

Как сократить время разработки изделий на процессоре с ядром ARM: процессорные модули Qseven

Алексей НЕКРАСОВ
alexey.nekrasov@eltech.spb.ru

Как известно, разработка изделия с применением нового процессора на ядре ARM требует определенного времени. Ведь несмотря на то, что архитектура самого процессора с ядром ARM известна (так как лицензируется фирмой ARM), у разработчика остаются проблемы с подключением к процессору различной периферии и отдельных устройств. И для решения этих проблем необходимо определенное время на изучение вопроса. Затем разработчику нужно получить рабочий образец нового изделия, после чего часть времени потратить на его дальнейшую отладку. Это стандартный процесс разработки изделий на процессорах с ядром ARM, и в среднем он может занимать около года. Причем на решение вопросов аппаратного обеспечения приходится лишь 40–30% времени, а остальные 60–70% уйдут на разработку нового либо модернизацию уже существующего программного обеспечения для нового процессора. В результате время, необходимое для быстрого запуска в производство нового изделия, будет не очень эффективно потрачено на разработку. И один из главных вопросов, который стоит перед началом каждой новой разработки того или иного изделия с применением процессора с архитектурой ARM, — как сократить время разработки.

Эта статья была написана для того, чтобы показать, каким образом можно значительно сократить время разработки изделий, базирующихся на процессорах с ядром ARM (причем независимо от семейства), используя принципы и стандарты, применяемые в архитектуре x86 уже много лет. И речь в ней пойдет об открытом стандарте Qseven.

Казалось бы, какое отношение архитектура x86 имеет к разработке изделий с применением ARM? Действительно, в начале своего пути архитектуры x86 и RISC, а в дальнейшем и ARM, преследовали абсолютно разные задачи и были направлены на различные области применения. Архитектура x86 использовалась там, где требовалась высокая производительность и предполагалось наличие сложных математических задач с большим объемом вычислений, а RISC (а затем и ARM) предназначена для снижения энергопотребления. Однако в данный момент четкие гра-

ницы между этими двумя архитектурами постепенно размываются, особенно хорошо это заметно в области мобильных и переносных приборов и устройств.

Конечно, все зависит от конкретной задачи. Но очевидно, что с задачами, которые связаны с кодированием и декодированием видеоряда данных, отображением графического пользовательского интерфейса и т. п., процессоры на ядрах ARM старших семейств (Cortex A8, Cortex A9) справляются не хуже, чем процессоры младших моделей архитектуры x86 (Intel Atom, AMD Fusion и др.). А по соотношению энергопотребление/производительность процессоры на ядре ARM значительно выигрывают. К тому же если сравнивать стоимость изделий на ARM и на x86, то ARM выглядит весьма привлекательно в качестве недорогой платформы для новых разработок.

Однако очень часто для решения описанных выше задач за основу берутся именно изделия на x86. Почему?

Ответ прост: разработка любого нового изделия всегда начинается с поиска. Поиска оптимального пути по созданию требуемого изделия, соответствующего всем заданным в ТЗ характеристикам. Причем очень часто оптимальный путь может оцениваться по времени, затраченному на разработку изделия. Конечно, так же важны и другие ресурсы (к примеру, стоимость разработки), но очень часто именно время является ключевым показателем.

Поэтому при разработке нового изделия всегда появляется вопрос. Каким путем лучше пойти, чтобы обеспечить требуемую вычислительную мощность изделия: использовать стандартное готовое устройство или же разработать свое собственное?

С одной стороны, используя стандартные устройства, можно значительно сократить время разработки изделия, минимизируя при этом стоимость самой разработки. Особенно это актуально, когда речь идет о небольших серийно выпускаемых партиях изделия. Однако при этом необходимо учитывать, что разработчик зависит от длительности выпуска выбранного устройства. И если производитель устройства по какой-либо причине снимет его с производства, разработку

придется начинать заново. К тому же стандартное устройство выпускается производителем с жестко фиксированными размерами и характеристиками (например, жестко ограниченный набор интерфейсов), и разработчик никак не может влиять на это.

