

Бойцов Григорий Викторович

Инженер-программист
ООО «НПП САТЕК плюс»
(Рыбинск, Россия)
E-mail: grig_b@mail.ru

Boitsov Grigoriy Viktorovich

Software Engineer
LLC "NPP SATEC plus"
(Rybinsk, Russia)
E-mail: grig_b@mail.ru

Петров Александр Викторович

Генеральный директор
ООО «НПП САТЕК плюс»
(Рыбинск, Россия)
E-mail: petrov@nppsatek.ru

Petrov Aleksandr Viktorovich

CEO LLC "NPP SATEC plus"
(Rybinsk, Russia)
E-mail: petrov@nppsatek.ru

Шаров Владимир Григорьевич

Кандидат физико-математических наук,
профессор, Рыбинский государственный
авиационный технический университет
(Рыбинск, Россия)
E-mail: svg45@mail.ru

Sharov Vladimir Grigoryevich

PhD in Physical and Mathematical Sciences,
Professor, Rybinsk State Aviation
Technical University
(Rybinsk, Russia)
E-mail: svg45@mail.ru

**АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ
РЕАЛИЗАЦИИ ПРОТОКОЛОВ
ВЗАИМОДЕЙСТВИЯ УСТРОЙСТВ
ВО ВСТРАИВАЕМЫХ
МИКРОКОНТРОЛЛЕРНЫХ
СИСТЕМАХ**

**AUTOMATIC IMPLEMENTATION
GENERATION OF INTERACTION
PROTOCOLS FOR DEVICES
IN EMBEDDED
MICROCONTROLLER SYSTEMS**

Аннотация. Авторы рассматривают разработанную программную среду, основанную на новом подходе к генерации программных средств, обеспечивающих взаимодействие устройств в распределенной микроконтроллерной среде. В основе предлагаемого подхода три высокоуровневых предметно-ориентированных языка описания коммуникационных протоколов. Система позволяет сократить время на разработку за счет автоматической генерации программного кода по высокоуровневому описанию, а также повысить надежность сгенерированного исполняемого кода.

Abstract. The authors consider the developed software system based on a new approach to the software generation that provides interaction between devices in a distributed microcontroller environment by a high-level description. The system is based on three high-level domain specific languages describing communication protocols. The system allows user to reduce development time through automatically generating code by a high-level description and also to increase the reliability of the generated executable code.

Ключевые слова: встраиваемые микроконтроллерные системы, генерация реализации протоколов, языки описания протокола

Keywords: embedded microcontroller systems, generation of protocol implementation, protocol description languages

Введение

Одной из современных тенденций развития вычислительных систем является выход за рамки персональных компьютеров и распространение вычислительных устройств во всех сферах жизни человека в виде встраиваемых систем и сетей. Разновидностью встраиваемых систем являются микроконтроллерные системы, основным вычислительным элементом в которых выступает микроконтроллер. При этом прослеживается тенденция увеличения гетерогенности распределенных микроконтроллерных систем, включения в состав разнородных компонентов, объединенных разнообразными средствами коммуникации. В этих условиях существенно усложняется задача реализации эффективного взаимодействия компонентов системы. В случае, когда для конкретного устройства отсутствует специализированная библиотека, поддерживающая протокол взаимодействия с другим устройством, реализация протокола предполагает значительные трудозатраты по низкоуровневому программированию, что увеличивает общий срок разработки и снижает надежность системы.

Среди работ, посвященных исследованию этой проблемы, можно выделить исследовательский проект z2z лаборатории LaBRI – University of Bordeaux [2], посвященный автоматической генерации программного кода, реализующего протокольный шлюз в телекоммуникационных системах. В статье [2] рассматривается проблема реализации протокольного шлюза, объединяющего различные протоколы (SIP for telephones, RTSP for televisions, X2D for thermostats, X10 for lamps) в системах «Умный дом». Данный инструмент позволяет сгенерировать программный код реализации протокольного шлюза, который представляет собой конвертер протоколов, однако не решает задачу генерации пользовательского высокоуровневого программного интерфейса для взаимодействия с каким-либо устройством встраиваемой микроконтроллерной системы.

