

Н. И. Кольский, «МКА: ВКС»

Язык программирования встроенных систем: свобода выбора или жёсткий детерминизм?

Кроха сын к отцу пришёл, и спросила кроха: «Си сегодня хорошо, ну а Java...?»

Фольклор

Анализ новостных лент и анонсы компаний, работающих на рынке встроенных систем, позволяют предположить, что сегодня фокус внимания игроков этого рынка смещается в сторону программного обеспечения и средств его разработки. В пользу этого говорят не только активность софтверных компаний, но и самих вендоров процессоров и компьютерных модулей в части всеобъемлющей поддержки их платформ операционными системами, средствами написания кода и его отладки и верификации. Один из злободневных вопросов программирования приложений для встроенных систем – выбор оптимального набора языков программирования.

Один из «индексов популярности» языков программирования рассчитывается на основании их рейтингов в поисковых машинах Интернета. Январские данные 2010 года представлены в табл. 1 (для простоты оставлена лишь первая десятка, «подмявшая» львиную долю «рынка»).

Как видно из этой таблицы, более половины популярности приходится на 4 языка, два из которых

часто упоминаются в аналитических обзорах и статьях как два Аякса – C/C++. В целом же картина, представляемая данными из табл. 1, отражает сильную диверсификацию рынка встроенных приложений. Но есть в ней и отзвуки бурсацкой вольности программирования, той области техники, что лишь относительно недавно получила в английском языке добавку engineering к термину software. Ещё

10 лет назад в патриархе отечественных компьютерных СМИ об этой вольнице было написано: «...Программисты, пишущие софт для встроенных систем, по определению и по натуре являются экстремалами. Много степеней свободы (то структура «железа» поменялась, то задачи расширились, причём всё меняется на ходу) делают процесс проектирования системы похожим на опасную гонку на велосипедах/мотоциклах/джипах (соответственно для 4-, 8-, 16-, 32-битовых платформ) по пере-

Табл. 1. Рейтинг популярности языков программирования на основании их «индекса цитируемости» в поисковых машинах Интернета

Позиция в рейтинге на январь 2010 г.	Позиция в рейтинге на январь 2009 г.	Язык	Рейтинг января 2010 г.
1	1	Java	17,482%
2	2	C	16,215%
3	5	PHP	10,071%
4	3	C++	9,709%
5	4	(Visual) Basic	7,354%
6	6	C#	5,767%
7	7	Python	4,453%
8	8	Perl	3,562%
9	9	JavaScript	2,707%
10	11	Ruby	2,474%

сечённой местности с решением ряда головоломок, возникающих в реальном времени. Инженеры, разумеется, используют ООП и структурное программирование и ещё, например, модели Royce Waterfall, Boehm's Spiral, Rapid Iterative (Gilb) и самый последний писк – методологию Хайнса-Ортеги». Это цитата из статьи Андрея Фаткуллина в журнале «Компьютерра» №27 2001 года.

О последствиях «экстремизма» молодых программистов, потоком поступающих на стремительно растущий рынок встроенных систем, писал позже в своей статье «Некоторые мысли по поводу программирования встроенных систем» («Компоненты и технологии», № 8, 2006) Владимир Татарчевский: «...каждый программист работает фактически на тех инструментальных средствах, на которых хочет работать. Один программирует на Delphi 7, другой никак не может «слезть» с Delphi 5, третий выбрал C++ Builder и пишет на нём... Программисты в фирме не взаимозаменяемы. Мало того, что каждый программирует, на чём хочет, они ещё ведут проекты в одиночку, и все нюансы проекта находятся у разработчика в голове... Выводы просты. В пределах одной фирмы следует сократить ассортимент используемых инструментальных средств до абсолютно необходимого минимума. Должно использоваться всего два пакета: один для компьютерного программирования, второй для программирования микроконтроллеров».

Хотя в некоторых из этих высказываний желающие могут найти «перегибы», в целом проблемы программирования встроенных систем и возможные пути их решения сформулирова-

ны, на мой взгляд, правильно. В этой статье хочется акцентировать внимание на проблеме однородности «языковой среды» разработчиков приложений для встроенных систем и качестве «языковой подготовки».