С другой стороны, собственная разработка дает возможность сократить стоимость конечного изделия за счет подбора элементной базы. Но что более важно: разработчик волен сам определять параметры устройства и его характеристики — размер, количество интерфейсов и т. п. Однако при небольших партиях серийно выпускаемого изделия разработка такого устройства станет довольно дорогой. К тому же сама разработка может затянуться на многие месяцы, и можно просто не успеть вывести изделие на рынок либо представить его заказчику к установленному сроку.

Золотая середина между этими двумя путями была найдена во встраиваемых системах на архитектуре x86 в начале 2000 года, когда с появлением первого стандарта «компьютера на модуле» (Computer-on-Module, CoM [1]) стало возможным сократить время разработки изделия, имея при этом возможность определять размеры устройства и необходимые интерфейсы.

Модульный стандарт CoM очень быстро завоевал популярность среди производителей встраиваемых систем во всем мире именно благодаря простоте и скорости разработки готовых изделий с нестандартными требованиями к размеру, интерфейсам и т. п. Также, благодаря модульному принципу стандарта CoM, стало возможным быстрое и безболезненное обновление версий изделия и взаимозаменяемость модулей различных производителей встраиваемых систем между собой. К примеру, один из таких стандартов «компьютеров на модуле» — COM Express — очень хорошо известен в России благодаря таким производителям, как Kontron, RadiSys, Advantech и др. Существуют и другие стандарты «компьютеров на модуле», например ETX и XTX. Тем, кто интересуется ими, мы рекомендуем обратиться к [2, 3].

Однако что касается изделий, использующих процессоры с архитектурой ядра ARM, здесь до сих пор нет единого стандарта раз-



Рис. 1. Модуль SECO Qseven с процессором NVIDIA Tegra2 на ядре Cortex-A9

Таблица. Основные модули SECO Qseven

	Quadmo747	Quadmo747/ EXTREME	Quadmo747-X/ OMAP3	Quadmo747-X/ i.MX51	Quadmo747-X/ i.MX28	Quadmo747-X/T20
ЦП	Intel Atom серии Zxx (1,1 или 1,6 ГГц)	Intel Atom серии ZxxPT (1,1 или 1,33 ГГц)	Texas Instruments OMAP 35xx (Cortex A8 600 МГц – 1 ГГц)	Freescall i.MX513/i.MX515, (OpenGL ES 2.0 3D и OpenVG 1.1 2D графические ускорители)	Freescall i.MX28 с ядром 454 МГц ARM926EJ-S	NVIDIA Tegra T290 с двухъядерным ЦП ARM CORTEX-A9 MPCore, частота 1 ГГц
LVDS	18/24-разрядный, одноканальный (разрешение до 1366×768)			18/24-разрядный LVDS	18/24-разрядный, разрешение до 800×480	18/24-разрядный LVDS, разрешение до 1680×1050
PCI Express	До двух линий ×1		1 линия ×1		–	До двух линий ×1
USB	8×USB 2.0		1×USB OTG 2×USB Host 2.0	1×USB OTG 4×USB Host 2.0		6×USB 2.0
Ethernet	1×GBE	1×GBE (опционально)		10/100 Base-T		
SDIO	1×8-разрядный, внешний интерфейс, 1× слот microSD	1×4-разрядный, внешний интерфейс	MMC/HC MMC/ SD/SDHC/SDIO	MMC/HC MMC/SD/ SDHC/SDIO 1× слот microSD	MMC/SD/SDIO 1× слот microSD	MMC/SD
SATA	1×SATA		–			2×SATA
PC Bus	Да					
LPC	Да					
CAN	–		Да			
Размеры	70×70 мм					

работки. Производитель процессора обязан поддерживать саму архитектуру ядра, но подключение различных устройств периферии остается за разработчиком изделия. Поэтому каждый производитель процессоров с ядром ARM вынужден выпускать свое руководство по разработке, которое специалистам необходимо изучить прежде, чем начинать новую разработку. Таким образом, разработка изделий на ARM требует намного больше времени и серьезных навыков инженеров и программистов, чем разработка на x86.

