

УДК 004.424

Н.П. Винтоняк, В.А. Федотов, В.Д. Семенов

Методика проектирования программного обеспечения микроконтроллеров для быстродействующих систем управления

В статье рассмотрены методики проектирования программного обеспечения микроконтроллеров. Предложена методика, позволяющая реализовывать сложные программные алгоритмы управления, обладающие максимальным быстродействием с их полным дальнейшим описанием и документированием.

Ключевые слова: микроконтроллер, сложные алгоритмы управления, проектирование программного обеспечения, структурная схема программного обеспечения, диаграмма состояний.

История развития электронной техники показывает, что цифровые системы управления являются её перспективным направлением развития. Они позволяют значительно расширить эксплуатационные параметры электронной техники и заложить значительно более сложные алгоритмы управления, повышающие её гибкость. Применение микроконтроллеров (МК) в качестве основного узла цифровой системы управления позволяет улучшить массогабаритные показатели, расширить функциональные возможности, уменьшить влияние разброса параметров элементов на точностные характеристики системы управления и упростить настройку электронной техники в процессе производства. Применение цифровых систем управления позволяет снизить время разработки новой электронной техники. Это обусловлено возможностью использования ранее разработанного программного обеспечения (ПО) МК и его дальнейшую модернизацию на любом этапе разработки [1].

При реализации **сложных** алгоритмов управления (хранящих информацию о ходе выполнения алгоритма в памяти, т.е. зависящих от предыстории) разработчики ПО МК часто сталкиваются с затруднениями по его отладке и модернизации. Данные затруднения связаны с использованием несистемного подхода, при котором ПО разрабатывается интуитивно (эвристически) в ходе написания исходного кода. Несистемный подход влечет за собой сложность алгоритмизации и трудность внесения изменений в разработанный алгоритм. Используя несистемный подход, программисты описывают разработанное ПО с помощью блок-схем алгоритмов ЕСПД [2], которые в этом случае получаются громоздкими и затрудняют понимание сложных алгоритмов управления.

Современная постановка задачи проектирования ПО МК предполагает выполнение следующих требований к разрабатываемому ПО:

- возможность реализации сложных алгоритмов управления;
- обеспечение предсказуемого максимального быстродействия алгоритмов управления при обработке аппаратных сигналов, в масштабах времени от сотен наносекунд, до единиц микросекунд (например, сигналов перегрузки силовых ключей или сигналов высокоскоростных протоколов обмена данными);
- упрощение процесса отладки и модернизации проектируемого ПО с возможностью его повторного использования в других системах управления;
- возможность разделения работ по проектированию ПО между несколькими исполнителями;
- документирование и сертификация проектируемого ПО.

При реализации сложных алгоритмов управления прибегают к системным подходам [3], которые предполагают первоначально проводить формализованный анализ проектируемого ПО, и только после этого приступать к созданию исходного кода ПО. Системные подходы проектирования ПО широко представлены в различных методиках проектирования ПО [4–7]. Одной из таких методик является SWITCH-технология, разработанная А.А. Шальто [4], которая относится к парадигме автоматического программирования. В SWITCH-технологии ПО представляется в виде схемы взаимосвязанных конечных автоматов, проектируемых по графам переходов (диаграммам состояний). Особенности данной методики позволяют проектировать сложные алгоритмы управления и установить

однозначную связь между графом переходов и исходным кодом ПО. Также повышается «наблюдаемость» ПО (состояние конечного автомата отслеживается с помощью одной переменной), что ведет к упрощению его отладки и сертификации. Эта технология позволяет изначально «правильно» проектировать ПО, а не отлаживать его до тех пор, пока оно не станет «правильным». Успешность данной технологии описана в отзывах [8, 9] на работу [4] и в многочисленных статьях о её применении [10].

Несмотря на достоинства SWITCH-технологии, у неё есть ограничения при применении. Так как данная технология разработана для систем логического управления, в частности для программируемых логических контроллеров (ПЛК), она не обеспечивает максимально возможное быстрое действие алгоритмов управления работающих на МК. В данной технологии предполагается, что считывание входных воздействий конечного автомата производится в основном цикле ПО методом опроса, а их обработка производится на следующем такте основного цикла ПО [11]. Временные задержки на обработку входных воздействий вводятся самим алгоритмом ПО, разработанным с помощью SWITCH-технологии [12], т.к. все воздействия обрабатываются на следующем такте основного цикла ПО. Модификация SWITCH-технологии для разработки ПО МК [5], в которой для каждого сообщения выделяется три этапа, после прохождения которых сообщение поступит на вход конечного автомата, ведет к упрощению отладки ПО [13], но еще больше увеличивает время реакции МК на различные события. Методики, описанные в [4] и [5], целесообразно применять для разработки ПО, реализующего «небыстродействующее» логическое управление устройством.

