ВИРТУАЛИЗОВАННОЕ ПРОЕКТИРОВАНИЕ ВСТРАИВАЕМЫХ ЭЛЕКТРОННЫХ СИСТЕМ

АЛЕКСАНДР КОЛАЕВ, техн. консультант, «ИД Электроника»

Статья посвящена вопросу виртуализации электронных устройств. Дано описание основных подходов к созданию виртуальных платформ. Приведены рекомендации по выбору метода виртуализации встраиваемых систем и советы по организации виртуального проектирования.

ВВЕДЕНИЕ

Виртуализация — многоплановое понятие. В широком смысле виртуализация — это абстракция некоторого процесса или объекта, скрывающая его настоящую реализацию.

Продуктом виртуализации является нечто удобное для использования, но на самом деле имеющее более сложную или совсем иную структуру, отличную от той, которая воспринимается объектом. Виртуализация делится на программную и аппаратную.

Программная виртуализация основана на создании виртуальной платформы (машины) на базе имеющихся аппаратных и программных средств. Аппаратная виртуализация предполагает комбинирование или упрощение представления аппаратных ресурсов для пользователя и получение пользовательских абстракций оборудования, пространств имен, сетей и т.д.

Технологии виртуализации позволяют запускать на одном физическом компьютере (хосте) несколько виртуальных экземпляров операционных систем (гостевых ОС) в целях обеспечения их независимости от аппаратной платформы и сосредоточения нескольких виртуальных машин на одной физической. Виртуализация предоставляет множество преимуществ как для инфраструктуры предприятий, так и для конечных пользователей. За счет виртуализации происходит значительная экономия на аппаратном обеспечении и облуживании, повышается гибкость IT-инфраструктуры, упрощается процедура резервного копирования и восстановления после сбоев. Не зависимые от конкретного оборудования виртуальные машины могут распространяться в качестве предустановленных шаблонов и работать на любой аппаратной платформе поддерживаемой архитектуры.

Виртуальная машина (ВМ) — программная система, созданная на основе существующих аппаратно-программных комплексов. Операционная система, предоставляющая аппаратные ресурсы и программное обеспечение, называется хостовой (host), а симулируемые ей системы — гостевыми (guest). Для стабильной работы гостевых систем необходимо, чтобы программное и аппаратное обеспечение хоста было достаточно надежным и предоставляло требуемый набор интерфейсов для доступа к его ресурсам. Имеется несколько видов виртуализации платформ, в каждом из которых осуществляется свой подход к понятию «виртуализация». В основном они различаются степенью полноты симуляции аппаратного обеспечения (полная, частичная, виртуализация адресного пространства, приложения, ОС и т.д.).

Гипервизор, или монитор виртуальных машин (VMM — virtual machine monitor), — программная среда или аппаратная схема, координирующая одновременную работу нескольких ОС на одном физическом компьютере. При этом каждая гостевая ОС работает так, как если бы она была запущена на отдельном компьютере. Гипервизор обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между запущенными ОС и управление ими.

Гипервизор должен поддерживать не только ОС общего назначения, но и ОСРВ, а также уметь эмулировать работу предыдущих версий аппаратной реализации, чтобы отлаживать переход ОС на новую платформу.

Аппаратная виртуализация выполняется на уровне архитектуры процессора и заключается в том, что когда гостевая операционная система генерирует исполняемую процессором привилегированную команду, то процессор прерывает свою работу и передает управление гипервизору, который должен решить, выполнять команду или эмулировать ее выполнение другими средствами. Гипервизор способен перехватить не все «проблемные» команды, а только привилегированные.

ОТЛИЧИЕ АППАРАТНОЙ ВИРТУАЛИЗАЦИИ ОТ ПРОГРАММНОЙ

Классическая архитектура программной виртуализации подразумевает наличие хостовой операционной системы, поверх которой запускается платформа виртуализации, эмулирующая работу аппаратных компонентов и управляющая аппаратными ресурсами в отношении гостевой ОС. Реализация такой платформы достаточно сложна и трудоемка и нередко приводит к потере производительности. Безопасность виртуальных машин также находится под угрозой, поскольку получение контроля над хостовой операционной системой автоматически означает его установление над всеми гостевыми системами.

