

4. *Tommaso Bresciani*. Modeling, identification and control of a quadrotor helicopter // Master thesis, Lund University. 2008.
5. *Luis Rodolfo García Carrillo, Alejandro Enrique Dzul López, Rogelio Lozano, Claude Pégard*. Quad Rotorcraft Control // Springer London Heidelberg New York Dordrecht. 2012.
6. *Колесников А.А.* Синергетическая концепция системного синтеза: единство процессов самоорганизации и управления // Известия ТРТУ. – 2006. – № 6 (61). – С. 10-38.
7. *Веселов Г.Е., Скляр А.А., Скляр С.А.* Синергетическое управление траекторным движением гусеничного робота // Известия ЮФУ. Технические науки. – 2012. – № 4 (129). – С. 187-193.
8. *Веселов Г.Е.* Синергетический подход к синтезу иерархических систем управления // Известия ТРТУ. – 2006. – № 6 (61). – С. 73-84.

Статью рекомендовал к опубликованию д.т.н., профессор С.В. Тарарыкин.

Веселов Геннадий Евгеньевич – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: gev@sfedu.ru; 347900, г. Таганрог, ул. Чехова, 2; тел.: 88634360450; факультет информационной безопасности; декан.

Скляр Андрей Анатольевич – e-mail: s.andrey.88@mail.ru; тел.: 89612957403; кафедра СиПУ; аспирант.

Скляр Сергей Анатольевич – mail: predator86@mail.ru; тел.: 89034026114; кафедры СиПУ; аспирант.

Veselov Gennady Evgen'evich – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: gev@sfedu.ru; 2, Chekhov street, Taganrog, 347900, Russia; phone: +78634360450; college of informational security; dean.

Sclyarov Andrey Anatolevich – e-mail: s.andrey.88@mail.ru; phone: +79612957403; the department of SCP; postgraduate student.

Sclyarov Sergey Anatolevich – e-mail: predator86@mail.ru; phone: +79034026114; the department of SCP; postgraduate student.

УДК 621.52.3

В.И. Лапшин, А.А. Рыбаков, С.П. Котиков, Ю.П. Ларионов

МОДЕЛИРОВАНИЕ ПОДВИЖНОГО ЗАПРАВОЧНОГО КОМПЛЕКСА

Разрабатывая сложные многопроцессорные распределенные устройства, приходится сталкиваться с таким фактором, как трудоемкость отладки взаимодействия составных частей в режиме реального времени. Кроме того, приступить к отладке программного обеспечения можно только имея изготовленные устройства, и очень часто на этапе отладки выясняется, что из-за просчетов в проектировании необходимо: либо изменить схемотехнику того или иного устройства управления, либо внести изменения в работу периферийных (управляемых) устройств.

Учитывая тот факт, что разрабатываемый комплекс содержит несколько функционально законченных контроллеров было решено разработать систему моделирования комплекса, позволяющую выполнить отладку «штатного» ПО контроллеров, входящих в его состав, без их реального изготовления. Это позволяет выполнить проверку правильности заложенных в основу комплекса схемотехнических решений и алгоритмов работы.

Моделирование; заправочный комплекс; микропроцессорные распределенные устройства.

V.I. Lapshin, A.A. Rybakov, S.P. Kotikov, U.P. Larionov

MODELLING OF THE MOBILE

Developing sophisticated multiprocessor distributed devices have to deal with such laborious debugging of components interaction in real time. In addition, it is possible to start software debugging only having the made devices, and very often at a stage of debugging it becomes clear that because of miscalculations in design it is necessary to either change the circuitry of a control devices, or to make changes to the work of peripheral (controlled) devices.

Given the fact that the developed complex contains several functionally complete controllers decided to develop a modeling engine of the complex that allows you to debug "regular" controller software that are part of it, without their actual production. It allows to execute check of correctness put in a basis of a complex of circuitry decisions and algorithms of work.

