Обозримость и удобство работы с кодом программы логического управления. Визуальные языки программирования, к которым относится и язык функциональных блоков, интуитивно понятны, однако становятся громоздкими и неудобными в использовании с ростом количества блоков. Пользовательские функциональные блоки позволяют

скрыть часть кода, относящуюся к конкретному узлу или технологическому элементу, что позволяет получить визуально хорошо структурированную программу логического управления и сократить значительный её объем.

- Возможность повторного применения набора библиотек функциональных блоков, реализующих логику работы с технологическим оборудованием или отдельными технологическими узлами. Такой подход позволяет повторно использовать ранее полученный программный код, что позволяет при наличии набора пользовательских библиотек существенно сократить время разработки программ логического управления.

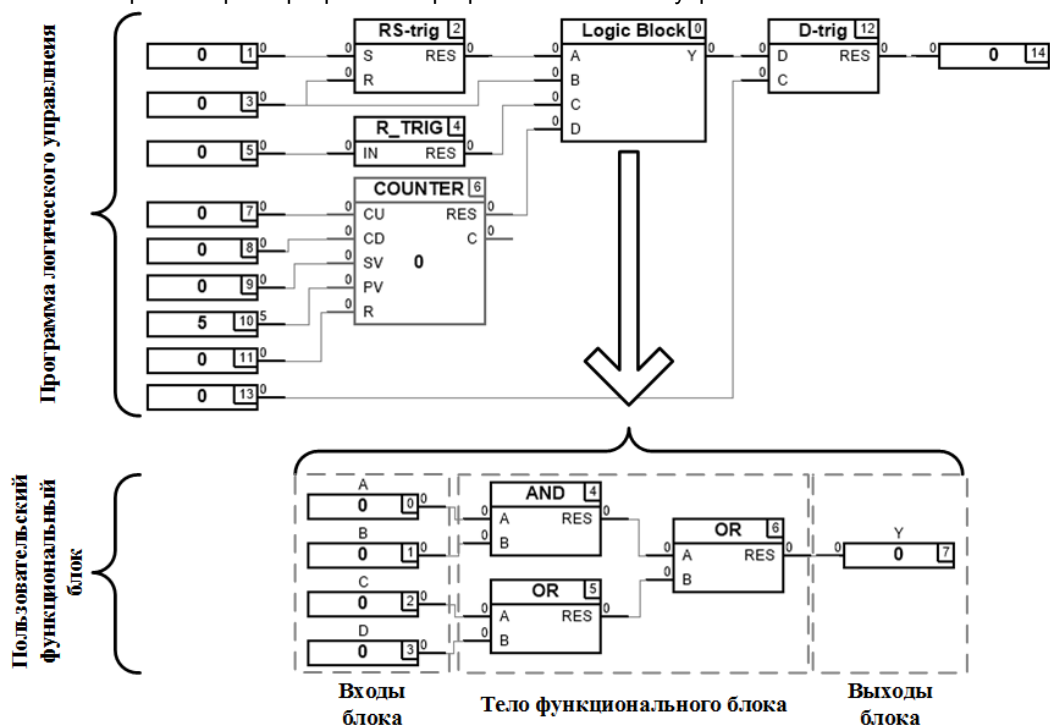


рис. 2. Создание пользовательского функционального блока в среде разработки программ логического управления

Разработка программы логического управления. Рассмотрим подробнее процесс создания и отладки программ логического управления. Программы логического управления разрабатываются исходя из технического задания, которое содержит алгоритмы и циклограммы работы объекта управления. Программа логического управления состоит из основной программы, которая хранится в отдельном файле и множества библиотек пользовательских подпрограмм, хранящихся в файлах.

Тестирование разработанной системы управления. Под тестированием системы логического управления будем понимать процесс обнаружения ошибок в программной и аппаратной частях путем исполнения программного кода на вычислительных ресурсах системы с входным набором тестовых данных. В ходе тестирования осуществляется сбор динамических характеристик системы управления в рамках работы на базе конкретной операционной системы, выявление дефектов, ошибок и отказов причиной которых становятся нерегулярные ситуации или аварийное прекращение работы.

Декомпозиция задачи логического управления технологическим оборудованием

Проектирование программного обеспечения для любой системы управления должно начинаться с изучения технического задания в котором указаны основные критерии и характеристики будущего программного продукта. Определение функций системы осуществляется по результатам анализа технического задания и обсуждения с заказчиком особенностей работы технологического процесса и оборудования.