В отличие от программной техники, с помощью аппаратной виртуализации можно получить изолированные гостевые системы, управляемые гипервизором напрямую. Такой подход может обеспечить простоту реализации ВМ и увеличить ее надежность при одновременной работе с несколькими гостевыми ОС. Кроме того, благодаря отсутствию потерь быстродействия на обслуживание хостовой системы производительность гостевых ОС близка к производительности реальных.

Аппаратная виртуализация имеет следующие преимущества над программной.

– Упрощение разработки платформ виртуализации за счет предоставления аппаратных интерфейсов управления и поддержки виртуальных гостевых систем.

– Возможность увеличения быстродействия платформ виртуализации за счет использования гипервизора.

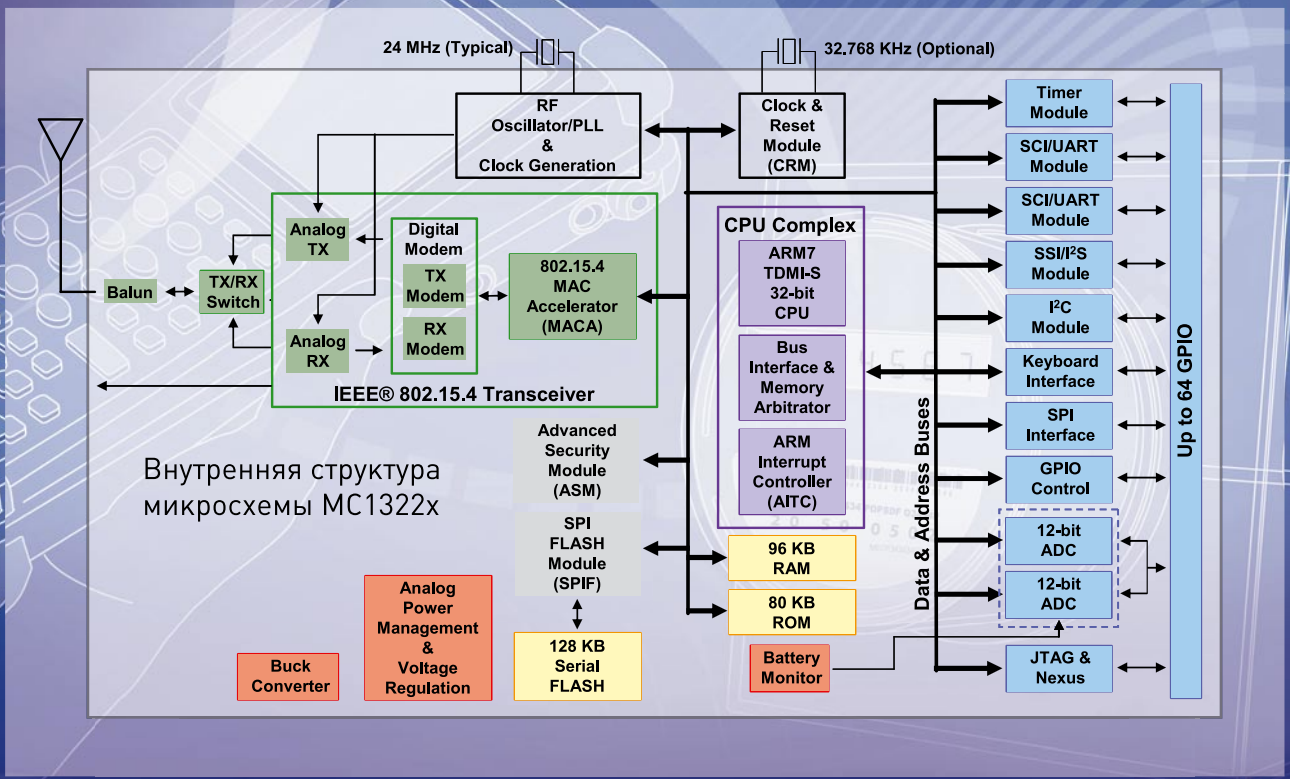
– Возможность независимой работы нескольких ВМ и переключения между ними на аппаратном уровне. При этом повышается быстродействие и защищенность виртуальных машин.

– Независимость гостевой ОС от архитектуры хостовой платформы и ВМ. Например, с помощью технологий аппаратной виртуализации возмо-

freescale semiconductor

ОДНОКРИСТАЛЬНОЕ ZIGBEE-РЕШЕНИЕ ОТ FREESCALE

Не требует внешних компонентов!



