

# WINDOWS В ПРИЛОЖЕНИЯХ РЕАЛЬНОГО ВРЕМЕНИ

**АЛЕКСЕЙ ИСАЕВ**, директор направления системного ПО, «РТСофт»  
**ЛЕОНИД АКИНШИН**, обозреватель, «МКА:ВКС»

*Пока существуют операционные системы (ОС) настольного класса, будет существовать спрос на решения, позволяющие использовать такие ОС во встраиваемых системах (ВС) и в задачах реального времени. Одним из наиболее эффективных решений такого типа, охватывающих широкий диапазон прикладных областей, является продукт RTX компании IntervalZero.*

Если бы на рынке осталась всего одна ОС, отвечающая требованиям абсолютно всех задач — офисных, встраиваемых, ответственных, реального времени и т.д. — то, как это ни странно, такая фантастическая ситуация устроила бы очень многих, если не всех. Получив опыт работы с определенной ОС в определенном классе приложений, многие хотели бы использовать этот опыт, а, следовательно, и саму ОС, в других сферах, включая те, для которых она изначально не предназначалась. Весьма успешной попыткой воплотить мечту об одной ОС для всех рынков выглядит расширение RTX (Real Time eXtension), добавляющее в стандартные офисные и встраиваемые ОС семейства Windows (кроме линейки Windows CE) поддержку реального времени.

## ПОЯВЛЕНИЕ WINDOWS NT

Реальный шанс для воплощения основной идеи RTX — сделать так, чтобы популярную недорогую ОС общего назначения можно было комфортно использовать во встраиваемых системах и системах жесткого реального времени — появился лишь с началом эпохи Windows NT. Став, по сути, первой профессиональной ОС с человеческим лицом, платформа Windows NT открыла новую эру в истории корпорации Microsoft и в общей истории операционных систем.

Что касается рынка встраиваемых компьютерных технологий (ВКТ), то роль Windows NT в его формировании и развитии тоже весьма велика. В середине 1990-х гг. ВС на базе Windows NT стали возникать массово. Для комфортного использования Windows NT во встраиваемых системах и задачах реального времени разработчикам не хватало лишь трех условий:

- возможности уменьшать размеры ОС до приемлемых величин;
- средств обеспечения работы ОС в бездисковых конфигурациях (из ПЗУ

или флэш-памяти), в т.ч. без монитора и без клавиатуры;

- средств обеспечения детерминированного поведения.

## КОМПАНИЯ VENTURCOM

Осознав эти пользовательские потребности, компания выпустила свои теперь уже легендарные продукты Component Integrator и RTX. Component Integrator представлял разработчику ОС Windows NT в качестве «кубиков» — отдельных компонентов, из которых он мог легко собрать ОС с необходимой только ему функциональностью, расширение же RTX позволяло добавлять в обычную настольную ОС Windows NT поддержку работы в режиме жесткого реального времени.

В результате применения инструмента Component Integrator значительно (нередко в несколько раз) снизились требования к аппаратуре, а также еще более повысились надежность и производительность. Впоследствии технология Component Integrator была куплена корпорацией Microsoft, и под именем Component Designer стала основой нового бизнес-направления Microsoft: производство и поставка встраиваемых ОС, линейки продуктов Windows Embedded. Что касается RTX, то этот продукт позволил себе и вовсе доселе немислимое: использовать стандартные настольные ОС Microsoft, в том числе и для приложений реального времени.

Нужно внести ясность относительно названия компании-производителя. RTX — детище компании VenturCom. В 2004 г. VenturCom сменила название на Ardence, затем в 2006 г. Ardence была куплена корпорацией Citrix, наконец, в 2008 г. на основе подразделения корпорации Citrix, занимавшегося ПО реального времени, была создана отдельная компания, получившая название IntervalZero Inc. Но, несмотря на все

эти пертурбации, сегодня решения серии RTX развивает и поддерживает та же команда, что и раньше, поскольку костяк технических специалистов и менеджеров компании VenturCom перешел во вновь образованную компанию IntervalZero.