В качестве другого аналога можно привести исследовательский проект "Melange" [4], целью которого является автоматическая генерация реализации сетевых протоколов и структур данных, а также верификация полученных реализаций для наиболее распространенных сетевых протоколов (Ethernet frame, IPv4, ICMP, TCP, UDP, SSH, DNS, DHCP). Основу подхода составляют два языка описаний (Meta Packet Language и Statecall Policy Language) и их трансляторы на язык OCaml.

Отметим, что оба аналога хотя и решают похожую задачу автоматической генерации реализации коммуникационных протоколов, но не учитывают специфики встраиваемых микроконтроллерных систем. В области разработки программного обеспечения для распределенных систем на базе микроконтроллеров проблема создания универсальных средств описания протоколов до настоящего времени не решена, что значительно тормозит развитие этого перспективного направления информационных технологий. Одной из причин непопулярности методов формального описания является отсутствие специализированных компиляторов с языков описания протоколов на некоторый промежуточный язык с учетом специфики микроконтроллерных систем. Таким образом, проблема разработки методов и программного обеспечения, позволяющих описывать на формальном языке взаимодействие элементов в микроконтроллерных распределенных системах, а также генерировать реализацию протокола по этому описанию, несомненно, является актуальной.

Основная часть

Целью настоящего исследования является разработка универсального инструмента, который позволяет на основании описания протокола на высокоуровневом предметно-ориентированном языке сгенерировать программный код, обеспечивающий взаимодействие устройств микроконтроллерной системы. При этом используется общий подход, аналогичный z2z и Melange, но его цель – генерация специализированной библиотеки, предоставляющей пользователю высокоуровневый интерфейс прикладного программирования (API) для взаимодействия устройств в распределенных встраиваемых системах.

Для реализации предлагаемого подхода разработано три предметно-ориентированных языка описания взаимодействия пользователя с устройством:

- Device Interface Specification (DIS), описывающий интерфейс взаимодействия с устройством;
- Library Interface Specification (LIS), описывающий интерфейс взаимодействия с библиотекой;
- Message Translation Specification (MTS), описывающий трансляцию сообщений из LIS в DIS, подробное описание которых можно найти в работе [1].

По описанию протокола взаимодействия и интерфейсов на этих языках генерируется библиотека, предоставляющая высокоуровневый прикладной интерфейс для взаимодействия с устройством. Общая схема взаимодействия пользователя с устройством и место языков описания протокола в системе отражены на рис. 1.

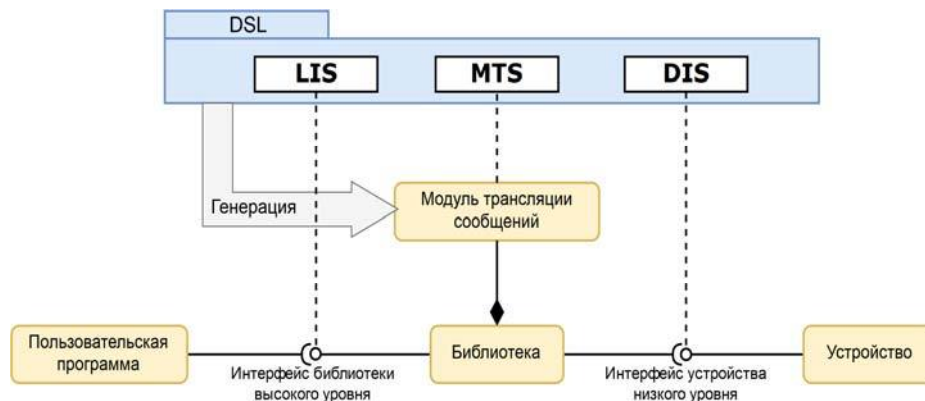


Рис. 1. Схема взаимодействия пользовательской программы с микроконтроллерным устройством

В данной статье остановимся подробнее на реализации конкретного примера с использованием интегрированной среды, которая была разработана с использованием фреймворков Xtext [4] и YAKINDU Statechart Tools [5].