Внутренняя структура микросхемы MC1322x

Симметрон
ГРУППА КОМПАНИЙ

Москва (495) 797-5545
moscow@symmetron.ru
Санкт-Петербург (812) 449-4000
spb@symmetron.ru
Новосибирск (383) 211-9295
sibir@symmetron.ru
Киев +38 (044) 239-2065
kiev@symmetron.ua
Харьков +38 (057) 750-8022
kharkov@symmetron.ua
Минск +375 (17) 209-8020
minsk@symmetron.ru
Донецк +38 (062) 337-1183
donetsk@symmetron.ua

www.symmetron.ru

жен запуск 64-битных гостевых ОС на 32-битных хостовых системах.

ВИРТУАЛИЗАЦИЯ ВСТРАИВАЕМЫХ ПРИЛОЖЕНИЙ

В рассмотренных нами подходах преследовалась одна и та же цель: максимально эффективно использовать имеющиеся аппаратные ресурсы между несколькими ОС. Для встраиваемых систем, где основное значение имеют предсказуемость работы (детерминизм) и возможность взаимодействия с нестандартным или специальным аппаратным обеспечением, требуется другой подход.

Главное различие между гипервизором для сервера и встраиваемой системы лежит в способе распределения физических ресурсов между виртуальными машинами. Гипервизор для ВС разделяет циклы ЦП, ячейки ОЗУ и порты ввода-вывода между всеми гостевыми ОС, а не мультиплексирует эти ресурсы между ВМ, как это происходит в случае виртуализации серверов. Такой подход также называют ассиметричной многопроцессорной обработкой. Он применяется там, где необходим детерминизм и высокое быстродействие, а не одинаковый уровень доступа для различных ОС или наиболее полное использование ресурсов имеющихся аппаратных средств.

Средства виртуализации, включенные в большинство современных многоядерных процессоров, могут быть использованы для резервирования ресурсов под нужды конкретной ВМ и ее гостевой ОС. Заметим, что модель ассиметричной обработки вовсе не предполагает назначение всех портов ввода-вывода только одной ВМ. Некоторые из них, например жесткий диск, консоль или адаптер Ethernet, могут быть общими. В этом случае гипервизор создает виртуальные устройства, чтобы упростить разделение аппаратного обеспечения между ВМ.

Детерминизм обеспечивается за счет использования многоядерных процессоров, что позволяет гипервизору отвести каждой гостевой ОС свое ядро. При этом отпадает необходимость в разделении ресурсов или времени ЦП, которые нежелательны в системах реального времени. Если в процессоре имеется достаточное количество ядер, то этот подход является самым низкозатратным и простым.

ПРОЕКТИРОВАНИЕ ВИРТУАЛИЗИРОВАННЫХ ВСТРАИВАЕМЫХ СИСТЕМ

Проектирование виртуализированных встраиваемых систем (VSD — virtualized systems development) позволяет делать то, что раньше было невозможным. Выгоды данного подхода проявляются на всех стадиях проектирования, от определения продукта до его применения.

При виртуализованном проектировании разработчики аппаратного и программного обеспечения могут одновременно работать над одними и теми же модулями и взаимодействовать более плотно, что в конечном счете позволяет отслеживать и устранять недоработки на ранней стадии, сокращая затраты и время выхода устройства на рынок.

С ростом сложности встраиваемых электронных систем потребность в различных инструментальных средствах проектирования становится более очевидной. Даже небольшие современные устройства могут содержать несколько процессоров, DSP, СБИС, ПЛИС и другие устройства. В дополнение к этому в системе может использоваться сразу несколько операционных систем и стеков приложений.

К сожалению, средства разработки не поспевают в развитии за встраиваемыми системами. В результате разработчики сталкиваются с проблемой поиска новых путей для определения системных архитектур, разработки и интеграции миллионов строк кода, а также для создания таких сложных систем. С одной стороны, они должны сократить время разработки, с другой — представить качественный и простой в поддержке и обслуживании продукт.

Высокое быстродействие — одно из свойств, отличающих VSD-платформы от САПР и подобных средств. Хотя они и точные, но их быстродействия не хватает для эмулирования работы ОС, приложений или системных программ.

ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

Виртуальные платформы лучше всего охарактеризовать как функциональные модели физического оборудования. Они, как маяки, служат ориентиром и конечной целью разработки программного обеспечения. Эти модели настолько точны, что ПО не сможет отличить их от реального оборудования. Таким образом, для симуляции не требуется писать дополнительный код или вносить изменения в ПО, которые потенциально могут повлиять на работу встраиваемой системы. Кроме того, программную и схемотехническую реализацию модуля можно разрабатывать одновременно.

В отличие от традиционного, виртуализованное проектирование обладает следующими достоинствами.

1. Гибкость. Виртуальные платформы развиваются и дополняются постепенно, по мере проектирования, помогая разработчикам программного обеспечения на всем пути от идеи устройства до окончательной модели полной системы.

На определенных стадиях такие блоки как устройства ввода-вывода, СБИС и модули памяти удобно или «выкинуть» из системы, или упростить, заменив стандартными компонентами, аналогичными по функциональности. Виртуализованное проектирование позволяет сделать это.

Кроме того, группы, разрабатывающие аппаратное и программное обеспечение, могут экспериментировать с новыми архитектурами, создавая несколько виртуальных платформ и сравнивая их характеристики. Впоследствии эти платформы можно использовать при создании прототипа и отладке проекта на стадии определения архитектуры.

2. Упрощенная среда проектирования. Поскольку виртуальная платформа прогоняет те же двоичные файлы программного обеспечения, которые будут запускаться впоследствии готовой физической системой, отсутствует необходимость в создании специальных модулей для кросс-компиляции, симуляции и отладки. Этот аспект виртуализованного проектирования может значительно уменьшить количество версий кода, сократив расходы на обслуживание и снизив риск появления ошибок из-за несовместимости версий.

3. Возможность создания библиотек моделей. Все разработанные виртуальные платформы можно использовать повторно, ускоряя разработку последующих продуктов или обновлений существующих устройств.

Однако не всегда использование виртуализованного проектирования оправданно. Чтобы понять, следует ли его применять, необходимо выделить проблемы, возникающие при проектировании, риски и затратность, а затем оценить, насколько виртуализованная реализация поможет снизить эти издержки.

Кроме того, важно понимать, что переход на новый принцип проектирования сопровождается фундаментальными изменениями налаженного процесса. Это задача не решается быстро, а требует временных и материальных затрат на внедрение нового подхода, обучение персонала и т.д. Для проектов, имеющих долгосрочные перспективы на рынке, либо очень сложных и дорогостоящих встраиваемых систем, требующих тщательной настройки и отладки, этот переход оправдан, поскольку все затраты быстро окупаются.

В целях снижения рисков испытывать виртуализованный метод разработки следует в первую очередь на тех продуктах, которые позволят сделать это проще всего. Например, к ним относятся устройства, состоящие из стандартных и недорогих блоков или имеющие несложную программную реализацию.

РЕКОМЕНДАЦИИ

Для максимальной реализации возможностей виртуализованного проектирования следует придерживаться нескольких правил, которые помогут сделать проект универсальным и годным для многократного применения.

– Использование функции импорта стандартных регистровых языков (например, IP-XACT от SPIRIT Consortium) и моделей (например, TLM-2.0 от SystemC).

– Включение поддержки нескольких языков моделирования для обеспечения гибкости и межоперационной совместимости.

– Использование нескольких уровней абстракции. Для большинства компонентов модели в зависимости от программной задачи, которую они выполняют, требуется различный уровень функциональности. Среда моделирования должна поддерживать различные уровни абстракции, от максимального упрощения представления системы (путем удаления некоторых блоков) до моделирования полной функциональности.

– Создание масштабируемых моделей. Крупные и сложные модели следует собирать из более мелких. При этом предполагается, что инженерные группы имеют доступ не только к собственным моделям, но и к разработанным другими коллективами.

– Разработка интерфейса с физическим оборудованием. Необходимо предусмотреть возможность подключения виртуальных платформ к физическому оборудованию через стандартные интерфейсы связи или RTL-эмуляторы, чтобы расширить сферу применения разработанных виртуальных платформ.

Чтобы модели отвечали требованиям, предъявляемым к реальным системам, они должны обладать следующими свойствами.