За 10 с лишним лет существования RTX удобство и эффективность этого продукта смогли оценить очень многие разработчики, реализовавшие на его основе сотни тысяч проектов. И все это время RTX успешно развивался, обрстая новыми функциями и возможностями. Посмотрим, что представляет собой современный RTX с технической точки зрения.

## RTX: ВЗГЛЯД ИЗНУТРИ

Продукт RTX состоит из двух частей. Первая часть — пакет разработчика RTX SDK (Software Development Kit) — предназначен для создания собственных приложений для работы в среде RTX. Вторая часть — подсистема исполнения RTX Runtime — представляет собой непосредственно подсистему реального времени для обслуживания приложений RTX. Подсистема RTX Runtime устанавливается на целевые системы, где предполагается запуск RTX-приложений. Пакет разработчика также включает подсистему реального времени, поэтому разрабатываемые приложения RTX можно запускать и отлаживать непосредственно на локальном месте разработчика. Кроме того, очень ценным свойством SDK является его тесная интеграция со средой разработки Visual Studio от компании Microsoft.

На момент написания статьи последней версией RTX являлась RTX 2011. С версии RTX 2009 нумерация несколько изменилась. Ранее версии имели номера (вплоть до 8.1.1), теперь вместо номеров используются годы выпуска. Компания IntervalZero обеспечивает 10-летний срок поддержки своих про-

дуктов, поэтому приобрести системы исполнения к ранее купленной версии не составляет труда. Самая ранняя, поддерживаемая на данный момент версия, — RTX 7.1.

Каким образом на основе Windows можно создать систему реального времени? Стоит сразу отметить, что все предлагаемые на рынке решения не меняют функциональности и возможностей самой Windows, в ее поведении и работе ничего не меняется. Подсистема, реализующая функции системы реального времени, работает «рядом» с Windows. Обычные задачи Windows выполняются в недетерминированной среде, как и раньше. Один из вариантов реализации подобного механизма — разделение ресурсов между Windows и подсистемой реального времени. Подобный подход реализовала, например, компания TenAsys, предлагающая продукт INTime. При использовании INTime на одной аппаратной платформе запускаются две виртуальные машины — с Windows и с подсистемой реального времени INTime.

Здесь в целом неважно, какая ОС выполняется параллельно с подсистемой реального времени, поддержка Windows определяется наличием модифицированного уровня аппаратных абстракций HAL для работы в таком режиме и наличием интерфейсов API для связи задач, работающих в режиме реального времени и в Windows-среде. С популяризацией многоядерных архитектур стало возможным выделение отдельных ядер под Windows и под систему реального времени. Однако довольно сложная архитектура INTime не позволяет добиться высокой производительности в задачах реального времени.

В отличие от описанного подхода RTX тесно интегрируется с ОС Windows.

На рисунке 1 приведена его архитектура. RTX дополняет стандартный HAL Windows своим расширением (RTX HAL Extension). На этом уровне кроме организации доступа к аппаратуре обрабатываются прерывания от таймера подсистемы реального времени. Непосредственно функциональность реального времени реализует слой RTSS (Real-Time SubSystem). Это ядро всей подсистемы реального времени. Здесь находится свой планировщик, который оперирует выполнением задач реального времени и предоставлением ресурсов задачам Windows-среды. Фактически, любая задача RTSS имеет более высокий приоритет, нежели любая задача, выполняющаяся в Windows. Кроме того, этот слой полностью реализует API реального времени (Real-Time API — RTAPI), на основе которого создаются приложения подсистемы RTSS.

Приложения реального времени (RTSS Process) выполняются здесь на уровне ядра Windows, имеют те же привилегии и ограничения, что и драйверы устройств. Этот подход отличается от других реализаций (например, INTime обеспечивает приложениям реального времени пользовательский режим, собственное адресное пространство и изоляцию). Но за счет этого достигается очень быстрое переключение контекста задач.

Доступ к функциям RTAPI возможен как для процессов RTSS, так и для обычных приложений Win32. Это позволяет выделять в Windows-задачах отдельные, критичные ко времени выполнения, участки. Такой возможности лишены системы, где Windows и подсистема реального времени работают параллельно. Кроме того, расширение интерфейса Win32 API функциями RTSS позволяет приложениям Win32 и приложениям RTSS обмениваться данными.