Реальную расстановку сил в области языков программирования для встроенных систем достаточно аргументированно описал Майкл Барр (Michael Barr), автор трёх книг и более 50 статей по теме разработок встроенных систем, бывший некогда главным редактором журнала *Embedded Systems Design Magazine*. В своей статье «Real men program in C» Майкл постарался на основе объективных статистических данных выразить своё отношение к ситуации, сложившейся в программировании встроенных систем как у нас принято иногда говорить «на западе». В качестве вариантов перевода названия

упомянутой статьи можно предложить «Язык C: программирование для настоящих мужчин» или «Конкретные мужики программируют на C».

Толчком для написания статьи Майкла послужила его встреча с двумя выпускниками Университета Джона Хопкинса, специализирующимися в области информационных технологий и работающими в компании, которая в том числе разрабатывает и программное обеспечение для Web-ориентированных баз данных. Инструментарием для этой работы в компании был выбран Ruby On Rails (формально и кратко о Ruby On Rails см. врезку «Ruby on Rails»).

В беседе с этими молодыми людьми Майкл Барр услышал заинтересовавший его отзыв о языке C: «...real men program in C», что, как уже говорилось выше, переводится как «програм-

Ruby on Rails

Ruby on Rails – программное обеспечение класса «framework» для создания Web-приложений, написанное на языке программирования Ruby.

Ruby – объектно-ориентированный язык высокого уровня, отличающийся удобством и быстротой программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, поддерживает «сборку мусора» и ряд других возможностей.

Ruby on Rails предоставляет архитектурный образец Model-View-Controller (модель-представление-контроллер) для Web-приложений, обеспечивает их интеграцию с Web-сервером и сервером базы данных, позволяет разрабатывать динамические AJAX-интерфейсы, с обработкой запросов и выдачи данных в контроллерах, отражением предметной области в базе данных.

AJAX (Asynchronous Javascript and XML/асинхронный JavaScript и XML) – подход к построению интерактивных пользовательских интерфейсов Web-приложений, заключающийся в «фоновом» обмене данными браузера с Web-сервером. В результате при обновлении данных Web-страница не перезагружается полностью и Web-приложения становятся более быстрыми и удобными.

Ruby on Rails является открытым программным обеспечением и распространяется под лицензией MIT (допускается использование кода в проприетарном программном обеспечении).

мирование на С – занятие для настоящих мужчин». Майкл порасспросил своих собеседников и понял, что выражение это жаргонное и характеризует определённые трудности в изучении и применении языка С поколением, к которому относились его молодые собеседники, а также связанный с этим «возрастной ценз» программистов на языке С. Майкл понял, что для молодёжи язык С трудноват, а многие студенты «компьютерных» специальностей считают, что выбор языка С в качестве объекта изучения сродни выбору латинского языка в качестве специализации для студентов языковых специальностей.

Но, по мнению Майкла, язык С – это не только история, но и главный инструмент для разработки приложений для встроенных систем, области программирования, которая стремительно развивается сегодня. Об этом свидетельствуют, в частности, приведённые в статье Майкла графики (рис. 1). Эти кривые – результат ежегодного анализа «языковой» ситуации издателями журнала *Embedded Systems Design*.

Разрыв в приведённых кривых вызван сменой формулировки вопроса и возможных ответов на него в 2005 году. До 2005 года разработчиков спрашивали: «Какие из перечисленных языков вы использовали в своих проектах для встроенных систем последние 12 месяцев?» В 2005 году разработчики должны были закончить фразу: «Программирование в моём текущем проекте ведётся в основном на...». До 2005 года допускался многозначный ответ. При этом оказывалось, что многие респонденты используют два и более языков в своих проектах. Наибольшее влияние на изменения в кривых с переходом на односложный ответ выявилось в отношении ассемблера. До 2005 года средний процент опрошенных, называвших этот язык, достигал 62% в связи с тем, что, как хорошо известно, практически любой проект по разработке программного обеспечения

для встроенных систем требует хотя бы небольшого объёма работ, проводимых на ассемблере.