– Гибкость и модульность. Сложные модели должны состоять из более простых. Следует, однако, иметь в виду, что точные модели физического оборудования могут оказаться слишком сложными и непригодными для использования в других проектах. Чтобы этого избежать, необходимо тщательно проанализировать функции, которые должно выполнять ПО, и определить, какие регистры и процессы должны быть смоделированы отдельно.

– Поддержка различных типов СБИС и многоядерных процессоров.

– Поддержка смешанных архитектур процессоров.

– Использование ОС смешанного типа (поддержка ОСРВ, работа с хостовой ОС и без нее, с гипервизором и без него). Это необходимо для обеспечения работы на любой платформе.

– Поддержка основных стандартов передачи и связи (Ethernet, PCI, PCI-Express, RapidIO, MIL-STD-1553, ARINC 429, SpaceWire, FireWire, USB, ATM, последовательный порт и др.).

– Наличие интерфейса между виртуальной и физической реализациями для обеспечения взаимодействия между ними.

– Возможность установки контрольных точек и точек обратимости, которые позволят вернуться к известному состоянию системы. Это важно для совместной работы инженерных групп.

Для удобства разработчиков ПО среда проектирования должна иметь следующие возможности и свойства.

– Наличие привычных инструментов проектирования. Разработчики ПО для виртуальных платформ должны иметь возможность применять те же инструменты разработки программного обеспечения (например, компиляторы, линкеры, отладчики и IDE), что и для программирования физического оборудования.

– Возможность отладки системы в неактивном состоянии.

– Повторяемость. Эта функция позволяет несколько раз моделировать одну и ту же конфигурацию, чтобы несколько разработчиков могли видеть результаты и совместно отлаживать проблемные места кода.

– Контрольные точки и снимки системы. С их помощью можно сохранять и впоследствии полностью восстанавливать состояние системы на определенном этапе. Это позволяет экономить время за счет создания одной многоразовой аппаратно-программной установки для нескольких разработчиков.

– Расширенная отладка — поддержка аппаратных и программных точек останова и выполнения операций в обратном порядке.

– Полный обзор и контроль аппаратных схем. Разработчики должны иметь возможность видеть регистры и данные, которые будут невидимыми на физическом оборудовании.

– Использование скриптов для автоматизации процессов тестирования и оценки системы.

ВЫБОР МЕТОДА ПРОЕКТИРОВАНИЯ

Как говорилось выше, существует несколько подходов к созданию виртуальных платформ, каждый из которых имеет свои сильные стороны. Рассмотрим основные вопросы, которые помогут выбрать наилучший метод.

1. Аппаратная или программная виртуализация?

Существуют две основные категории виртуальных платформ: программные и аппаратные. В программной реализации на одной виртуальной машине поверх хостовой ОС запускается несколько гостевых, а при аппаратной реализации гипервизор запускает одновременно несколько ВМ и осуществляет контроль над обменом данными на низком уровне без использования ОС хоста.

Для большинства встраиваемых приложений больше подходит аппаратная виртуализация, поскольку гипервизоры поддерживают ОСРВ и открывают для гостевой ОС прямой доступ к аппаратным средствам.

Для тестирования ПО, находящегося на стадии проектирования, или запуске приложений, рассчитанных для разных ОС, лучше выбрать программную виртуализацию, когда гипервизор работает через хостовую ОС.

Существуют два базовых программных метода виртуализации: динамической трансляции и модификации гостевой ОС (паравиртуализация). При динамической, или бинарной, трансляции проблемные команды гостевой ОС перехватываются гипервизором и модифицируются (заменяются на безопасные), после чего управление возвращается гостевой ОС.

В результате гостевая операционная система вместе со всеми приложениями фактически становится одним «обычным» приложением хостовой операционной системы, из-под которой она запущена.

При паравиртуализации исходный код гостевой ОС частично подвергается модификации: все проблемные

инструкции и спорные места заменяются командами перехода на гипервизор, который генерирует код замещения проблемной команды. Этот метод позволяет добиться более высокой производительности, однако он применим лишь в том случае, если гостевые ОС имеют открытые исходные коды — их можно модифицировать в соответствии с лицензией.

Не стоит забывать об обратной стороне использования гипервизора. Обращение к нему требует резервного копирования и восстановления данных о состоянии. Это снижает быстродействие системы и в некоторых случаях недопустимо. Наиболее эффективные виртуализованные решения сводят к минимуму обращения к гипервизору, чтобы минимизировать временные задержки.