Как можно видеть из рисунка 1, в составе RTX присутствуют компоненты, организующие для приложений RTSS сеть реального времени и работу в реальном времени с устройствами USB. Это стек протоколов RT-TCP/IP и стек RTX USB. Сетевой стек обеспечивает работу всех сетевых протоколов и физического уровня в детерминированном режиме. Аналогично работает поддержка USB — все компоненты работают с полностью предсказуемыми задержками.

Наличие стека реального времени означает работу всех функций и драйверов с гарантированным временем, т.е. разработчик может быть уверен, что входящий пакет будет обработан, и исходящие данные будут доставлены до физического уровня за ограниченный промежуток времени. При этом наличие стека RT-TCP/IP не означает, что вся сеть будет работать в реальном времени, даже если данный стек будет стоять на всех узлах сети. Сеть состоит из многих элементов, в частности, в ней есть оборудование, которое реализует ее топологию. На физическом уровне сети могут происходить различные процессы, дополнительное оборудование может вносить свои задержки, кроме того, ОС Windows тоже не бездействует в плане сетевой активности. Поэтому сетевой стек реального времени позволяет получить гарантированную производительность, но не гарантирует работу всей сети без сбоев и задержек. Стек RT-TCP/IP предназначен только для достижения максимума производительности в сетевых задачах. Наиболее приближенную к реальному времени работу сети можно получить, если выделить все сетевые адаптеры для RTX в один сетевой сегмент.

### ВЫГОДА ОТ ПРИМЕНЕНИЯ RTX

Что RTX может дать разработчику? Во-первых, использование стандартной ОС Windows позволяет обойтись без дополнительного аппаратного обеспечения и специализированных систем реального времени. Весь проект от начала до конца может быть реализован специалистами, ранее разрабатывавшими только обычные Windows-приложения, поскольку API RTX максимально приближен к API Win32 и не требует много времени на освоение. Благодаря огромному сходству между этими двумя API разработчику, никогда не имевшим дело с программированием систем реального времени, не составит большого труда научиться создавать приложения реального времени под RTX. Этому весьма поспособствует и интеграция со знакомой системой разработки Microsoft Visual Studio. Все это в совокупности позволит значительно (очень часто в разы) сократить затраты

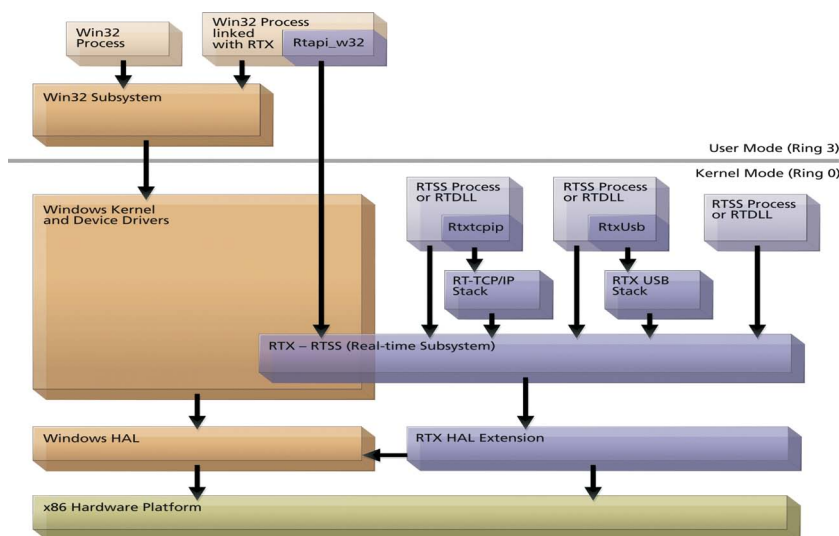


Рис. 1. Устройство подсистемы RTX

на развертывание проекта, поскольку стоимость владения и применения RTX сопоставима (а в большинстве случаев ниже) со стоимостью традиционных ОСРВ.