В качестве примера опишем на трех разработанных языках взаимодействие микроконтроллера ATmega семейства AVR с высокоточным аналого-цифровым преобразователем ADS1258. Язык DIS позволяет задать требуемые параметры интерфейса взаимодействия с устройством для корректного обмена данными. Для этого необходимо описать информационную модель устройства и интерфейс взаимодействия.

Информационной моделью устройства называется совокупность признаков и параметров устройства микроконтроллерной системы, которая рассматривается как объект взаимодействия в распределенной среде. При описании информационной модели необходимо указать ее вид и параметры. Аналогово-цифровой преобразователь ADS1258 управляется посредством регистров, поэтому при описании на языке DIS указывается модель «регистры». Это означает, что с точки зрения пользователя устройство может быть представлено как последовательность регистров, а управление происходит посредством записи данных в определенный регистр или считывания значения регистра. В качестве параметров модели необходимо указать разрядность регистров. Имеется возможность в качестве адреса регистра и устанавливаемых значений указывать не числовые данные, а имена, это позволяет значительно повысить читаемость кода и удобство программирования.

Аналогово-цифровой преобразователь ADS1258 подключается по интерфейсу SPI, поэтому на языке DIS приводится полное описание интерфейса и его параметров, а также объявление констант для именования регистров и их значений. На рис. 2 представлено описание интерфейса взаимодействия с устройством и его информационной модели с использованием разработанного программного инструмента.

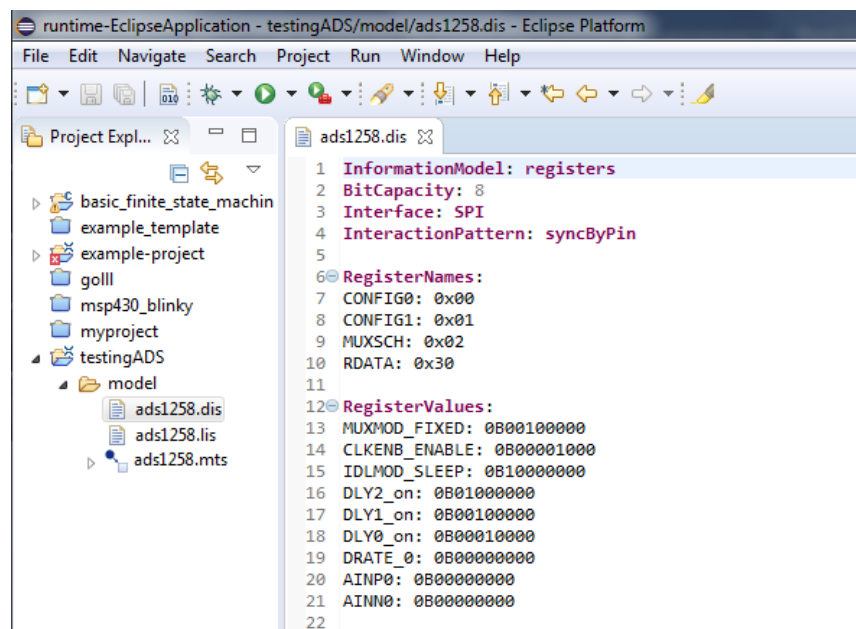


Рис. 2. Описание интерфейса взаимодействия с устройством

На языке LIS описывается интерфейс генерируемой библиотеки. Это гибко настраиваемый под нужды пользователя язык описания структур данных и сигнатур функций прикладного интерфейса библиотеки. На языке LIS специфицируются все функции и структуры данных, которые должен предоставлять пользователю API-интерфейс. Разрабатывая программу для микроконтроллера, взаимодействующего с

каким-либо устройством, пользователь может подключить сгенерированную API-библиотеку и вызывать высокоуровневые функции или манипулировать структурами данных. Таким образом, язык LIS предназначен только для внешней спецификации функций, а алгоритмы реализации этих функций описываются на языке MTS, в основе которого лежит модель конечных автоматов.