Метод программной реализации дискретных управляющих автоматов во встраиваемых системах управления [6], адаптированный к задачам быстродействующего управления оборудованием в реальном времени, учитывает только уровень логического управления оборудованием, но не рассматривает уровень прямого цифрового управления оборудованием [6], что влечет за собой трудности по отладке и модернизации проектируемого ПО.

Унифицированный язык моделирования UML [7], использующий графические обозначения для создания UML-модели, помимо множества прочих диаграмм, также использует диаграммы состояний для представления различных объектов и их реакции на возникающие события. Он является языком общего назначения, и совместим со всеми языками программирования. Однако UML обладает некоторой избыточностью, сильной взаимосвязью с парадигмой объектно-ориентированного программирования, нечеткостью описания последовательности действий при проектировании ПО и отсутствием рекомендаций по его использованию в ПО МК. Это влечет за собой трудности по его адаптации для проектирования ПО МК.

Также существуют другие методики проектирования ПО, но, в силу отсутствия их подробного описания или невозможности их адаптации для проектирования ПО МК, они не получили широкого распространения и не рассматриваются в данной статье.

Отсутствие методики проектирования ПО МК для быстродействующих систем управления, отвечающей всем современным требованиям, делают её разработку актуальной.

Указанная цель достигается методикой проектирования программного обеспечения микроконтроллеров для быстродействующих систем управления, описанной в [14], которая относится к парадигмам автоматного и событийно-ориентированного программирования. Методика предполагает проводить проектирование ПО в три этапа, которые можно назвать:

- этап эскизного проекта ПО;
- этап оформления исходного кода ПО и его отладки;
- этап документирования ПО и его дальнейшей сертификации.

На первом этапе проектирования ПО сначала создается программно-аппаратная структурная схема (рис. 1, а), которая разрабатывается с учетом принципиальной схемы объекта управления. Принципиальная схема обуславливает сигналы, поступающие и отправляемые МК. Алгоритм разработки программно-аппаратной структурной схемы представляет собой выполнение следующих операций. На программно-аппаратной структурной схеме отображаются узлы объекта управления, с которыми взаимодействует МК (например, кнопка, двигатель, органы индикации), без собственных линий связи, и выделяется область ПО. Элементы структурной схеме ПО, помещенные в область ПО, являются программным кодом, а элементы, которые находятся вне этой области, являются аппаратными узлами объекта управления соединённого с МК. За пределы программной области, вы-

деленной прямоугольником, выводятся программно-аппаратные связи (пронумерованные штрихпунктирные линии). Они представляют собой выделенные аппаратные узлы МК (порты ввода / вы-вода, таймеры, аналого-цифровые преобразователи и др.). Программно-аппаратные связи демонстрируют взаимодействие ПО МК с аппаратной частью объекта управления. Внутри программной области располагаются блоки ПО, которые служат для представления в программе некоторого объекта, обладающего определенными свойствами, поведением и отношениями с другими объектами и их функциями. Представляемый блоком ПО объект может быть как внешним по отношению к МК (например, кнопка, двигатель, органы индикации и т.п.), так и внутренним (например, таймер, модуль АЦП, модуль ШИМ и т.п.). Внутри области ПО находится «системный» блок ПО, не отображаемый на структурной схеме ПО, который реализует алгоритм работы главной функции исходного кода ПО и координирует работу остальных блоков ПО. Блоки ПО связаны между собой внешними сообщениями (пронумерованные сплошные линии) и запросами (пронумерованные пунктирные линии), которые реализуются в исходном коде ПО в виде отдельных функций. Внешние сообщения позволяют блокам взаимодействовать друг с другом и осуществлять заданный алгоритм управления, а запросы позволяют передавать данные между блоками (текущее время, ток, напряжение, и т.п.).