Во-вторых, и это очень важно, применение RTX не является неким компромиссом или ограничением возможностей. Разработчик получает современную, мощную, быструю и гибкую подсистему реального времени, не уступающую по функциональности классическим ОСРВ. Абсолютно все процессы в подсистеме RTSS детерминированы, а это является основой любой системы реального времени.

Кроме того, несмотря на тесную интеграцию с операционной системой, RTX имеет такое важное свойство как устойчивость к сбоям среды Windows. Подсистема RTSS умеет корректно обрабатывать завершения Windows, как штатные, так и аварийные («синий экран»). Также в RTX имеется механизм, с помощью которого можно сделать доступной память, которую «не видит» Windows. Этой цели служат, в частности, функции PAE (Physical Address Extensions), расширяющие адресное пространство до 36 разрядов и позволяющие получать на 32-разрядных системах доступ к 64 Гбайт памяти.

Еще одной очень важной особенностью RTX является организация связи между задачами Win32 и RTSS. Механизмы взаимодействия процессов (IPC) являются привычные события, мьютексы, семафоры и общая память. При этом взаимодействие между детерминированной подсистемой RTSS и недетерминированной подсистемой Windows организовано так, чтобы исключить их перекрестное влияние на работу друг друга. Еще одним механизмом, направленным на обеспечение безопасности взаимодействия подсистем, является механизм прокси-процессов, осуществляющий обращение из среды Windows к RTSS-окружению.

Однако самым важным свойством, которое появилось с версии RTX 2009, является поддержка для задач RTSS работы в режиме SMP — симметричной мультипроцессности. В версии RTX 2011 было отменено деление продуктов по признаку наличия либо отсутствия поддержки SMP. Теперь при разработке программ с помощью SDK всегда есть возможность использовать SMP, а Runtime отличаются только числом ядер, которые можно выделить под RTX (от 1 до 31). В сочетании с различным оборудованием запуск RTX возможен в различных режимах.

### РЕЖИМ БЕЗ МУЛЬТИПРОЦЕССНОСТИ

Данный вариант поддерживали предыдущие версии без SMP. Это един-

ственный вариант работы для RTX Runtime Solo с поддержкой выделения одного ядра. RTX и Windows выполняются здесь на одно- или многопроцессорной (многоядерной) системе. Возможны два варианта:

– разделяемый режим:

в этом случае RTX и Windows работают на одном ядре процессора, разделяя его ресурсы. Соответственно и возможности такой системы довольно ограничены, особенно при большой загрузке. Такой вариант применим для систем с не очень большими требованиями к подсистеме реального времени. Если загрузка довольно серьезная, то лучше применить другой режим;

– режим с выделенным процессором:

в этом случае RTX полностью занимает одно процессорное ядро. Естественно, для этого необходимо, чтобы в системе было более одного ядра (или процессора). На остальных ядрах выполняется Windows. В такой системе RTX уже не требуется делить вычислительные ресурсы с Windows, потому здесь возможна реализация более серьезных систем реального времени. В современных условиях, когда широко распространены процессоры с двумя и более ядрами, выделение отдельного ядра под RTX не влечет за собой каких-либо серьезных затрат.

### РЕЖИМ С ПОДДЕРЖКОЙ SMP

Этот режим предполагает наличие нескольких ядер или процессоров. Режим SMP поддерживает выделение под задачи RTX от 2 до 31 процессорного ядра, которые используются по возможности с полностью симметричной загрузкой. На оставшихся ядрах будет выполняться Windows. Реализация

такого режима позволяет строить системы реального времени, которые крайне требовательны к производительности и вычислительным мощностям и занимаются очень мощными вычислениями.

Наличие различных версий SDK и Runtime позволяет управлять своими вложениями в систему посредством выбора нужного числа поддерживаемых ядер. Чем больше в системе ядер, поддерживаемых RTX, тем дороже конечное решение. Все Runtime, даже с поддержкой выделения 31 ядра, сохраняют поддержку работы на одно- или двухядерных системах, где нельзя реализовать режим SMP для RTSS-задач. Число используемых подсистемой RTSS ядер задается в настройках и ограничено лишь тем максимумом, который поддерживает установленный в систему Runtime.

