

Особенности разработки программного обеспечения для Linux-контроллеров

М.А. Смирнов, В.В. Олоничев, Б.А. Староверов

Костромской государственной технологической университет

amt@kstu.edu.ru

Аннотация. В статье рассмотрены некоторые особенности разработки программного обеспечения для современных встраиваемых систем, работающих под управлением операционной системы Linux. Описанный подход является альтернативой для средств проектирования стандарта МЭК и предназначен для реализации эффективных алгоритмов цифрового управления технологическими установками.

Ключевые слова. Программируемый логический контроллер, операционная система Linux, процессор ARM, кросс-компиляция, toolchain, реализация на языке Си, цифровое адаптивное управление, ssh-протокол, научная библиотека GNU GSL, средства межпроцессного взаимодействия.

1 Введение

На сегодняшний день ведущие производители промышленных компьютеров (JetBox, Atmel, Techbase и др.) и программируемых логических контроллеров (WAGO, ICP DAS, NPE и др.) предоставляют пользователям возможность реализовывать гибкие и эффективные алгоритмы цифрового управления на языках высокого уровня благодаря поддержке многозадачной операционной системы (ОС) Linux [1].

В современном ядре ОС Linux большинство системных вызовов являются вытесняемыми и имеются таймеры высокого разрешения. Вследствие этого Linux начинает занимать области, ранее принадлежащие классическим ОС реального времени, таким как QNX, OS-9 и VxWorks [2–4]. Поддержка спецификации POSIX данными ОС позволяет сравнительно легко переносить программы между ними.

К преимуществам Linux-устройств можно отнести также низкую цену (в своем сегменте средств автоматизации), открытую архитектуру, низкие требования к аппаратным ресурсам [5].

В связи с высокими требованиями, предъявляемыми сегодня к качеству выпускаемой продукции, разработка программного обеспечения (ПО) под описываемые вычислительные системы весьма актуальна [6–8]. В то же время,

как показывает практика работы с Linux-контроллерами отечественного производства (например, ПЛК серии 304/308 фирмы «Овен»), пользователь по причине отсутствия универсальных пошаговых инструкций сталкивается с серьезными не для профессионала проблемами. К ним можно отнести следующие [9]:

- необходимость модификации и конфигурирования ядра Linux;
- необходимость поиска и построения инструментального пакета (toolchain);
- необходимость интеграции дополнительных драйверов, приложений, библиотек;
- необходимость тестирования, отладки и масштабирования проекта.

Указанные причины, по мнению авторов, сдерживают широкое применение Linux-контроллеров и заставляют разработчиков использовать ограниченные возможности языков стандарта МЭК (язык структурированного текста, функциональных блоков, релейных схем и др.).

Изложенный далее материал позволяет ликвидировать проблему масштабирования и быстрого реконфигурирования встраиваемого программного обеспечения. Что касается переноса исполняемых файлов с персонального компьютера (ПК) на целевую платформу, то рассмотрено конкретное решение для процессора ARM AT91RM9200 фирмы ATMEL, на базе которого работает большое количество отечественных и зарубежных контроллеров, в частности ПЛК304/308 «ОВЕН». Наиболее важные рекомендации по общему порядку действий даны в работах [9, 10].

2 Мультипроцессный комплекс управления технологическими установками

Для класса аperiodических объектов с одной доминирующей постоянной времени, подверженных за технологический цикл девиации параметров в широких пределах, что в свою очередь требует перенастройки коэффициентов регулятора, разработана адаптивная многопроцессная управляющая система, функциональная схема которой представлена на рис. 1 [11]. За основу принята классическая САУ с регулятором состояния и наблюдателем [12]. Идентификатор однократно или в темпе с процессом вычисляет параметры технологического объекта управления, которые в свою очередь используются для расчета коэффициентов настройки динамического регулятора.

В состав мультипрограммного продукта входят следующие процессы [13]: «диспетчер», «регулятор состояния», «наблюдатель полного порядка», «адаптатор», «задающее устройство эталонного сигнала», «цифровая модель объекта управления», «связь с реальным объектом», «идентификаторы» («пассивный идентификатор», «идентификатор однократного действия», «идентификатор многократного действия»). Данный комплекс реализован на

языке Си с помощью GNU Scientific Library (GSL) v1.3 – библиотеки для научных расчетов. В качестве средств межпроцессного взаимодействия используется разделяемая память и семафоры System V.