Modeling, filling complex; multiprocessor distributed devices.

Подвижный заправочный комплекс (ПЗК) должен обеспечивать дистанционное автоматизированное управление технологическим оборудованием объекта автоматизации при проведении заправочных и сливных работ.

ПЗК состоит из следующих составных частей (агрегатов):

- ◆ заправочной автоцистерны 15Г95М / 96М / 95 / 96 (ЗАЦ М / ЗАЦ);
- ◆ подвижной заправочной станции 15Г105М / 106М (ПЗС);
- ◆ машины управления заправкой 15В59П1М (МУЗ).

Каждый из агрегатов оснащен:

- ◆ различными технологическими элементами (баками, трубопроводами, насосами, клапанами и т.д.);
- ◆ элементами контроля (датчиками давления, уровня и температуры, расходомерами и т.д.);
- ◆ элементами управления (реле, электропневмопреобразователями и т.д.);
- ◆ элементами связи между агрегатами;
- ◆ контроллером, осуществляющим сбор технологических данных и управление технологическим оборудованием по заданным алгоритмам.

Кроме того, в состав пультов управления, контроля и разрешения МУЗ входят органы управления и индикации.

Программное обеспечение устройства заправки при выполнении операций заправки/слива, управляя протеканием процесса, должно однозначно определять возникновение нештатных ситуаций и выполнять необходимые действия для предотвращения аварийных случаев. Для составления максимально полных и точных алгоритмов операций, включающих реакции на все состояния совокупности показаний датчиков, которые может иметь система, и отслеживания всех протекающих процессов необходимо производить наиболее полную и приближенную к реальности имитацию всех гидравлических составляющих и взаимодействующих между собой компонентов системы. Гидравлическая подсистема содержит достаточно большое число клапанов, которые должны открываться/закрываться в строго определенном порядке при выполнении конкретной операции заправки/слива. Как и любое механическое устройство, клапаны имеют определенное время срабатывания. При этом поведение всех гидравлических составляющих должно четко определяться. Это означает, что нужно точно знать, какое реле и в течение какого времени должно срабатывать при открытии клапана, какой датчик должен при этом включаться и т.д. и т.п.

Для решения задач отладки, составления и отработки четкой реакции на срабатывание какого-либо датчика или реле при проведении технологически сложных процессов заправки/слива и в связи с необходимостью наиболее полной проработки всех возникающих ситуаций была создана программная модель, которая моде-

лирует все электрические элементы системы, управляющие узлы (микроконтроллеры) и гидравлические составляющие, т.е. весь набор клапанов, датчиков, реле и насосов, присутствующих в системе.

Программная модель разработана с использованием двух основных частей:

- ◆ программного эмулятора микроконтроллера семейства i8051 (в частности используемого в контроллерах ПЗК российского микроконтроллера 1882BE53) – для моделирования электронной подсистемы [1];
- ◆ программного имитатора периферийного оборудования (оконечных устройств) – для моделирования гидравлической подсистемы.

Программный эмулятор микроконтроллера (далее по тексту – эмулятор) состоит из следующих составных частей:

- ◆ интерактивная оболочка (в виде EXE-файла), обеспечивающая оконный диалоговый режим работы с пользователем;
- ◆ динамически подключаемый модуль (в виде DLL-файла) эмуляции собственно микроконтроллера и полностью эмулирующий все алгоритмы работы микроконтроллера с поддержкой всех внутренних ресурсов микроконтроллера (ALU, регистров, счетчиков, UART, SPI, и т.д.);
- ◆ динамически подключаемые модули (в виде DLL-файлов) имитации периферийного оборудования, подключаемого к микроконтроллеру по «общей» шине данных (шине данных отображаемой на пространство внешней памяти данных микроконтроллера) и обеспечивающие необходимый стык эмулятора с программой имитатора периферийного оборудования.