### ДРУЖЕСТВЕННОСТЬ К ПОЛЬЗОВАТЕЛЮ

Несмотря на сложность архитектуры RTX, для пользователя общение с RTX является простым и прозрачным. Например, свойства подсистемы RTX задаются из одного окна (см. рис. 2). К тому же в составе RTX SDK имеется обширная документация по работе с RTX (см. рис. 3). Программисты, знакомые с Microsoft Visual Studio, имеют возможность пользоваться данным инструментарием для разработки задач RTX. В среду разработки добавляются шаблоны возможных вариантов приложений RTX (см. рис. 4). Стандартные отладочные механизмы Microsoft Visual Studio доступны при этом для отладки и приложений RTX. По сути, вся разработка ведется привычными методами и с помощью известных инструментов.

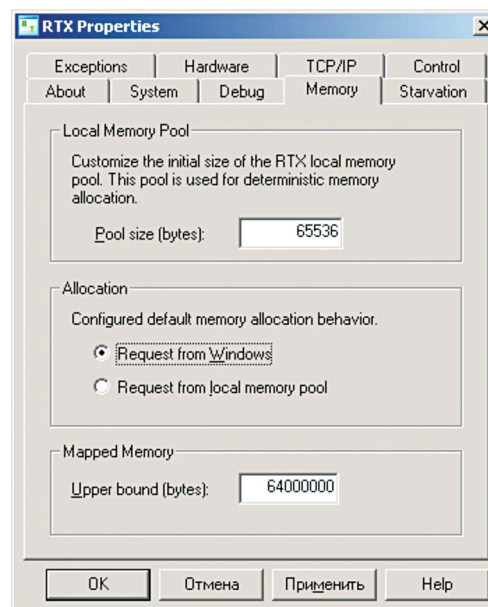


Рис. 2. Окно настроек RTX



Для знакомства с RTX доступна оценочная (Evaluation) версия RTX SDK, работающая 30 дней. В ней имеется практически все, что есть в коммерческой версии SDK, включая документацию, поддержку интеграции с Microsoft Visual Studio, средства оценки производительности системы, тестовые примеры, а также максимальный RTX Runtime с поддержкой выделения 31 процессорного ядра.

### РАЗУМНЫЙ ВЫБОР

Сфера влияния RTX весьма обширна, известно около миллиона различных его применений. В промышленности, медицине и телекоме пользователями RTX являются такие компании как Siemens, ABB, Quanser, National Instruments, VMIC, General Electric. В аэрокосмической отрасли RTX используется компаниями Raytheon, Boeing, Honeywell. Автомобильная индустрия также не обошла этот продукт вниманием: реализованные RTX-проекты есть у Porsche, Renault, Peugeot-Citroen

и у других автопроизводителей. Примером из промышленного сектора может служить проект Siemens по переводу своей Windows-платформы для контроллеров на базу RTX. В военной сфере RTX использовался, в частности, компанией Raytheon в системах обеспечения станций по тестированию ракет Patriot.

Все это происходит потому, что стремление использовать недорогие, хорошо знакомые ОС общего назначения во встраиваемых системах и приложениях реального времени уже неистребимо. Задумайтесь: так ли уж нужны все эти OCPB, если пакет Visual Studio уже есть, недостатка людей, умеющих им пользоваться, не предвидится, а расширение RTX стоит сущие копейки? Если вы создаете бездисковое решение реального времени, достаточно использовать RTX в комбинации с ОС серии Windows Embedded — это будет дешевле, быстрее и проще, чем если использовать любую традиционную OCPB. Если же бездисковость не требуется, то это еще лучше, поскольку тогда

можно ограничиться комбинацией из RTX и самой обычной недорогой ОС настольного класса — той самой, что работает в вашем персональном компьютере.

Правильная стратегия выбора ОС должна выглядеть следующим образом: прежде, чем покупать OCPB, удостоверьтесь в том, что вы не можете решить вашу задачу с использованием обычной стандартной ОС. Если никакая из разновидностей классических ОС Windows и Linux вам не подходит, подумайте над тем, как интегрировать в компьютер, работающий под управлением Windows/Linux, подсистему реального времени. И лишь в самую последнюю очередь, окончательно убедившись, что ни первый, ни второй варианты вам не подходят, подумайте о том, как решить свою задачу при помощи OCPB. OCPB — это крайнее средство, к которому следует прибегать лишь тогда, когда все иные пути отрезаны.