Казалось бы, применение принципов CoM в случае с процессорами на ядре ARM могло существенно снизить нагрузку на разработчиков, сократив при этом время и стоимость самой разработки так же, как это было сделано и с процессорами на x86. Но вся проблема состояла в том, что до 2010 года не было стандарта «компьютера на модуле» для процессоров с ядром ARM.

Ситуация изменилась после того, как компания SECO [4] разработала спецификацию нового стандарта CoM — Qseven [5]. Как и остальные стандарты «компьютеров на модуле», стандарт Qseven дает логическое и электрическое описание всех электрических сигналов, используемых в стандарте, и тип электромеханического разъема между двумя

частями CoM — вычислительным (или иначе процессорным) модулем и платой-носителем (или кросс-платой). На процессорном модуле расположен центральный процессор, оперативная память и пр. Другими словами, процессорный модуль — это вычислительная часть устройства, которая изготавливается тем или иным производителем. Кроме того, в версии 1.2 стандарта Qseven стало возможным использовать в одном и том же устройстве как модули с процессорами x86 (Intel, AMD, VIA), так и модули с процессорами на ядре ARM (Texas Instruments OMAP3 и OMAP4, Freescall i.MX28/i.MX51, NVIDIA Tegra2 и др.) [6].

Внешний вид модуля Qseven с процессором NVIDIA Tegra2 от фирмы SECO представлен на рис. 1, а его характеристики приведены в таблице. Здесь же в таблице даны краткие характеристики основных модулей SECO Qseven с описанием используемых процессоров и интерфейсов. Используя эту таблицу, разработчик может выбрать модуль именно с теми характеристиками, которые ему необходимы для решения определенной задачи.

В итоге, руководствуясь стандартом Qseven, разработчику остается сконструировать кросс-плату, на которую выводятся интерфейсы с процессорного модуля через элек-

троемеханический разъем MXM, имеющий 230 контактов. При этом кросс-плата может содержать только те необходимые интерфейсы, которые будут использоваться в изделии, и не содержать лишних. А так как вся вычислительная часть, включая процессор, вынесена на модуль, то разработка всего устройства (по сути, разработка кросс-платы) значительно дешевле (и несравнимо проще), чем разработка платы, содержащей на себе процессор на ядре ARM. Кроме того, вполне реальную помощь может оказать и сам производитель. Так, компания SECO предоставляет электрическую схему собственной кроссплатформенной отладочной платы с описанием всех интерфейсов в формате OrCad. К примеру, часть электрической схемы, включающей описание последовательного интерфейса и интерфейса Ethernet, приведена на рис. 2.

Если рассматривать стандарт Qseven более подробно, то можно отметить следующие основные особенности, выгодно отличающие его от других стандартов «компьютеров на модуле»:

1. Стандарт дает полное логическое описание всех сигналов, получаемых из модулей с процессорами как на ядре ARM, так и на архитектуре x86. В том числе это касается:

- высокоскоростных интерфейсов, таких как PCI Express, SATA, USB 2.0, Gigabit Ethernet и т. д.;
- графических интерфейсов, таких как LVDS, SDVO, HDMI и DisplayPort;
- основных промышленных интерфейсов (включая CAN, SPI, SDIO, I²C и т. п.) (рис. 3).

2. В стандарте приведено полное электрическое описание сигналов указанных интерфейсов.

3. В стандарте дано полное конструкторское описание соединительного разъема — недорогого и доступного электромеханического разъема MXM, с указанием местоположения сигналов на разъеме.