После 2004 года ассемблер превратился в аутсайдера – в среднем лишь 7% респондентов по результатам опросов пяти лет. Эта цифра достаточно точно отражает процентное содержание проектов на языке ассемблер в общем количестве проектов по написанию приложений для встроенных систем. Данные также свидетельствуют о стабильном снижении популярности этого языка с 8% до 5% в период 2005–2009 гг.

Возвращаясь к анализу ситуации с языком С, Майкл отметил, что данные за 13 лет свидетельствуют о доминирующей роли этого языка. Язык С был самым популярным в 1999 году, в 2009 году и на всём временном отрезке между ними. Язык С доминирует в опросах, допускающих многозначный ответ (в среднем 81%), и в опросах, когда ответ мог быть только односложным (в среднем 57%).

По мнению Майкла Барра, в последние 5 лет язык С отнимал долю рынка и у ассемблера, и у С++. Совершенно очевидно, что С++ – один из языков программирования для ряда проектов и основной язык для 27% проектов последних 5 лет. По личным оценкам Майкла, в последние 13 лет использование С++ стремительно нарастало с конца 90-х годов прошлого века, достигло пика в 2001 году и ныне стабилизировалось или же, возможно, немного снижается.

Подводя итоги анализа графиков на рис. 1, Майкл Барр выразил уверенность в том, что использование языка С программистами встроенных систем вряд ли прекратится в обозримом будущем. Для такого вывода есть ряд оснований. Во-первых, компиляторы языка С имеются для широко спектра 8-, 16- и 32-разрядных процессоров. Во-вторых, для этого языка характерно оптимальное соотношение высоко- и низкоуровневых возможностей программирования процессоров и написания драйверов.

Для того чтобы популярность языка С сохранялась с учётом всех его достоинств в длительной перспективе, должен расти рынок встраиваемых систем и расширяться сообщество программистов, работающих на этом рынке и знающих этот язык.

В своей статье «Real men program in C» Майкл Барр высказал в общем-то не слишком

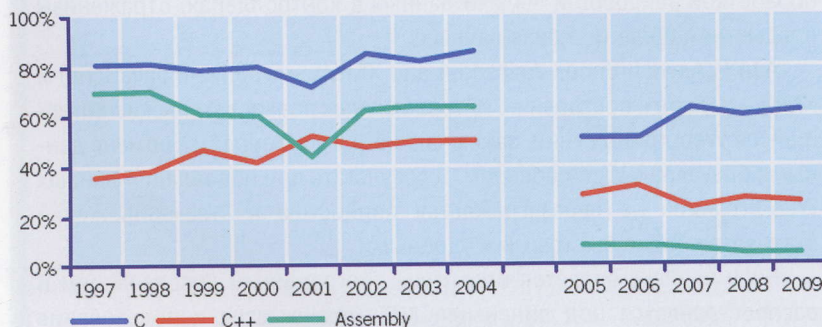


Рис. 1. Популярность языков программирования для разработки приложений для встроенных систем

оригинальное мнение о том, что рынок встраиваемых систем, безусловно, растёт. Около 98% процессоров, выпускаемых ежегодно, ориентировано на использование во встраиваемых системах. Количество новых процессоров, появляющихся каждый год, растёт. В своей статье Майкл привёл рис. 2, на котором представлено количество производимых ежегодно процессоров (оценка Майкла Барра) в сопоставлении с ростом индекса Nasdaq (точные биржевые данные).

Как следует из графиков на рис. 2, количество производимых процессоров удваивается каждые 10 лет, несмотря на то, что биржевой индикатор за тот же период демонстрирует снижение. В целом же графики на рис. 2 демонстрируют факт стабильного спроса на программирование встроенных систем и на услуги специалистов, способных писать такие программы, тем более что по мере удешевления процессоров появляется всё больше приложений и задач для встроенных систем. Уместно заметить, что абсолютное количество производимых 8-разрядных процессоров по-прежнему велико, а для написания приложений для проектов под такие аппаратные платфор-

мы привлекаются, как правило, небольшие команды программистов из одного-двух человек.