### ОТВЕТЫ НА ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ ОБ RTX

**Чем отличается расширение RTX от других аналогичных продуктов, например, от INTime компании TenAsys?**

При использовании INTime на одной аппаратной платформе работают две разных ОС: Windows и дополнительная подсистема реального времени, которая представляет собой, по сути, самостоятельную ОС с уровнями безопасности, разделением памяти, распределением процессорного времени и т.д. Современные многоядерные процессоры позволяют реализовать данный подход с использованием как INTime, так и какой-нибудь классической системы реального времени вроде QNX или VxWorks. Расширение же RTX встраивается в Windows очень плотно, становится в полном смысле слова частью ОС и, работая полностью на уровне драйверов, обеспечивает очень быстрое переключение задач. К тому же, приложение для RTX можно писать в стандартном пакете Microsoft Visual Studio. Если совсем коротко, то основные преимущества RTX по сравнению с INTime — это быстродействие, компактность и удобство.

**Облегчает ли расширение RTX создание детерминированных сетевых приложений?**

Да, конечно. Это одна из традиционно сильных сторон RTX. Комплект поставки RTX включает детерминированные стеки и полный набор других компонентов, необходимых для обеспечения работы сетей Ethernet в реальном масштабе времени.

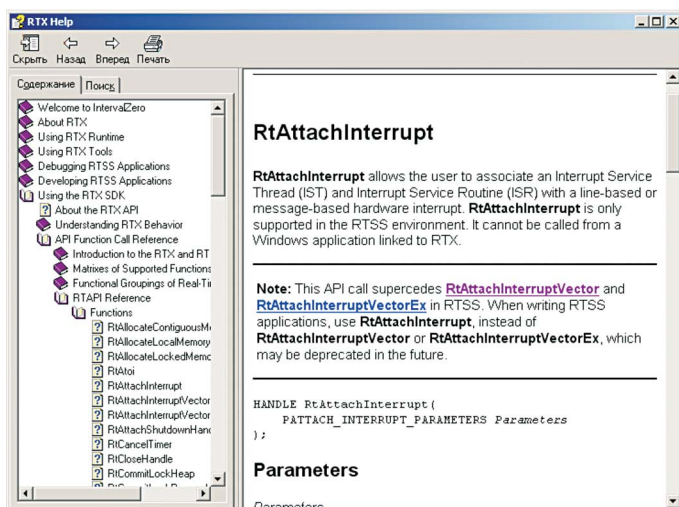


Рис. 3. Справка по RTX

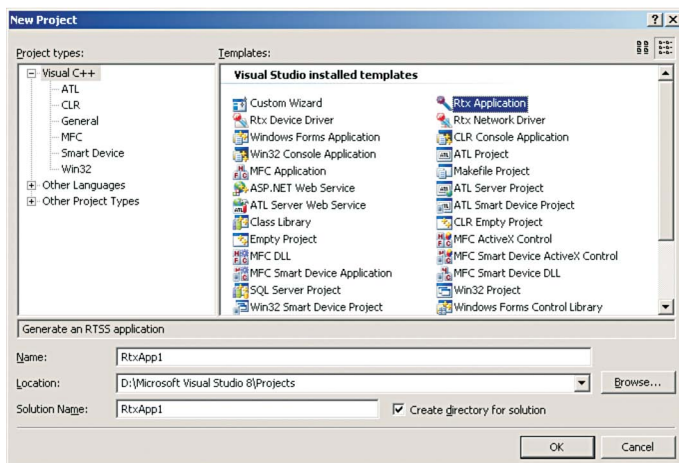