Функции программы логического управления сводятся к выполнению цикла работы логического контроллера: принять входные сигналы, обработать алгоритм управления и выдать выходные сигналы. Для успешного выполнения указанного функционала необходимо реализовать:

- первичную обработку и верификацию входных сигналов, в том числе аналоговых;
- программную разработку алгоритмов логического управления;
- формирование выходных сигналов, в том числе аналоговых;
- поддержку обмена данными с системами управления верхнего уровня.

Рассмотрим этап программной реализации алгоритмов логического управления, который включает в себя этапы: декомпозиции исходной задачи на подзадачи, проектирования и математического моделирования, верификации математической модели, программной реализации на конкретном языке программирования и тестирования.

На начальном этапе реализации программы производится декомпозиция исходной задачи управления. Декомпозиция позволяет заменить решение одной сложной задачи решением ряда взаимосвязанных задач меньшего объема и позволяет анализировать любой объект управления как сложный, состоящий из отдельных взаимосвязанных подсистем, которые могут быть также разделены на части.

При проведении декомпозиции необходимо руководствоваться следующими правилами:

- разделение задач производится по уровням, при этом исходный объект управления находится на нулевом уровне;
- разделение производится по единому признаку внутри уровня;

- полученные подсистемы должны в полной мере характеризовать исходный объект управления или подсистему предыдущего уровня;
- глубина декомпозиции определяется исходя из квалификации программистов которые будут реализовывать полученные подсистемы.

В качестве примера декомпозиции рассмотрим вариант разбиения задач логического управления для обрабатывающего центра наклонной компоновки СА-650, представленный на рисунке 3.



рис. 3. Декомпозиция задач логического управления обрабатывающего центра наклонной компоновки СА-650

Каждая из полученных при декомпозиции задач в свою очередь может быть разделена на более мелкие, вплоть до «атомарных» задач, состоящих из одного действия. Каждая из отдельных подсистем объекта управления полученная в результате декомпозиции реализуется отдельным функциональным блоком, для формализации описания которого можно использовать различные математические аппараты. Выбор математического аппарат должен быть обусловлен имеющимися инструментальными средствами, которые позволят наиболее оптимально реализовать полученные математические модели. К числу наиболее часто применяемых при разработке программ логического управления математических аппаратов можно отнести следующие: Булева алгебра, автоматные модели, сети Петри, операторные формулы.

Выводы

Предложенная методика позволяет формализовать процесс проектирования систем логического управления и определяет фиксированный набор практических шагов, приводящих к требуемому результату. В качестве входных данных методика использует техническое задание на разработку системы логического управления, на выходе, после прохождения всех шагов имеем систему логического управления, соответствующую заявленным характеристикам

Литература

1. Расширение функциональных возможностей систем ЧПУ для управления механо-лазерной обработкой, Нежметдинов Р.А., Соколов С.В., Обухов А.И., Григорьев А.С., Автоматизация в промышленности. 2011. № 5. С. 49-53.
2. Применение технологии «клиент-сервер» при проектировании контроллера типа Soft PLC для решения логической задачи в рамках систем ЧПУ, Шемелин В.К., Нежметдинов Р.А., Автоматизация. Современные технологии. 2010. № 3. С. 20-24.
3. The russian multi-functional CNC system AxiOMA CONTROL: practical aspects of application, Martinova L.I., Kozak N.V., Nezhmetdinov R.A., Pushkov R.L., Obukhov A.I., Automation and Remote Control. 2015. Т. 76. № 1. С. 179-186.
4. Modular design of specialized numerical control systems for inclined machining centers, Martinov G.M., Nezhmetdinov R.A., Russian Engineering Research. 2015. Т. 35. № 5. С. 389-393.
5. Extending the functional capabilities of NC systems for control over mechano-laser processing, Nezhmetdinov R.A., Sokolov S.V., Obukhov A.I., Grigor'ev A.S., Automation and Remote Control. 2014. Т. 75. № 5. С. 945-952.
6. Принципы построения кроссплатформенного программно реализованного контроллера электроавтоматики систем ЧПУ высокотехнологичными производственными комплексами, Мартинов Г.М., Нежметдинов Р.А., Емельянов А.С., Вестник МГТУ Станкин. 2013. № 1 (24). С. 42-51.
7. Специфика построения редактора управляющих программ электроавтоматики стандарта МЭК 61131, Мартинов Г.М., Нежметдинов Р.А., Никишечкин П.А., Вестник МГТУ Станкин. 2014. № 4 (31). С. 127-132.