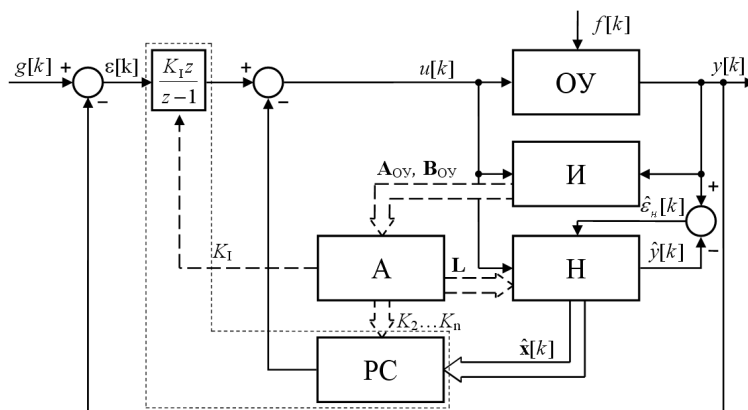


Рис. 1. Цифровая адаптивная самонастраивающаяся система управления: ОУ – объект управления; И – идентификатор; А – адаптатор; Н – наблюдатель; РС – регулятор состояния

Главное преимущество мультипроцессной организации цифровой системы управления заключается в том, что на основе небольших узкоспециализированных программ можно собирать целые комплексы произвольной конфигурации и теоретически неограниченной сложности (именно такой подход отвечает идеологии ОС UNIX [14]). В частности, в зависимости от требований технологического процесса «идентификаторы» реализованы в трех вариантах: «пассивный идентификатор», «идентификатор однократного действия», «идентификатор многократного действия» [11].

Процесс «диспетчер» призван координировать работу всех остальных программ. При запуске он получает два параметра: первый задает режим работы (0 – асинхронный, 1 – синхронный), второй – имя конфигурационного файла.

При асинхронном режиме обмен данными между процессами осуществляется по их готовности и происходит с максимально возможной для данной вычислительной системы скоростью. Данный способ взаимодействия применяется для моделирования поведения системы управления и может быть использован при отладке и тестировании программ на ПК.

При синхронном режиме все вычисления и межпроцессная коммуникация происходят по команде от источника синхросигнала, которым является таймер реального времени. Он генерирует импульсы с заданным пользователем периодом квантования. Описанный способ взаимодействия используется для управления работой технологической установки в режиме реального времени.

Конфигурационный файл содержит следующую информацию: количество регуляторов в системе управления, количество процессов-участников, не

включая «диспетчер», количество semaфоров, а также порядок объекта управления и период квантования.

Структурная схема установки, на которой была развернута мультипроцессная система адаптивного управления представлена на рис. 2. Она включает в себя персональный компьютер, сетевой коммутатор, ПЛК308 фирмы «Овен», модуль аналогового ввода МВА8 фирмы «Овен», модуль дискретного ввода-вывода МДВВ фирмы «Овен», твердотельное реле фирмы «Fotek», электрическую печь (объект управления) и датчик температуры.



Рис. 2. Структурная схема САУ: 1 – кабель Ethernet; 2 – кабель RS-485; 3 – широтно-импульсный сигнал (ШИМ); 4 – сигнал обратной связи

В ходе своей работы программный комплекс считывает по RS-485 с МВА8 сигнал с датчика (датчиков) температуры и выдает в соответствии с заложенным алгоритмом управляющее воздействие в виде процентного изменения мощности на модуль МДВВ, тот в свою очередь формирует на выходе ШИМ сигнал, а твердотельное реле коммутирует нагрузку (электрический нагреватель). Контролируемые параметры могут быть переданы на персональный компьютер или панель оператора. Для этого целесообразно использовать сетевые сокет.