Динамически подключаемый модуль эмуляции обеспечивает:

- ◆ возможность загрузки программного кода в память программ микроконтроллера из файла формата IntelHEX;
- ◆ поддержку всей системы команд и адресного пространства (внутренняя прямая/внутренняя косвенная/внешняя память данных, память программ) до 64Кб;
- ◆ поддержку механизма работы таймеров (0,1 и 2), за исключением режима работы таймеров от внешних сигналов (цепей управления);
- ◆ поддержку механизма генерации прерываний с учетом приоритетности и маскирования;
- ◆ поддержку механизма обмена данными через UART и SPI с использованием эмуляции обмена через COM-порт или «сообщения Windows»;
- ◆ возможность отладки загруженного кода (выдача дампов памяти, установка точек останова, дизассемблирование участков кода и т.д.);
- ◆ возможность подключения независимого динамически подключаемого программного модуля имитации периферийного оборудования с его управлением со стороны микропроцессора через порты ввода/вывода или внешнюю память данных;
- ◆ возможность комплексирования эмуляторов в единую систему для моделирования устройства, содержащего несколько микроконтроллеров;
- ◆ возможность управления «синхронным» выполнением на нескольких копиях эмулятора, в пределах одной ПЭВМ, отладки «параллельно» выполняемого эмуляторами кода;
- ◆ настройку особенностей работы микропроцессора и режимов эмуляции программного кода.

Поддержка эмулятором UART и SPI позволяет объединять несколько эмуляторов для моделирования системы, содержащей несколько микроконтроллеров объединенными указанными протоколами, в этом случае эмуляторы могут быть

запущены как на одной, так и на нескольких ПЭВМ [2]. Для синхронизации работы подобной системы предусмотрена возможность синхронизации одним эмулятором, работающим в режиме MASTER и других, работающих в режиме SLAVE.

Динамически подключаемые модули имитации периферийного оборудования позволяют обеспечить эмулятору практически любой стык процессора в части взаимодействия с элементами управления, индикации и т.д. Использование механизма подключения данных модулей в виде DLL к эмулятору позволяет, не внося никаких изменений в сам эмулятор, а сменив лишь DLL, полностью подстроить эмулятор под необходимую поддержку «периферии» в будущих разработках.

Данное построение эмулятора позволяет:

- ◆ модифицировать любую из частей без изменения остальных с условием сохранения протоколов связи их между собой;
- ◆ подключать дополнительные модули эмуляции периферийного оборудования, в случае добавления в систему заправки дополнительных узлов (например, узлов с измененными алгоритмами работы);
- ◆ использовать эмулятор в качестве среды отладки программного кода.

Таким образом, программный эмулятор позволяет создать достаточно точную среду выполнения ПО, соответствующую разработанному контроллеру, в результате чего получается полнофункциональная модель управляющего узла контроллера со всеми необходимыми функциями, которую можно, с помощью некоторой надстройки и путём эмулирования электронной обвязки (шины данных) и каналов передачи данных, соединять с другими такими узлами, а также подключать к таким узлам всевозможные блоки индикации, приёма/передачи информации и т.п.

Программный имитатор периферийного оборудования (далее по тексту – имитатор) состоит из программы, имитирующей процесс работы гидравлики системы, т.е. включения и/или выключения реле и датчиков в соответствии с состоянием того или иного клапана. Кроме того, имитатор предоставляет пользователю набор элементов управления (в виде тумблеров и переключателей), посредством которых пользователь может взаимодействовать с системой, и окон индикации, в которых осуществляется индикация состояния всех компонентов системы (клапанов, реле, сухих контактов, кнопок и переключателей).

Данный имитатор написан под имитацию конкретных устройств с конкретно заданными техническими характеристиками: число и тип элементов управления и исполнительных элементов; число, тип и вид информации элементов индикации. Окно имитации одного из контроллеров представлено на рис. 1.