Далее в своей статье «Real men program in C» Майкл Барр с горечью отмечает, что одновременно с объективным ростом спроса на программирование встроенных систем на языке C и важности этого языка для индустрии встроенных систем, всё меньшее внимание уделяется изучению этого языка. Эта проблема – часть более широкой проблемы пробелов в образовательном процессе, с которым сталкиваются компании, работающие на рынке встроенных систем. По мнению Майкла, система высшего образования в США не обеспечивает притока студентам должных практических навыков в области разработки надёжного программного обеспечения. Поэтому технологиям и приёмам программирования ответственных приложений реального времени, умению выбрать правильную архитектуру программного обеспечения для встроенной системы выпускникам приходится учиться уже на работе.

Проблема пробелов в образовательном процессе может быть проиллюстрирована рис. 3.

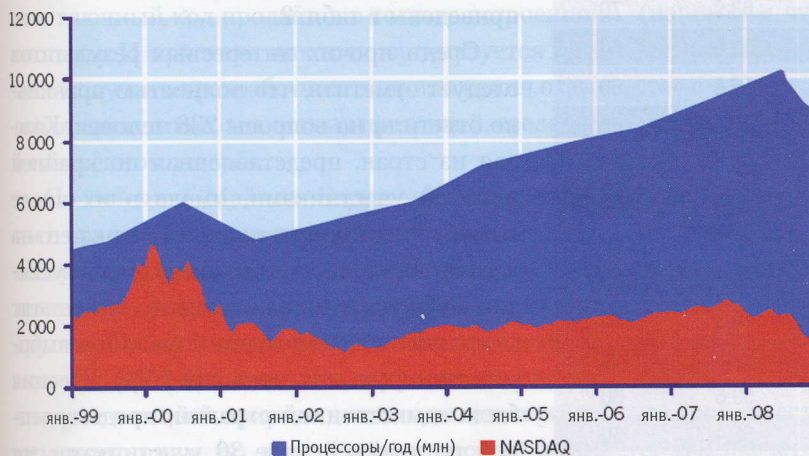


Рис. 2. Количество производимых ежегодно процессоров (оценка Майкла Барра) в сопоставлении с ростом индекса Nasdaq (точные биржевые данные)

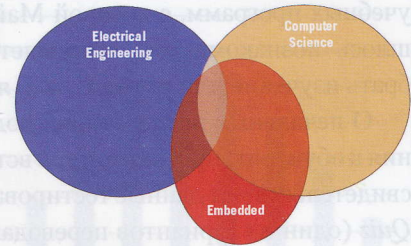


Рис. 3. Доля знаний о программировании встроенных систем в общем их объёме, получаемом студентами США, специализирующимся в электротехнике (electrical engineering) и в области компьютерных наук (computer science) по оценкам Майкла Барра

По мнению Майкла Барра, только малая толика знаний, получаемых студентами, специализирующимися в электротехнике (electrical engineering), применима к разработке встроенных систем: ближе к концу цикла лабораторных работ в небольшом объёме изучается программирование на ассемблере. Не намного больше знаний, необходимых для программиста встроенных систем, получают студенты, специализирующиеся в области компьютерных наук (computer science). Это сведения о компьютерных архитектурах на первых ступенях обучения и курсы по выбору в области программирования на C и C++, а также представления об управлении жизненным циклом программного обеспечения.

Правда, в последнее время делаются попытки устранить бреши в знаниях студентов компьютерных специальностей (computer engineering), однако, по мнению Майкла, эти усилия запоздали и недостаточны. По сути, это некая эклектика на основе имеющихся курсов для студентов, специализирующихся в областях электротехники и компьютерных наук, что позволяет лишь в очень малой степени дать им знания в области программирования встроенных систем. Так, одна из

учебных программ, с которой Майклу Барру пришлось познакомиться, позволяет студентам выбрать изучение или языка C, или языка Java.