Результаты работы мультипроцессной системы адаптивного управления представлены на рис. 3 на примере решения задачи стабилизации температуры печи при уменьшении напряжения питания с 220 В до 120 В, начиная с 4000 с. Как видно из рис. 3 а, применение традиционного ПИД-регулятора не позволяет оперативно отреагировать на возмущение и приводит к недопустимому снижению температуры, в то время как адаптивная система (рис. 3 б) обеспечивает высокие статические и динамические показатели.

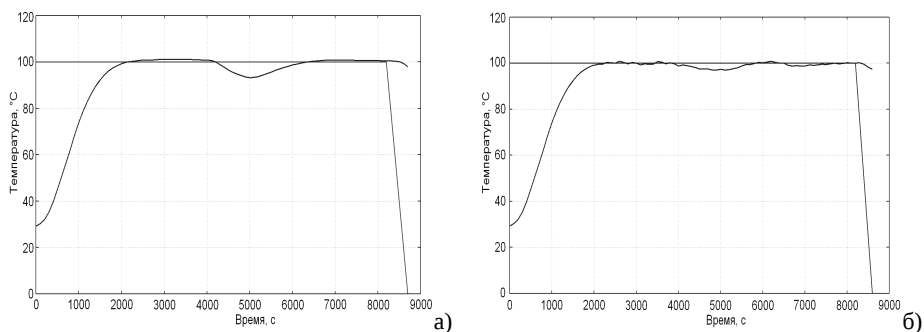


Рис. 3. Исследование работы мультипроцессной системы управления на незагруженной печи при внесении возмущающего воздействия: а) работает неадаптивная система; б) работает адаптивная система

3 Подготовка и запись исполняемых файлов в контроллер

Для компиляции и генерации исполняемого кода для целевой платформы (ПЛК308), работающей под управлением процессора ARM9, на ПК с ОС Linux (uClinux v 2.6), установлен и сконфигурирован набор необходимых программ – toolchain фирмы «Ronetix» – ronetix-arm-linux-uclibc-4.1.2 [15], благодаря которому можно осуществить кросс-компиляцию исходных кодов. Диаграмма развертывания описываемой системы представлена на рис. 4.

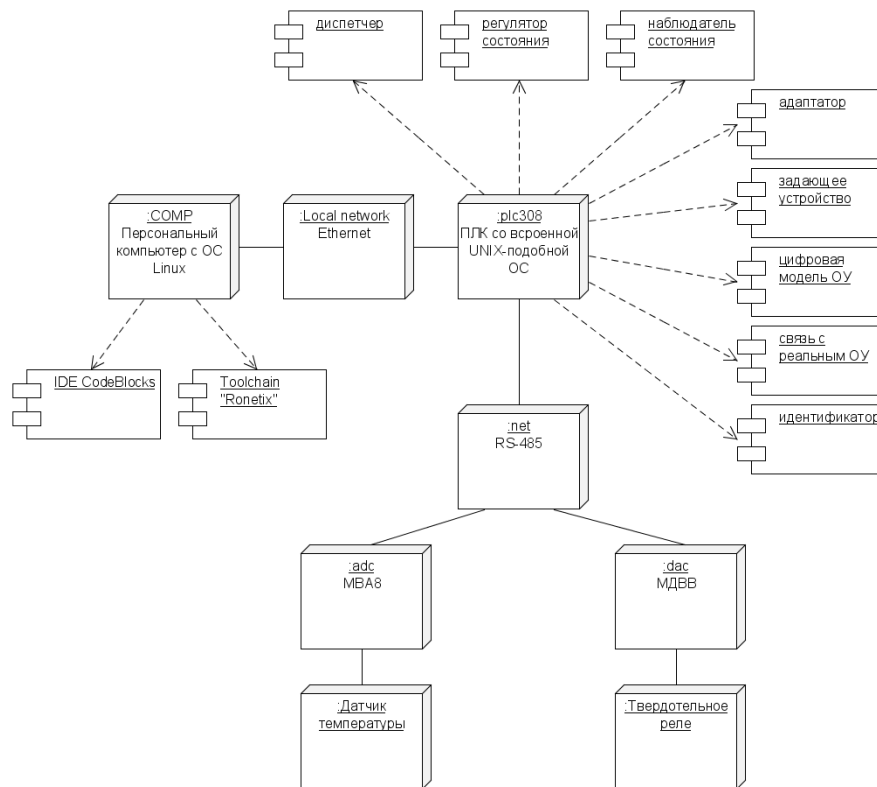


Рис. 4. Диаграмма развертывания мультипрограммной системы управления

При выборе toolchain необходимо знать процессор целевой платформы. Также следует проверить корректное выполнение собранных вместе со специальными библиотеками программ на ОС, установленной на ПЛК.