2. Какие ОС и наборы инструкций должен поддерживать гипервизор?

При проектировании встраиваемой системы следует закладывать в нее возможности расширения и модернизации. Соответствующие изменения в архитектуре (переход на новый процессор, добавление ОС и т.д.) необходимо продумать и учесть заранее.

3. Должны ли ОС иметь возможность взаимодействовать друг с другом?

Одно из преимуществ запуска нескольких операционных систем на одном компьютере — это прямые связи между ОС. Как правило, гипервизоры поддерживают два способа обмена данными между ОС: эмулированные интерфейсы и общая память. Эмулированные интерфейсы — это физические или виртуальные устройства, которые гипервизор представляет гостевым ОС. При использовании общей памяти в ОЗУ отводится область, которую используют обе ОС.

4. Следует ли изолировать виртуальные машины друг от друга и в какой степени?

Виртуализация всегда хотя бы частично изолирует гостевые ОС. Некоторые гипервизоры гарантируют полное разграничение, когда одна ОС не имеет доступа к данным или командам другой ОС. Полная изоляция является обязательным требованием в таких областях как безопасность, военная и авиакосмическая промышленность.

Хотя большинство встраиваемых систем предназначено для длительной непрерывной работы, необходимо предусмотреть реакцию гостевых ОС на сбой или перезагрузку одной из них. Для осуществления независимой перезагрузки какой-либо из гостевых ОС в виртуальной машине должна быть предусмотрена возможность корректного сброса устройств ввода-вывода без перезагрузки всей машины.

Если операционные системы связаны (например, одна ОС имеет доступ к винчестеру через другую), то независимая перезагрузка ОС может оказаться сложной задачей.

5. Какой гипервизор лучше: стандартный (готовый) или требующий настройки?

На установку гипервизора низкого уровня может уйти несколько недель или месяцев. Особенно это касается аппаратных гипервизоров и гипервизоров, модифицирующих код гостевой ОС. Такая тонкая настройка позволяет увеличить производительность, поэтому она оправдана для массовых продуктов. В более мелких проектах лучше воспользоваться стандартными версиями, чтобы сэкономить время и облегчить задачу разработчиков.

6. Требуется ли поддержка многопроцессорной обработки?

Гипервизоры могут поддерживать симметричную (SMP), асимметричную многопроцессорную обработку (AMP) или обе одновременно. Эти функции используются в электронных системах с несколькими процессорами. Если ядра разные, то необходима поддержка AMP, если одинаковые — то SMP. Если же ВМ использует только один процессор, то для запуска нескольких ОС одновременно применяется разделение процессорного времени. Соответственно, гипервизор должен поддерживать эту функцию.

7. Каковы лицензионные обязательства и условия предоставления технической поддержки?

Выше мы рассматривали только технические аспекты, однако имеется и экономическая составляющая.

Любое выбранное решение требует покупки лицензии на его использование. Кроме того, возможна дополнительная оплата каждого проекта или ежегодные комиссии. В случае выбора гипервизора низкого уровня обратите внимание на условия обслуживания и поддержки, поскольку при его настройке у вас, наверняка, появятся вопросы.

Кроме того, важным аспектом является обновление гипервизора для поддержки новых ОС и процессоров. Стоимость всех этих услуг следует уточнить заранее.

ЗАКЛЮЧЕНИЕ

Виртуальное проектирование дополняет процесс традиционной аппаратно-ориентированной разработки встраиваемой системы и предоставляет большие преимущества, с точки зрения эффективности, качества программного обеспечения, а также сокращения сроков поставки. Однако без полнофункциональной среды моделирования и достаточного набора стандартных и пользовательских моделей все выгоды сводятся на нет.

Выбор наилучшего решения зависит от многих факторов: требуемых характеристик, типа виртуальной платформы, количества проектов и т.д. Единого универсального решения нет.

Правильно организованный подход позволяет существенно сократить затраты на разработку, упростить взаимодействие между разработчиками и повысить качество самого продукта.

ЛИТЕРАТУРА

1. Paul Fisher. Getting a Handle on Virtualization and Putting it to Work//RTC. Сентябрь. 2008 г.

2. David Beal. Reducing Costs, Risks, Time to Market with Virtualized Systems Development// www.embedded.com/design/222700419.