О печальных последствиях подобного отношения к обучению программистов встроенных систем свидетельствуют данные тестирования *Embedded C Quiz* (один из вариантов перевода: «Викторина по Embedded C») программистов, разрабатывающих приложения на языке C для встроенных систем. Эта «викторина» была проведена на сайте проекта Netrino (<http://www.netrino.com/>). За период, чуть больший двух лет, с момента обновления сайта в декабре 2007 года, в тестировании приняло участие более 4 тыс. человек. Результаты и их анализ опубликованы в статье Майкла Барра «Embedded systems programmers worldwide earn failing grades in C. Why are embedded systems software developers getting a D- in C?», опубликованной в 2010 году на сайте Embedded.com. Название статьи можно перевести как «Уровень подготовки программистов встроенных систем на языке C во всём мире падает. Почему они получили двойку?»

В тестовом задании предлагалось 10 вопросов с четырьмя вариантами ответов, среди которых мог встречаться и вариант «Ни один из перечисленных». Все вопросы были независимы и следовали в одинаковом для всех участников порядке.

Табл. 2. Результаты тестирования Embedded C Quiz, проводимого на сайте проекта Netrino (<http://www.netrino.com/>)

Страна / Регион	Число участников	Средняя оценка	Оценка медианы
«Весь мир»	4073	61,0	60
США	1465	62,0	60
Индия	1361	59,3	60
«Остальной мир»	1247	61,7	60
Западная Европа	446	63,9	70
Азия (кроме Индии)	304	58,0	60
Ближний Восток	101	56,6	60
Канада	124	64,6	60
Объединённое Королевство	117	65,4	70
Турция	67	55,4	50
Германия	62	66,5	70
Бразилия	56	56,6	60
Китай	53	64,5	70
Австралия	51	67,8	70
Румыния	51	59,6	60
Италия	44	56,4	50
Франция	39	64,6	70
Швеция	30	67,0	70

Большая часть вопросов относилась к темам и подходам, описанным в имеющихся в свободном доступе на сайте проекта Netrino технических статьях и материалах. Все участники опроса могли после его окончания получить доступ к оценке результатов тестирования, ознакомиться с правильным ответом на каждый вопрос. Результаты тестирования были сохранены в базе данных, которая обеспечивала для каждого участника тестирования доступ только к его информации. Специализированный характер ресурса и отсутствие уж очень «дорогостоящих» стимулов для участия в тестировании дают основания полагать, что подавляющее большинство его участников относятся к сообществу разработчиков приложений для встроенных систем. Для окончательного анализа были отобраны полностью завершённые тесты (там, где ответ был дан на все 10 вопросов).

С точки зрения географического размещения респондентов по результатам опроса удалось выделить три региона с числом участников более 1 тыс. человек:

– США: 1465;

– Индия: 1361;

– «Остальной мир»: 1247.

Регион «Остальной мир» в свою очередь включал три крупных субрегиона:

– Западная Европа: 446;

– Азия (кроме Индии): 304;

– Ближний Восток: 101.

Средний результат по всем данным тестирования – 61,0%, то есть «средний» программист ответил неправильно на 4 из 10 вопросов из области программирования на языке C (уместно напомнить, что язык C++ сложнее языка C). Полные данные приведены в табл. 2.

Среди прочих интересных результатов следует отметить, что полностью правильно ответили на вопросы 238 человек. Каждая из стран, представленная по крайней мере 30 участниками, имела в их числе хотя бы одного, правильно ответившего на все вопросы. На другом конце шкалы успехов 209 участников, имевших результат меньший ожидаемого при случайном выборе ответов (то есть меньший 25%). Швеция была единственной страной, представленной по крайней мере 30 участниками, ни один из которых не продемонстрировал результат такого низкого уровня.

На рис. 4 представлена диаграмма результатов тестирования *Embedded C Quiz*, из которой следует, что треть участников набрали оценку «свыше 80%», а четверть — «менее 50%».

В ближайшее время Майкл Барр планирует проанализировать полученные данные с точки зрения возможности выявления наиболее трудных вопросов из области программирования на языке C или же выявления некорректно поставленных в рамках тестирования вопросов. В число планов Майкла входит также составление более широкого опросника — как для проверки знаний в области программирования на языке C, так и в смежных областях программирования систем реального времени и ответственных приложений.