В качестве примера реализуем высокоуровневую функцию вывода напряжения, считанного с АЦП в Serial-порт микроконтроллера. В таком случае описание прикладного интерфейса библиотеки на языке LIS будет состоять из одной функции – `print_volt_AIN0()`, которая выводит напряжение с аналогового входа AIN0 в Serial порт. Поскольку функция не возвращает значений и у нее нет параметров, в объявлении каких-либо структур данных нет необходимости.

На языке MTS описывается алгоритм реализации функций, объявленных в LIS. При этом могут использоваться ранее сгенерированные функции, соответствующие описанной на DIS информационной модели устройства. Спецификация на языке MTS представляет собой композицию конечных автоматов, оперирующую LIS-спецификациями высокого уровня и DIS-спецификациями интерфейса устройства низкого уровня.

Конечный автомат представляет собой описание алгоритма, в котором используются уже сгенерированные функции на основе описания интерфейса устройства на DIS. Таким образом, вместо автомата значительного размера (из-за ручной реализации функций по передаче данных другому устройству) мы получаем компактный конечный автомат. Важным преимуществом такого подхода является универсальность, поскольку при смене архитектуры или интерфейса не требуется переписывать код под другой интерфейс и архитектуру, нужно лишь внести необходимые правки на языке DIS и заново сгенерировать библиотеку. Если устройств множество, то для каждого можно дополнительно сгенерировать высокоуровневую библиотеку и подключить ее к пользовательской программе.

На рис. 3 в качестве примера на языке MTS описана функция вывода напряжения в Serial порт через последовательность вызовов низкоуровневых функций `setRegisterValue` и `getRegisterValue` с учетом регистровой модели устройства. При переходе в состояние готовности к считыванию значения напряжения (Ready to Read Voltage) необходимо сначала сконфигурировать устройство. Поэтому для перехода в это состояние выполняется установка необходимых значений регистров, что отражено в коде над дугой-переходом после символа «/». Теперь для перехода в состояние готовности к выводу значения напряжения (Ready to Print) необходимо сначала считать его значение с уже сконфигурированного устройства, что отражено в коде у дуги перехода в это состояние. Полученное значение напряжения выводится в Serial-порт, после чего автомат переходит в конечное состояние.

При оценке надежности алгоритмов генерации и сгенерированного программного кода выполнялся статический анализ спецификаций и сгенерированного кода. Кроме того, с использованием JUnit были спроектированы и разработаны unit-тесты, в которых сгенерированный программный код сравнивался с эталоном. При динамической оценке надежности сгенерированного кода по модели Бернулли отправка и прием исчерпывающего набора одно-, двух- и трехбайтовых сообщений прошли корректно.