3. А. Самойленко. «Виртуализация: новый подход к построению IT-инфраструктуры»//www.ixbt.com/cm/virtualization.shtml.

4. Casey Weltzin. 10 questions to ask when choosing a virtualization solution// www.eetimes.com/news/design/showArticle.jhtml?articleID=224000219.

5. Н. Елманова, С. Пахомов. «Виртуальные машины 2007»//КомпьютерПресс. №9. 2007.

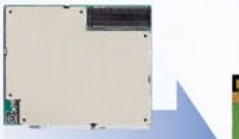
ЗАО "ГРАНИТ-ВТ" т./ф.: 8 (812) 274-04-48, e-mail: mail@granit-vt.com, www.granit-vt.ru 191014, Санкт-Петербург, ул. Госпитальная 3

MT system
www.mt-system.ru

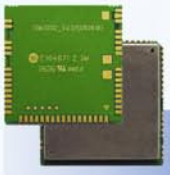
МТ-СИСТЕМ - ОФИЦИАЛЬНЫЙ АВТОРИЗОВАННЫЙ ДИСТРИБЬЮТОР SIMCom В РОССИИ

SIMCom
A company of SIM Tech

Новые GSM/GPRS модемы


SIM300Z

SIM900B


SIM300DZ

SIM900D

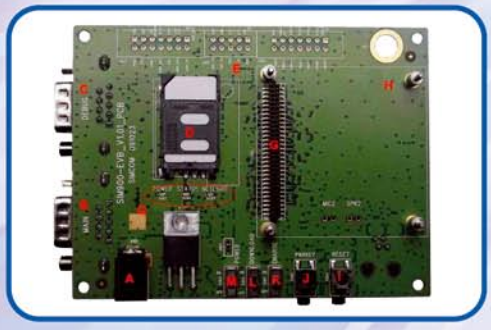
НОВЕЙШИЙ МОДУЛЬ

SIM900

Новая линейка модулей SIMCOM SIM900 в сравнении со старой SIM300

Наименование	SIM900	SIM900B	SIM900D	SIM300Z	SIM300DZ
Диапазон частот	GSM 850/900/1800/1900			GSM 900/1800/1900	
Класс GPRS	10B (4+2, всего 5), скорость 85,6/42,8 кит/с			8B (4+1), скорость 53/26,8 кит/с	
Размеры	24x24x3 мм	40x33x3 мм	33x33x3 мм	40x33x2,85 мм	33x33x3 мм
Масса	3,4 г	7 г	6,2 г	8 г	7 г
Рабочий диапазон температур	-40…+85°C			-30…+80°C	
Системный разъем	Нет	S00016300094	Нет	210-106001-001	Нет
ВЧ разъем	Нет	U-FL	Нет	GSC	Нет
Совместимость	Новый модуль	SIM300Z	SIM300DZ	Рекомендуемая замена SIM900B	Рекомендуемая замена SIM900D
Отладочные средства	SIM900EVB KIT+SIM900TE	SIM900B EVB KIT	SIM900EVB KIT+SIM900DTE	SIM300 EVB KIT	
Платформа	ARM926EJ-S (ARM9)			AD6720 (ARM7)	
Встроенный TCP/IP стек	Есть, мультисокетный (до 8 соединений)			Есть, односокетный	
Возможность установки держателя SIM карты	-	Есть	-	-	-
Напряжение питания	3,2…4,8 В	3,2…4,8 В	3,2…4,8 В	3,4…4,5 В	3,4…4,5 В
Аудио интерфейс	1	2	2	2	2

Отладочные и демонстрационные средства


SIM900&SIM900D EVB KIT

SIM900B EVB

Под проект предоставление полной технической документации, включая обновление версий ПО.

техническая поддержка • сопровождение проектов

Санкт-Петербург:
Ул. Калинина, 13
Тел.: (812) 325-3685
Факс: (812) 786-8579
e-mail: micro@mtgroup.ru

Москва:
1-й Котляковский переулок, 4
(ст. м. “Каширская”)
Тел/факс: (495) 988 20 73, 988 20 74
e-mail: info@mosmtgroup.ru

MT system
www.mt-system.ru

NXP НАВИС RENESAS SAMSUNG Tyco Electronics IXYS Panasonic SIMCom muRata Murata Power Solutions Futaba VARITRONIX ON Semiconductor Vincotech molex