В сентябре 2009 года на сайте был запущен новый опрос *Embedded C++ Quiz*. Предварительные результаты ещё хуже, чем те, что были показаны участниками тестирования *Embedded C Quiz*. Больше число участников не доходит до финиша тестирования. Желающие могут принять участие в новом опросе по адресу в Интернете <http://www.netrino.com/Embedded-Systems/Embedded-C++-Quiz>.

Майкл Барр считает, что изучение программирования встроенных систем плохо организовано и на рабочем месте, куда приходит бывший выпускник. Считается допустимым, если не превращается в обычную практику, доверять написание приложений для встроенных систем «специалистам», только что выпущенным из высшего заведения по специальности электротехника. Такие «программисты» начинают плодить ошибки, характерные для новичков, но при этом результаты их труда практически не анализируются на предмет качества кода; созданный неопитом программирования глючный код продаётся в составе встроенной системы. И при этом никак не организуется процесс получения разработчиком приложения отзывов от заказчиков, анализа их претензий, сведений о нанесённом ущербе.

По мнению Майкла Барра, представленные им в статье «Real men program in C» данные дают основания предполагать о существовании двух тенденций, вступающих в противоречие друг с другом. С одной стороны увеличивается потребность в приложениях для встроенных систем, которые по своей природе должны отвечать жёстким критериям по части правильности функционирования и надёжности. С другой стороны, качество образования программистов для встро-

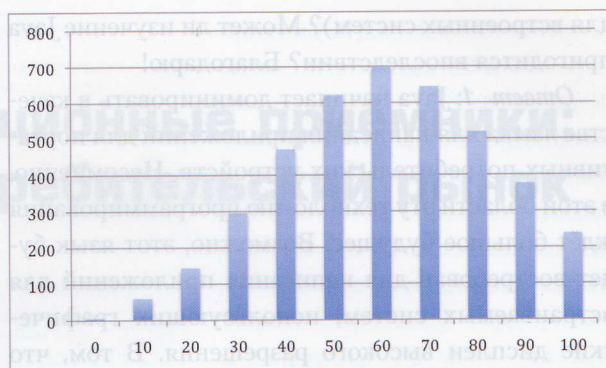


Рис. 4. Диаграмма результатов тестирования *Embedded C Quiz*

енных систем оставляет желать лучшего. Одна из причин этой проблемы — низкий уровень преподавания и соответственно знаний выпускников высшей школы языка C. И, что самое неприятное, студенты и будущие выпускники и не стремятся к изучению C. Такое положение дел, по мнению Майкла Барра, недопустимо. Да, язык C достаточно сложен, но нельзя ждать, что появятся более простые для изучения и использования языки программирования встроенных систем. Работать надо уже сейчас.

В связи с анализом популярности языка следует отметить, что хотя в табл. 1 за ним числится 17% «популярности», целый ряд экспертов полагает, что на практике около 80% проектов в области встроенных систем используют язык C. В связи с этим следует отметить, что ветераны embedded-программирования разделяют встроенные системы на «настоящие», и на те, которые рождены Интернет-бумом и бумом потребительской мультимедийной электроники. И именно с этим связана неоднозначная оценка популярности Java в практике embedded-программирования (напомним, что в табл. 1 Java первенствует). Здесь уместно будет привести вопрос одного из студентов в Интернет-форуме и пару характерных ответов на него, наиболее развёрнуто отражающих общее настроение отвечающих:

«Вопрос: Привет, я студент, специализирующийся в области компьютерных наук, и мне хотелось бы работать после обучения в области программирования встроенных систем. Я изучаю языки C и Java. Хотелось бы услышать комментарии профессионалов по такому вопросу: насколько мне известно, язык C широко распространён. А как насчёт Java ME (я имею в виду Java для встроенных систем)? Вы используете этот язык (или, может быть, слышали о проектах на Java

для встроенных систем)? Может ли изучение Java пригодится впоследствии? Благодарю!