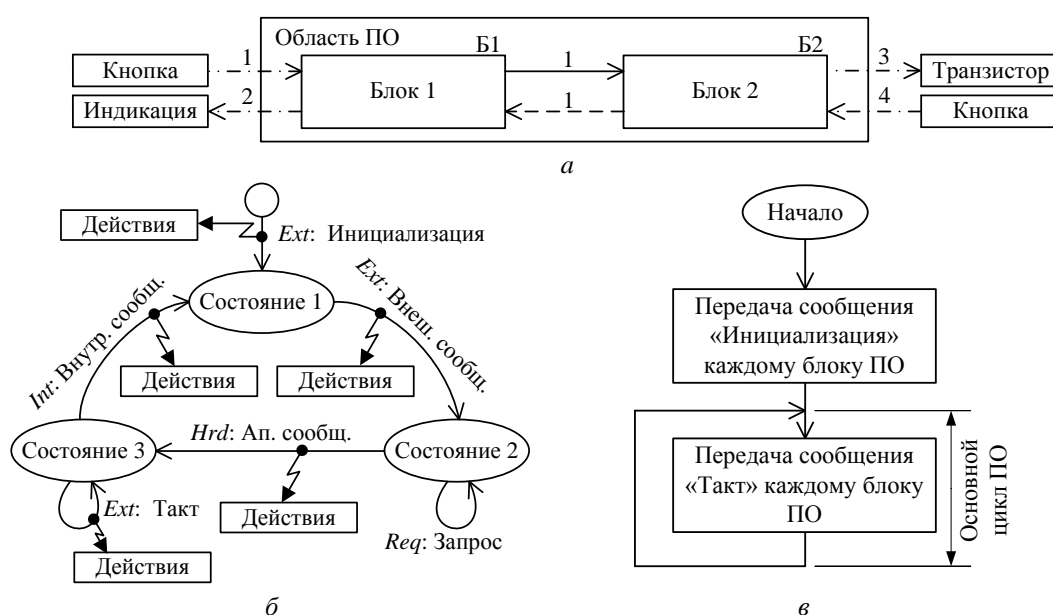


Рис. 1. Результаты эскизного проекта ПО: а – пример программно-аппаратной структурной схемы; б – пример диаграммы состояний блока ПО; в – алгоритм работы «системного» блока ПО

В дальнейшем для каждого блока ПО, входящего в структурную схему ПО, разрабатывается диаграмма состояний (рис. 1, б), описывающая процесс изменения внутреннего состояния каждого блока ПО. Блок ПО хранит в себе информацию о предыстории выполнения алгоритма, что позволяет реализовывать сложные алгоритмы управления. На диаграмме состояний отображаются возможные состояния блока ПО (овал с названием состояния), начальное состояние (круг), отображающее состояние МК после включения/сброса, внешние сообщения (указанные на структурной схеме ПО и начинающиеся с надписи «Ext:») и аппаратные сообщения (начинаются с надписи «Hrd:»), вызываемые аппаратными прерываниями МК. Также на диаграмме состояний отображаются внутренние сообщения (начинаются с надписи «Int:»), которые возникают при выполнении их условия и отображают события, возникающие внутри блока (достижение током заданного предела, конец временной задержки и т.п.). Каждое сообщение вызывает изменение состояния блока ПО (переход), за исключением «петли» (переход осуществляется на текущее состояние). Для запуска блока ПО используется специальное внешнее сообщение «Инициализация», посылаемое «системным» блоком ПО, которое позволяет блоку ПО выйти из начального состояния. Для вызова внешних сообщений других блоков ПО или описания действий, выполняемых при переходах, на диаграмме состояний отображаются прямоугольные области с пронумерованными операциями, на которую указывает «молния» (зигзагообразная стрелка) выходящая из соответствующего перехода. Направления переходов указываются прямыми или изогнутыми стрелками. Также на диаграмме состояний отобража-

ется специальное внешнее сообщение «Такт», посылаемое «системным» блоком ПО, которое проверяет условия внутренних сообщений и позволяет выполнять действия в состояниях.