4. В стандарте дано конструкторское описание самих модулей. Их размеры всего 70×70 мм, тогда как для COM Express это 125×95 мм, а для ETX/XTX 114×95 мм. Готовые одноплатные компьютеры формата PC/104 имеют больший размер: 96×90 мм (рис. 4).

Эта информация носит открытый характер и доступна любому желающему в документации на стандарт Qseven [7, 8].

Эти особенности делают стандарт Qseven уникальным среди всех стандартов «компьютеров на модуле», так как дают разработчикам возможность использовать любую аппаратную платформу, концентрируясь при этом только на конечных требованиях к устройству. Помимо этого стандарт Qseven позволяет значительно сократить время разработки и получить прототип готового устройства как можно раньше, так как конструировать необходимо не сам процессорный модуль, а кросс-

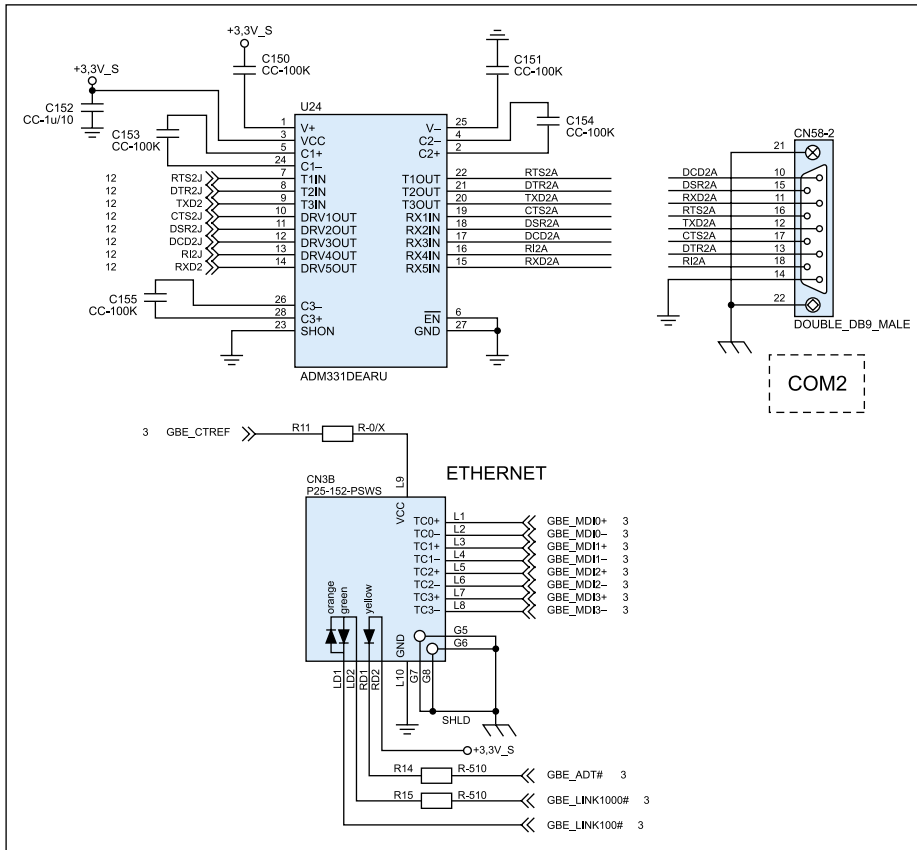


Рис. 2. Электрическая схема кроссплатформенной отладочной платы SECO для модулей Qseven в части интерфейсов COM и Ethernet

ра на модуле» приходит на помощь. Давайте рассмотрим это на примере отладочного комплекта Qseven компании SECO.

Как известно, имея на руках отладочную плату того или иного процессора с ядром ARM, можно приступить к разработке программного обеспечения. Однако, к сожалению, такая отладочная плата может не содержать всех интерфейсов, необходимых для разрабатываемого устройства. В результате программное обеспечение на такой отладочной плате нельзя протестировать на все 100%, и возможные проблемы с подключением нужной периферии не будут обнаружены.