Имитатор позволяет также управлять временными задержками срабатывания того или иного объекта и содержит ряд описанных последовательностей срабатывания объектов для имитации необходимых операций заправки/слива, что позволяет проверить реакцию всей системы на возникновение тех или иных ситуаций.

В отсутствии реально изготовленного устройства данная программная модель позволила:

- ◆ описать модель поведения «реального» микроконтроллера 1882BE53;
- ◆ выполнить отладку программного кода с достаточной большой долей вероятности дальнейшей работы отлаженного кода на «реальном» микроконтроллере;
- ◆ отработать программные механизмы межпроцессорного взаимодействия;
- ◆ отработать алгоритмы управления исполнительными элементами;
- ◆ оценить объём и достаточность органов управления и индикации.

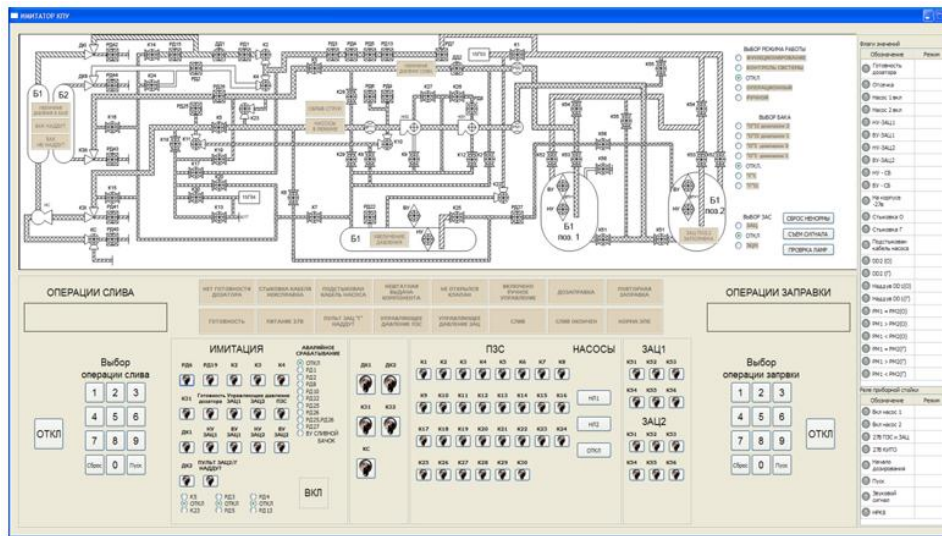


Рис. 1. Окно имитации

Таким образом, еще не имея реально изготовленного устройства, мы уже имеем возможность оценить правильность заложенных схмотехнических решений и при необходимости внести соответствующие изменения в схмотехнику или алгоритмы работы, что в конечном итоге ведет к снижению затрат на разработку устройства.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бояринов А.Е., Дьяков И.А. Архитектура микроконтроллеров семейства MCS-51: Конспект лекций. – Тамбов: Изд-во ТГТУ, 2005.
2. Бродин В.Б., Шагурин М.И. Справочник. Микроконтроллеры: архитектура, программирование, интерфейс. – М.: ЭКОМ, 1999.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Тютиков.

Лапшин Владимир Ильич – НИИ физических измерений и вычислительной техники; e-mail: info@niifvt.ru; г. Пенза, ул. Лермонтова, 3; тел.: 88412564520; генеральный директор.

Рыбаков Александр Александрович – зам. генерального директора по НИОКР.

Котиков Сергей Петрович – начальник НИО-2.

Ларионов Юрий Петрович – главный специалист.

Lapshin Vladimir Ilich – Research institute of physical measurements and computer technology; e-mail: info@niifvt.ru, 3, Lermontova street, Penza, Russia; phone: +78412564520; general director.

Rybakov Alexander Alexandrovich – deputy director general for innovation.

Kotikov Sergey Petrovich – head of second research branch.

Larionov Jury Petrovich – main specialist.