На втором этапе проектирования ПО, по структурной схеме ПО и диаграммам состояний блоков ПО оформляются заголовочные файлы и файлы исходного кода в используемой среде разработки ПО МК. Каждому блоку ПО назначается отдельный заголовочный файл и файл исходного кода. «Системный» блок ПО помещается в основную функцию исходного кода ПО (main.h и main.c). Для реализации диаграммы состояний в исходном коде ПО используется оператор «switch» языка Си. В качестве выражения оператора «switch» используется переменная состояния блока ПО, а в качестве меток перечисляются возможные состояния блока ПО. Каждое сообщение и запрос реализуется с помощью отдельной функции, в которой используется оператор «switch», проверяющий текущее состояние (таким образом реализуется передача потока выполнения программы при передаче сообщения). С помощью оператора «switch» достигается однозначная взаимосвязь разработанных диаграмм состояний и исходного кода ПО. В процессе инициализации «системный» блок ПО поочередно вызывает специальные внешние сообщения «Ext: Инициализация» всех блоков ПО, а в основном цикле ПО поочередно вызывает специальные внешние сообщения «Ext: Такт» всех блоков ПО (рис. 1, в). Программная реализация разработанного алгоритма ПО и его отладка выходит за рамки данной статьи.

На третьем этапе проектирования ПО структурная схема ПО, диаграммы состояний блоков ПО и исходный код ПО служат основой для документирования разработанного ПО. К ним добавляются ряд таблиц с описанием реализации аппаратных связей, внешних, внутренних и аппаратных сообщений в исходном коде ПО, а также дополнительное описание используемых типов данных, констант, переменных, вспомогательных функций и т.п. В дальнейшем проводится сертификация ПО. Процесс документирования ПО и его сертификация также в данной статье не рассматриваются.

Практическая ценность предлагаемой методике была подтверждена при разработке систем управления силовыми преобразователями электроэнергии и ряде других проектов при реализации алгоритмов логического управления, в том числе в [14–18].

Методика проектирования программного обеспечения микроконтроллеров для быстродействующих систем управления позволяет:

1. Реализовывать сложные алгоритмы управления за счет использования структурной схемы ПО и диаграмм состояний блоков ПО.
2. Обеспечить предсказуемое максимальное быстродействие алгоритмов управления при обработке аппаратных сигналов за счет уменьшения программных задержек при изменении состояний блоков ПО, использования аппаратных сообщений и действий, выполняемых при переходах.
3. Установить однозначную взаимосвязь проектируемого ПО и исходного кода ПО, что упрощает его отладку, модернизацию, повторное использование и автоматизацию процесса проектирования ПО.
4. Разделить задачу разработки и реализации ПО МК между исполнителями за счет использования независимых блоков ПО, связывающихся между собой внешними сообщениями и запросами, отраженными на структурной схеме ПО.
5. Документировать разработанное ПО, используя структурную схему ПО, диаграммы состояний блоков ПО и дополнительное описание, что упрощает дальнейшую сертификацию ПО.

В предлагаемой методике сформулированы четкие правила проектирования ПО, которые в дальнейшем можно перенести в специализированную среду разработки. Такая среда разработки позволит создавать эскизный проект ПО, из которого будет автоматически генерироваться исходный код ПО и вся необходимая документация.

Литература

1. Мелешин В.И. Управление транзисторными преобразователями электроэнергии / В.И. Мелешин, Д.А. Овчинников. – М.: Техносфера, 2011. – 576 с.
2. ГОСТ 19.701–90. Схемы алгоритмов, программ данных и систем. Условные обозначения и правила выполнения. – Введ. 1992-01-01. – М.: Изд-во стандартов, сор. 1992. – 33 с. – (Единая система программной документации).
3. Перегудов Ф.И. Основы системного анализа / Ф.И. Перегудов, Ф.П. Тарасенко. – Томск: НТЛ, 1997. – 396 с.
4. Шальто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. – СПб.: Наука, 1998. – 628 с.