Ответ_1: Java начинает доминировать в качестве языка для написания приложений для портативных потребительских устройств. Несомненно, в этой области эту технологию программирования ждёт большое будущее. Возможно, этот язык будет востребован для написания приложений для встраиваемых систем, использующих графические дисплеи высокого разрешения. В том, что касается других приложений для встроенных систем, скорее всего, у языка Java нет будущего. Если кто-то попытается высказать иное мнение, то ему придётся использовать аргументы 10-летней давности, когда все были убеждены, что Java займёт доминирующие позиции по всему «фронту» приложений. Поскольку за 10 лет ничего такого не произошло, то, наверное, уже и не произойдёт. Полагаю, что C/C++ и ассемблер продолжают занимать лидирующие позиции для приложений [в области встроенных систем].

Ответ_2: О моей квалификации: я активно работаю в области встроенных систем уже на протяжении 30 лет (начал с систем 8080 в 1978 году). Язык Java не распространён широко в разработках приложений для встроенных систем (даже, несмотря на то, что изначально этот язык разработан как раз для встроенных систем, эта попытка потерпела фиаско). Если говорить обо мне, я никогда не слышал о настоящих проектах в области встроенных систем, в основу которых был положен Java (разработки на Java приложений для портативных устройств НЕ МОГУТ рассматриваться как настоящая разработка для встроенной системы). Не думаю, что у Java есть будущее в области программирования встроенных систем, потому что Java слишком абстрагирован от аппаратного обеспечения. Разработка программного обеспечения для встроенной системы должна опираться на обеспечение для разработчика полного, прямого и неограниченного доступа ко всем возможностям аппаратной платформы. Язык C (но НЕ C++) является и был с начала 80-х годов «lingua franca» (общепринятый язык) или «de facto» стандартным языком программирования для встроенных систем. Все лучшие и наиболее широко распространённые средства разработки приложений для встроенных систем поддерживают язык C. Для встроенных систем разработано огромное количество стандартных библиотек и процедур на языке C, на порядок больше, чем для

других языков, используемых в области встроенных систем. Код для встроенных систем, предлагаемый в виде продуктов Open Source или в качестве примеров, почти всегда написан на C. Если ты не являешься экспертом в «чистой» ветке C (стандарт ANSI или в изложении K&R), ты не сделаешь заметной карьеры в области разработок встроенных систем. Рад помочь, если необходимо, не стесняйся и обращай ко мне».

В связи с этими высказываниями 2010 года хочется напомнить анализ Java в статье «Java здесь, Java там...» Мартина Тиммермана (Martin Timmerman) и Лорана Урса (Laurent Uhres), опубликованной в журнале «МКА» №3 за 1999 год: «...Синтаксис Java разработан на основе синтаксиса языков C/C++, что весьма облегчает его изучение. Вместе с тем он сохранил многие недостатки этих языков. ... Достоинство Java, в отличие от C/C++, – запрет манипулирования указателями. Благодаря этому повышается надёжность программ, поскольку в них наверняка не будет ошибочных действий с указателями. ...»

Вместе с тем, с точки зрения разработчиков встраиваемых систем, которым при создании драйверов и обработчиков прерываний требуется непосредственный доступ в память и к регистрам устройств, невозможность манипулирования указателями является недостатком. Разработчику приходится писать отдельные коды на C/C++ или ассемблере, которые должны вызываться из Java-программы или исполняться как независимые нити. Естественно, это не приближает Java к языку классического программирования, поскольку повышает сложность конечного продукта...

С нашей точки зрения, Java не совсем удовлетворяет требованиям, предъявляемым к языку программирования встроенных систем. Язык Java не улучшает стиль программирования встроенных систем и приложений реального времени, не обеспечивает существенного упрощения программ по сравнению со стандартными языками. Библиотека поддержки нитей не рассчитана на жёсткое реальное время. Новых решений язык Java не даёт... Подводя итог сказанному, мы считаем, что Java в нынешнем своем состоянии не способен показать сколько-нибудь значительные преимущества, способствующие его широкому применению во встроенных системах. В целом, плюсы Java недостаточны для того, чтобы уравновесить связанные с его применением трудности».