

ПОСТРОЕНИЕ ОКРУЖЕНИЯ РАЗРАБОТКИ ПРИКЛАДНЫХ ПРИЛОЖЕНИЙ АВТОМАТИЗАЦИИ ПОД ОДНОПЛАТНЫЕ КОМПЬЮТЕРЫ

Арыскин А.А.

Научный руководитель: Козак Н.В. – к.т.н., доцент

Кафедра «Компьютерные системы управления» ФГБОУ ВПО МГТУ «СТАНКИН»

Введение

Современные задачи автоматизации требуют повышения гибкости в управлении процессами. В настоящее время одноплатный компьютер является универсальной и производительной платформой, которая может быть использована для реализации любых задач АСУТП [1]. По сути, любой ПЛК также является компьютером. Разница в том, что одноплатный компьютер может не в полном объеме удовлетворять эксплуатационным требованиям, предъявляемым к ПЛК, поэтому использовать его как промышленный контроллер нельзя. Так же одноплатный компьютер требует разработки специализированных приложений.

На данный момент построение окружения разработки приложений под одноплатные компьютеры становится все более актуальным. Это связано с тем, что реализуется много новых различных технологий, для которых необходимы специализированные приложения. Однако ресурс одноплатного компьютера может быть не велик, что отражается на его быстродействии. Во избежание данного явления, используется принцип «исключения ненужного». Данный принцип представляет собой некую методологию формирования операционной системы, основанной на исключении не используемых дистрибутивов или замене их на малогабаритные для повышения производительности работы центрального процессора.

Постановка задачи

В практической задаче для одноплатного компьютера была поставлена задача активного сбора и обработки, данных окружающей среды для навигации робота и распознавания им объектов. Для повышения производительности системы требуется создание (конфигурирование) специализированной версии встроенной операционной системы (ОС) Linux, настроенной и оптимизированной под вышеперечисленные цели. Методически принципы настройки ОС под требования прикладной задачи могут быть использованы для других типов одноплатных компьютеров, в том числе отечественного производства.

Основная часть

В качестве основы выбран бюджетный вариант одноплатного компьютера RaspberryPI, производительность которого позволяет решать задачи касающиеся автоматизации и мониторинга объектов. Основные аппаратные характеристики одноплатного компьютера RaspberryPi модель А: процессор - Broadcom 2708 700МГц; оперативная память (SDRAM) -

256Мб; 26 пинов GPIO (низкоуровневый интерфейс ввода-вывода прямого управления). Интерфейсы данного компьютера показаны на рис. 1.

Применение миникомпьютера дает ряд преимуществ: порты GPIO для взаимодействия с периферией; низкое энергопотребление; удобные средства и решения для отладки и диагностики. К недостаткам можно отнести: большой размер готовых дистрибутивов ввиду их универсальности, отсутствие встроенных часов реального времени; GPIO обрабатывают только цифровые сигналы; Выходы не могут одновременно выводить картинку.

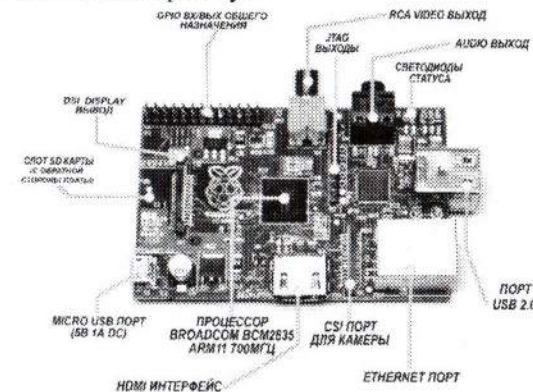


Рис. 1. Интерфейсы платы RaspberryPi. Реализация

Конфигурирование встроенных систем реализуют системы автоматических сборок, обеспечивающие разрешение зависимостей между устанавливаемыми пакетами, установку средств отладки и компиляции, контроль и обнаружение ошибок и т.д. Для данного проекта были рассмотрены несколько основных сред для автоматической сборки операционных систем (Таблица 1):

Таблица 1 Системы автоматической сборки ОС

Среда сборки ОС	Сложность сборки 0-5	Неизбежные ошибки (штук)	Кол-во этапов сборки	Время сборки (мин)	Размер образа ОС
OpenBricks	4	3	1	30-60	100-150мб
LTIB	2	2	1	20-40	120мб
PTXdist	2	2	2	30-60	120мб
OpenEmbedded	3	0	1	40-60	150мб
BuildRoot	1	0	1	20-40	50-100мб

OpenBricks, LTIB, PTXdist являются узконаправленными системами сборки специфичных проектов и для нашей цели не подошли. OpenEmbedded и BuildRoot это очень мощные и гибко-настраиваемые системы. В результате исследования была выбрана система сборки BuildRoot, причиной послужила простота интерфейса и управления (аналогично интерфейсу конфигурации ядра Linux) [2].