5. Применение Switch-технологии при разработке прикладного программного обеспечения для микроконтроллеров. Ч. 1: [Электронный ресурс]. – Режим доступа: http://kit-e.ru/articles/circuit/2006_11_164.php, свободный (дата обращения: 28.04.2015).
6. Козаченко В.Ф. Эффективный метод программной реализации дискретных управляющих автоматов во встроенных системах управления. [Электронный ресурс]. – Режим доступа: http://motorcontrol.ru/publications/state_mashine.pdf, свободный (дата обращения: 28.04.2015).
7. UML [Электронный ресурс]. – Режим доступа: <http://www.omg.org/spec/UML/>, свободный (дата обращения: 28.04.2015).
8. Герр Р. Новый поворот // PC MAGAZIN/RUSSIAN EDUTION. – 1998. – №10. – С. 88–90.
9. Лашманкин А. Возрождение автоматов // Компьютер дома и на работе. – 2001. – №23. – С. 50.
10. Кафедра «Технологии программирования»: О нас: Шалыто Анатолий Абрамович [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/aboutus/shalyto/>, свободный (дата обращения: 28.04.2015).
11. Вавилов К.В. Что плохого в неавтоматном подходе к программированию контроллеров? / К.В. Вавилов, А.А. Шалыто // Промышленные АСУ и контроллеры. – 2007. – №1. – С. 49–51.
12. Кафедра «Технологии программирования»: курсовые проекты: Интеграция механизма обмена сообщениями в Switch-технологию [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/projects/memech/>, свободный (дата обращения: 28.04.2015).
13. Применение SWITCH-технологии при разработке прикладного программного обеспечения для микроконтроллеров. Ч. 3. Обмен сообщениями и таймеры [Электронный ресурс]. – Режим доступа: http://kit-e.ru/articles/circuit/2007_1_146.php, свободный (дата обращения: 28.04.2015).
14. Винтоняк Н.П. Источник питания устройства тепловых экспресс-испытаний радиоэлементов с микропроцессорной системой управления / Н.П. Винтоняк, В.А. Федотов, В.Д. Семенов // Итоги научно-исследовательских работ курсового проектирования студентов 1–6 курсов кафедры промышленной электроники: матер. ежегод. науч.-практ. конф. / под ред. В.Д. Семенова. Вып. 6. – Томск: Том. гос. ун-т систем упр. и радиоэлектроники, 2012. – С. 6–22.
15. Разработка программного обеспечения моточного станка Roller DX7 / Н.П. Винтоняк, Р.Г. Калинин, В.А. Федотов, В.Д. Семенов // Научная сессия ТУСУР–2014. – Т. 2. – С. 195–199.
16. Винтоняк Н.П. Устройство для тепловых испытаний радиоэлементов на основе элементов Пельтье / Н.П. Винтоняк, В.А. Федотов, В.Д. Семенов // Научная сессия ТУСУР–2013. – Т. 2. – С. 180–182.
17. Калинин Р.Г. Выбор корректирующего звена резонансного преобразователя на основе экспериментальной АЧХ по управляющему воздействию / Р.Г. Калинин, А.В. Кобзев, В.Д. Семенов, В.А. Федотов // Доклады ТУСУРа. – 2014. – № 1. – 143 с.
18. Калинин Р.Г. Эффективность нагрева медного провода в зазоре индуктора поперечного магнитного поля для зачистки от лаковой изоляции / Р.Г. Калинин, В.Д. Семенов, В.А. Федотов // Доклады ТУСУРа. – 2014. – № 1(31). – 74–77 с.

Винтоняк Никита Павлович Для редакции:– моб. тел.: 9539222477

Аспирант каф. промышленной электроники ТУСУРа

Тел.: +7 (953-9) 22-24-77

Эл. почта: nic-nic91@yandex.ru

Федотов Владимир Александрович

Зав. лаб. ГПО каф. промышленной электроники ТУСУРа

Тел.: +7 (952-8) 88-46-13

Эл. почта: fva@vipelec.com

Семенов Валерий Дмитриевич

Канд. техн. наук, профессор каф. промышленной электроники ТУСУРа

Тел.: +7 (913-8) 21-22-92

Эл. почта: svd@ie.tusur.ru

Belichenko V.P., Zapasnoy A.S., Shestakov P.V.

Control of interference energy fluxes in the evanescent fields of the radiating systems

The formation process of energy fluxes in the evanescent fields of several spatially distributed elementary radiators substantially differs from the one in case of an isolated radiator. In particular, the complex interference energy flux is formed, which has a significant influence on both the structure of the stored electromagnetic energy, and the radiation power of these radiating systems. The study demonstrates the possibility of the effective control of the specified energy fluxes which provides maximizing the radiation power. We suggested that there is possibility of applying these control to problems in the near-field sensing media and objects.

Keywords: Radio frequency sounding, localized systems of the radiators, evanescent fields