С другой стороны, имея на руках отладочный комплект SECO (рис. 5), разработчик получает в свое распоряжение кроссплатформенную отладочную плату, на которую выведены абсолютно все интерфейсы, какие только можно получить с модуля Qseven (вне зависимости от того, на каком процессоре он выполнен) (рис. 6). Поэтому, имея эту отладочную плату, можно приступить к разработке и тестированию программного обеспечения, используя всю необходимую периферию. Кроме того, на каждый процессорный модуль SECO, выполненный в стандарте Qseven, предоставляется пакет драйверов (BSP) и модулей таких операционных систем, как Windows CE и Linux, а также на старшую версию популярной в последнее время ОС Android. Все BSP сопровождаются необходимой документацией и утилитами, благодаря чему работа по разработке ПО сокращается до минимума.

Заключение

Таким образом, очевидно, что можно в несколько раз сократить время разработки нового изделия с процессором на ядре ARM, используя при этом преимущества стандарта CoM. Так, используя в своем изделии процессорные модули стандарта Qseven, разработчик может значительно сократить время разработки аппаратной части устройства. А используя отладочный комплект с кроссплатформенной платой, параллельно с раз-

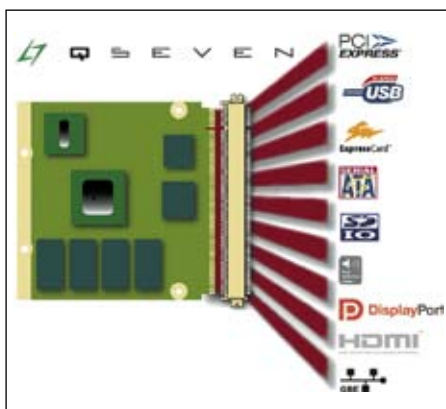


Рис. 3. Основные интерфейсы Qseven с высокой скоростью передачи данных

плату с набором необходимых интерфейсов. Соответственно, нет необходимости изучать руководство по разработке на тот или иной процессор, документацию на него и список изменений его последней версии. Поэтому время разработки аппаратной части устройства значительно сокращается.

Однако если с аппаратной частью все ясно, то остается последний серьезный вопрос — разработка программного обеспечения (или его модернизация). И если для архитектуры x86 данный вопрос не стоит столь остро (ведь программа, написанная для DOS 10 лет назад, и сегодня будет работать на любом новейшем процессоре Intel), то для ARM, как это кажется на первый взгляд, может стать проблемой. Но и тут принцип «компьюте-