Для сборки программ можно использовать следующий скрипт, получающий имя исходного файла в качестве параметра командной строки:

```
F=$1
Fo=$F.o
Fc=$F.c
Fa=$F.arm
arm-linux-gcc -I/usr/local/include -o $Fo -c $Fc
arm-linux-gcc -o $Fa $Fo
```

Для сборки программы вместе со специальными библиотеками (в нашем случае использовались статические библиотеки gsl) их следует указать явно:

```
F=$1
Fo=$F.o
Fc=$F.c
Fa=$F.arm
arm-linux-gcc -I/usr/local/include -o $Fo -c $Fc
arm-linux-gcc -o $Fa $Fo ./lib/libgsl.a
./lib/libgslcblas.a
```

Для копирования исполняемых файлов на ПЛК можно воспользоваться различными приемами, а именно:

- Применяем команду `scp`, использующую безопасный зашифрованный протокол `ssh`. Например: `scp ./myprog root@plc308:/home/arm/myprog`. При необходимости можно составить пакетный файл, копирующий за один прием несколько файлов.
- Если важна наглядность, можно использовать консольный файловый менеджер `mc`, настроив одну из его панелей на удаленное подключение по тому же протоколу `ssh: /#sh:root@plc308/home/arm`.

Представленная технология разработки и способ переноса программ прошли испытание на реально работающем оборудовании и подтвердили свою эффективность.

4 Выводы

В статье рассмотрены особенности разработки программного обеспечения для ПЛК-Linux и способ переноса программ на целевую платформу. Описанный подход может быть использован разработчиками программного обеспечения для встраиваемых Unix-систем.

Библиографический список

1. Все для автоматизации. Встраиваемые компьютеры, АСУ ТП, коммуникации, http://empc.ru/e-store/compact_pc_ark/
2. VxWorks, <http://www.windriver.com/products/vxworks/>
3. QNX, <http://www.qnx.com>
4. OS-9, <http://www.radisys.com/products/microware-os-9/>
5. Горбунов Н. О выборе встраиваемой ОС для проекта // СТА. – 2011. – №2. – С. 22-28
6. Ицкович Э.Л. Проблемы развития контроллеров российских производителей // Промышленные АСУ и контроллеры. – 2007. – №2

7. Ицкович Э.Л. Методы совершенствования систем регулирования технологических процессов // Промышленные АСУ и контроллеры. – 2011. – №1. – С. 3-9
8. Колесников А.А. Промышленный компьютер на базе Linux со встроенным GSM-модулем // ИСУП. – 2009. – №1
9. Даммер С. О реальной стоимости «доморощенной» Linux // СТА. – 2010. – №4. – С. 20-26
10. Zimmerly B. Install the GNU ARM toolchain under Linux // EN, developerWorks, May, 2009
11. Староверов Б.А., Олоничев В.В., Смирнов М.А. Реализация законов адаптивного управления технологическими установками на Linux-контроллерах // Промышленные АСУ и контроллеры. – 2012. – №7. – С. 48-53
12. Изерман Р. Цифровые системы управления. М.: Мир, 1984. – 541 с.
13. Олоничев В.В., Смирнов М.А., Староверов Б.А. Мультипроцессный комплекс цифрового управления технологическими установками. Свидетельство о государственной регистрации программы для ЭВМ № 2011611749. © М.: РОСПАТЕНТ, 2011
14. Реймонд Э. Искусство программирования для Unix. – М.: Издательский дом «Вильямс», 2005
15. Toolchain ronetix, <http://www.ronetix.at>