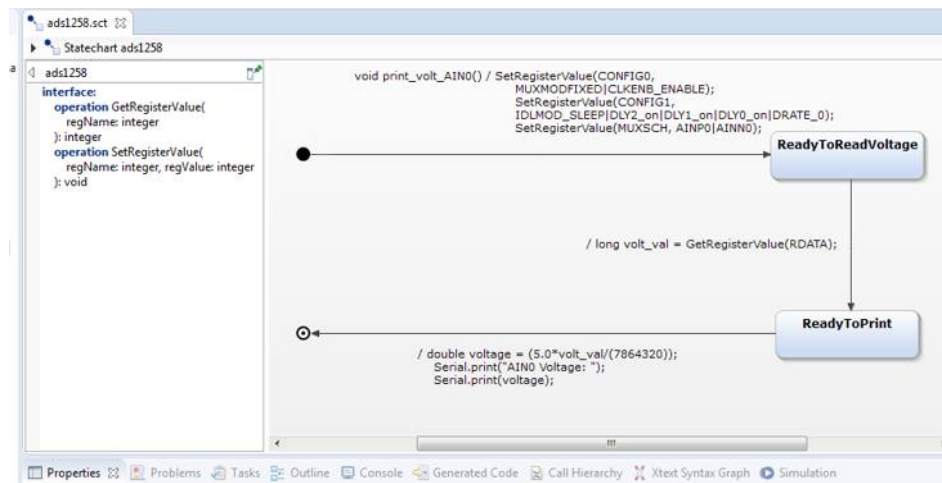


Рис. 3. Описание алгоритма реализации прикладной функции библиотеки на языке MTS

Выводы

В работе рассматривается разработанная программная система, основанная на предложенном новом подходе к генерации программных средств, обеспечивающих взаимодействие устройств в распределенной микроконтроллерной среде по высокоуровневому описанию. В основе системы – три высокоуровневых предметно-ориентированных языка описания протокола взаимодействия устройств в микроконтроллерных системах. Разработанная система позволяет сократить время на разработку и внесение изменений в исполняемый программный код, обеспечивающий взаимодействие с устройством, за счет его автоматической генерации по высокоуровневому описанию, а также повысить надежность сгенерированного исполняемого программного кода.

Литература

1. Шаров В.Г., Петров А.В., Бойцов Г.В. Об одном подходе к построению системы автоматической генерации программных средств, обеспечивающих взаимодействие микроконтроллера и электронных компонентов с фиксированной логикой работы // Актуальные проблемы инновационного развития: Международная научно-практическая конференция. М., 2017. С. 449–463.
2. Bromberg Yerom-David, Reveillere Laurent, Lawall Julia L., Muller G. Automatic Generation of Network Protocol Gateways // 10th ACM/IFIP/USENIX International Conference on Middleware Article No. 2, 2009, pp. 21–40. URL: <https://pages.lip6.fr/Julia.Lawall/middleware09.pdf>
3. Efftinge S. Xtext – Language Engineering Made Easy. URL: <http://xtext.itemis.com/>.
4. Madhavapeddy A., Ho A., Deegan T., Scott D., Sohan R. Melange: Creating a “Functional” Internet // EuroSys 2007: Proceedings of the 2007 conference on EuroSys., Lisbon, Portugal, 2007, pp. 101–114. URL: <https://www.tjd.phlegethon.org/words/eurosys07-melange.pdf>
5. Nyßen Al., Terfloth A. YAKINDU SCT – Domain-Specific Statecharts. EclipseCon 2012. URL: <https://www.itemis.com/en/yakindu/statechart-tools/>

References

1. Sharov V.G., Petrov A.V., Boitsov G.V. *Ob odnom podhode k postroeniiu sistemy avtomaticheskoi generacii programmnyh sredstv, obespechivaiushchih vzaimodeistvie mikrokontrollera i ehlektronnyh komponentov s fiksirovannoi logikoi raboty* [Current problems of innovative development: International Scientific and Practical Conference]. Moscow, 2017, pp. 449–463.
2. Bromberg Yerom-David, Reveillere Laurent, Lawall Julia L., Muller G. Automatic Generation of Network Protocol Gateways. In: *10th ACM/IFIP/USENIX International Conference on Middleware Article*, 2009, No. 2, pp. 21–40. Available at: <https://pages.lip6.fr/Julia.Lawall/middleware09.pdf>
3. Madhavapeddy A., Ho A., Deegan T., Scott D., Sohan R. Melange: Creating a “Functional” Internet. *EuroSys 2007: Proceedings of the 2007 conference on EuroSys.*, Lisbon, Portugal, 2007, pp. 101–114. Available at: <https://www.tjd.phlegethon.org/words/eurosys07-melange.pdf>
4. Efftinge S. *Xtext – Language Engineering Made Easy*. Available at: <http://xtext.itemis.com/>.
5. Nyßen Al., Terfloth A. *YAKINDU SCT – Domain-Specific Statecharts. EclipseCon 2012*. Available at: <https://www.itemis.com/en/yakindu/statechart-tools/>.

Для цитирования: Бойцов Г.В., Петров А.В., Шаров В.Г. Автоматическая генерация реализации протоколов взаимодействия устройств во встраиваемых микроконтроллерных системах // Вестник Череповецкого государственного университета. 2018. №4(85). С. 16–15. DOI: 10.23859/1994-0637-2018-4-85-2

For citation: Boitsov G.V., Petrov A.V., Sharov V.G. Automatic implementation generation of interaction protocols for devices in embedded microcontroller systems. *Bulletin of the Cherepovets State University*, 2018, no. 4 (85), pp. 16–22. DOI: 10.23859/1994-0637-2018-4-85-2