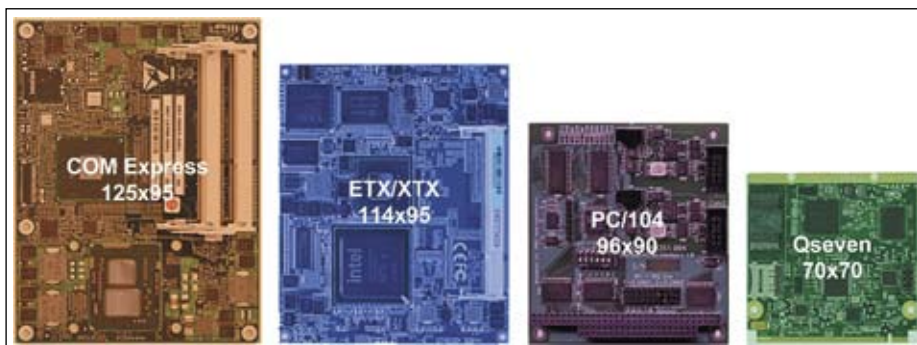


Рис. 4. Сравнение размеров основных стандартов CoM



Рис. 5. Отладочный комплект SECO для модулей Qseven

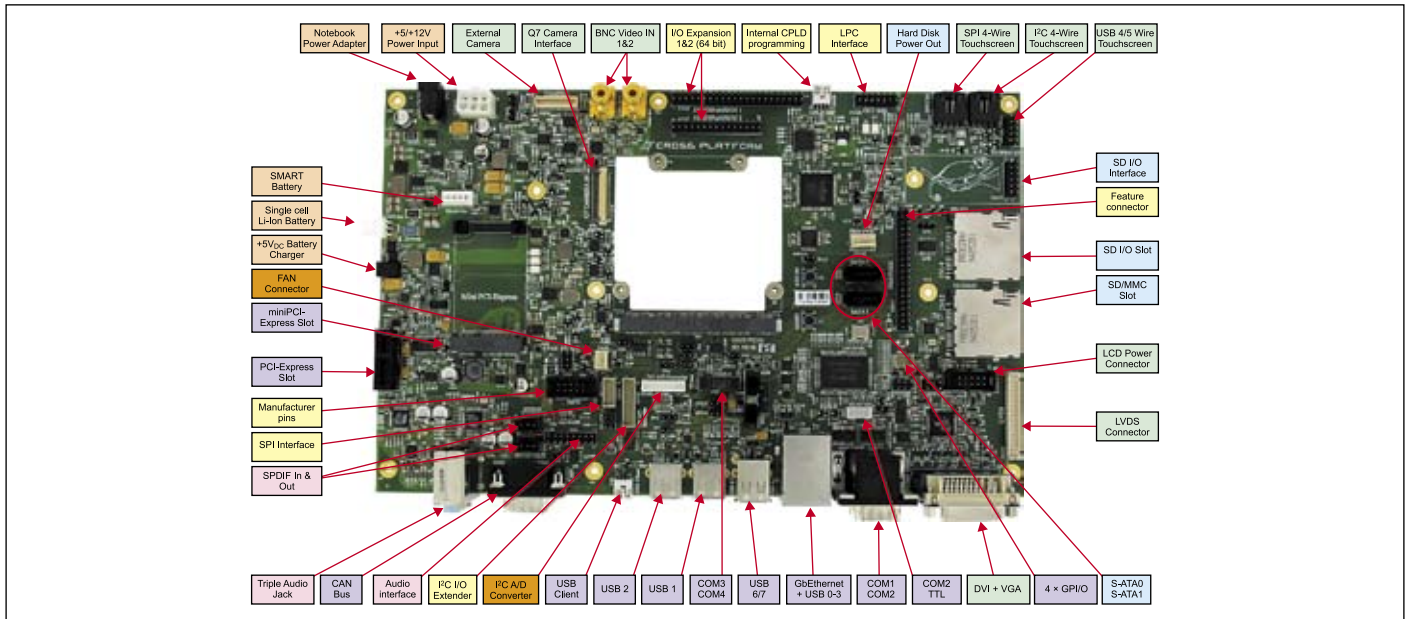


Рис. 6. Кроссплатформенная плата SECO для модулей Qseven

работкой аппаратной части он может выполнять разработку (либо проводить модернизацию) программного обеспечения для любого выбранного процессора на ядре ARM. В итоге стандарт Qseven дает возможность разработчику получить все необходимое для разработки программной и аппаратной части изделия независимо друг от друга, тем самым можно сократить общее время разработки изделия. ■

Литература

1. Румянцев С. Архитектура и стандарт COM Express // Мир электронных компонентов. 2008. Вып. 3.
2. COM Express Carrier Design Guide, Rev. 1.0. C. Eder. Jan 16, 2009.
3. XTX pinout, signal description and implementation guidelines, Rev. 1.2. XTX Consortium. Dec. 5, 2006.
4. www.secoqseven.com
5. www.qseven-standard.org
6. <http://www.eltech.spb.ru/profiles.html?company=SECO>
7. Qseven pinout, electromechanical description and implementation guidelines. Rev. 1.20. Qseven Consortium. 12.08.2010.
8. Qseven Design Guide, Revision 1.0. Qseven Consortium. 18.04.2011.
9. www.picmg.org