В данной системе есть выбор целевой архитектуры, процессора, особенности работы ядра Linux, драйвера, программы и т.д. то, что необходимо включить в будущую встроенную систему. BuildRoot включает компоненты для загрузки пакетов и исходных кодов через интернет (autotools, make, git, ncurses и wgetscp). На рис.2. показана структура окружения разработки приложений под RaspberryPi.

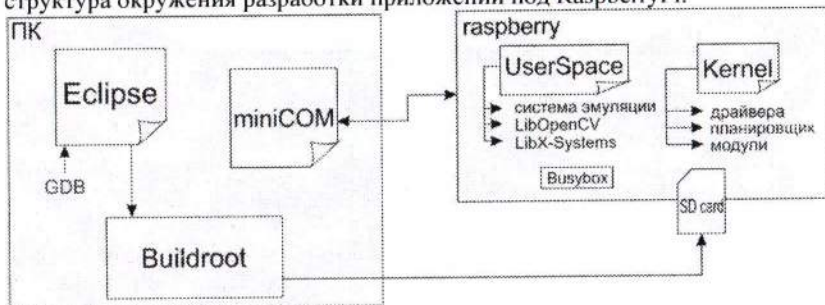


Рис. 2. Структура окружения разработки приложений под RaspberryPi

Для персонального компьютера (ПК) в окружение разработки входит: Eclipse – кроссплатформенная свободная интегрированная среда разработки; отладчик GDB -свободно распространяемый организацией GNU; miniCOM – терминал, для удаленной работы через последовательные порты; LibOpenCV и LibX-Systems библиотеки компьютерного зрения и передачи данных через Ethernet соответственно; BusyBox— набор утилит командной строки.

Сконфигурированное ПО для RaspberryPi состоит из следующих компонентов: ядро ОС Linux; система инициализации BusyBox; программы отладки ядра trace-cmd , kgdb; слежение за процессами htop; текстовые редакторы vim, nano; dhcpcd – dhcp клиент; библиотека компьютерного зрения opencv; библиотека работы с микрофоном sox; утилиты работы с сетью nettools.

Построение окружения разработки приложений

Для создания окружения все пакеты указываются в BuildRoot в меню TargetPackages. За набор компонентов на Raspberry отвечает сценарий в BuildRoot, поэтому в целевой системе компиляторы и библиотеки для сборки могут отсутствовать.

Для разработки собственного приложения, возможны два варианта действий:

1. Сборка целевой системы на Raspberry с программами компиляции, сборки и отладки, для разработки ПО внутри системы.

2. Разработка и сбора компонентов на ПК (кросс-компиляция), и добавление компонентов в пакеты целевой системы на Raspberry.

Заключение

Выше описанный метод позволяет формировать целевое программное обеспечение и создавать окружение разработки приложений для реализации прикладных задач на базе одноплатных компьютеров [3].

Результаты исследования вошли в проект «Интеллектуальная робототроника» Международной лаборатории «Сенсорика», ИПМ им. М.В. Келдыша РАН.

Библиографический список:

1. Солдатов А.В. Российские промышленные одноплатные компьютеры и их применение в системах автоматизации // ИСУП. - 2009. № 2 (22);
2. Обзор систем сборки для RaspberryPi [Электронный ресурс] // habrahabr.ru: IT-сообщество. URL: <http://habrahabr.ru/post/172349/> (Дата обращения: 10.03.2015)
3. Sergej N. Grigoriev, Georgi M. Martinov Scalable open cross-platform kernel of PCN system for multi-axis machine tool // Procedia CIRP 1 (2012) p.p. 238 – 243.

СОЗДАНИЕ ПРОГРАММНО-АППАРАТНОЙ ПЛАТФОРМЫ ДЛЯ ТЕСТИРОВАНИЯ ЖИДКОКРИСТАЛЛИЧЕСКИХ ИНДИКАТОРОВ ВЫВОДА ИНФОРМАЦИИ СОСТОЯНИЯ РАБОТЫ ЯДРА СИСТЕМЫ ЧПУ

Бабин М.С.

Научный руководитель: преподаватель Ковалев И.А.

Кафедра «Компьютерные системы управления» ФГБОУ ВПО МГТУ «СТАНКИН»

Современные персональные компьютеры позволяют проводить систематизированный сбор сведений о подконтрольном объекте и средствах воздействия на его поведение с целью достижения определенных целей, таким образом, решая все задачи управления (геометрическую, логическую, терминальную) чисто программным путем, без какой-либо дополнительной аппаратной поддержки [1,2].

При использовании систем реального времени для выполнения функций ядра системы управления обычно используется безграфические операционные системы. Вся информация с таких систем поступает на терминал оператора (терминал управления). Но из этого следует, что связь между ядром системы управления и терминалом будет установлена только

¹ Работа выполнена по договору № 2332 ГУ/2014от 19.06.2014 об условиях использования гранта на выполнение научно-исследовательских работ