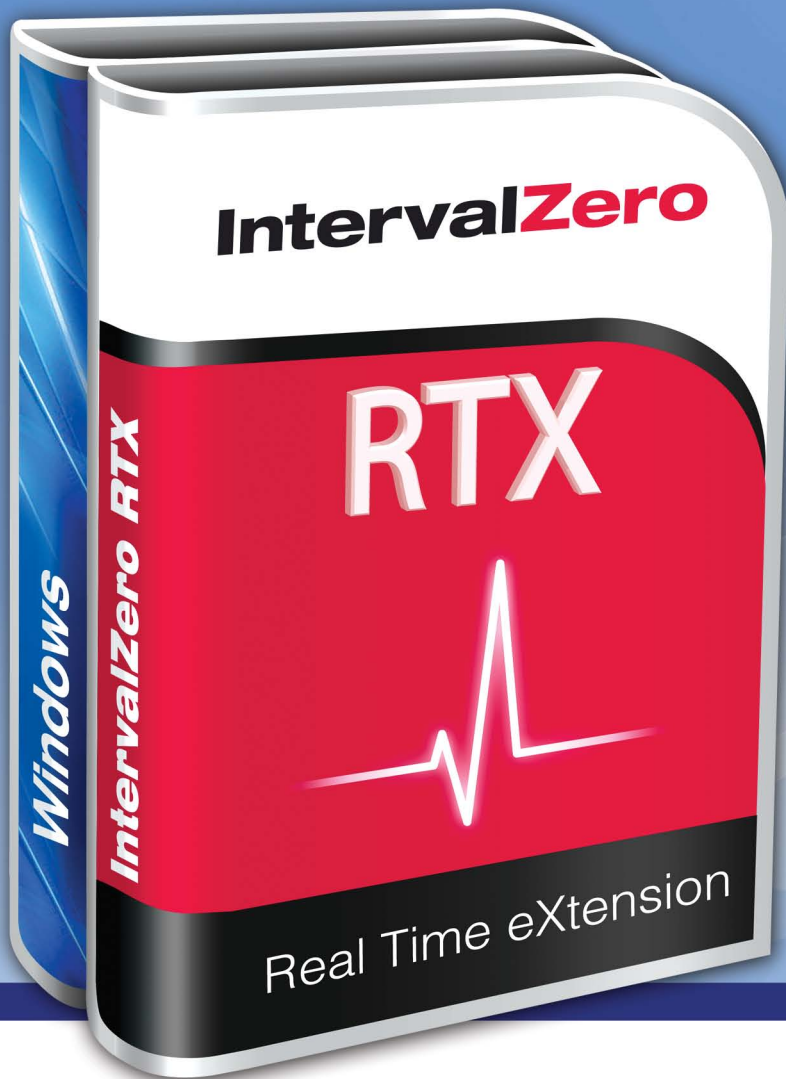


Рис. 4. Шаблоны RTX-приложений в среде Microsoft Visual Studio



## Преврати Windows в систему реального времени

**IntervalZero RTX - самое доступное и бюджетное средство для добавления в ОС Windows функционала операционной системы жесткого реального времени. RTX дополняет и интегрируется с Windows, не изменяя ее модулей или поведения, и позволяет получить подсистему реального времени, работающую совместно с Windows без применения дополнительного аппаратного обеспечения и специальных ОСРВ.**

- Полный функционал ОСРВ
- Легкое в освоении решение, позволяющее стартовать проекту в кратчайшее время
- RTX работает даже при «синем экране смерти» Windows
- Симметричное паралельное выполнение задач реального времени
- Поддержка многопроцессорных и многоядерных систем
- Совместимо с Microsoft Windows 7, Vista, 2000, XP Embedded, Embedded Standard 7, Server
- Поддержка пакета Microsoft Visual Studio для полноценной разработки
- Независимый от Windows стек TCP/IP реального времени, совместимый с WinSock
- Очень простая прямая работа с аппаратурой – не сложнее, чем в DOS

**IntervalZero**



**RTSoft**  
средства и системы автоматизации

Москва, Никитинская 3, ЗАО «РТСофт»  
Тел.: (495) 976-1505  
Факс: (495) 742-6829  
E-mail: [rtsoft@rtsoft.ru](mailto:rtsoft@rtsoft.ru); [www.rtsoft.ru](http://www.rtsoft.ru)

**Получается, что расширение RTX позволяет отказаться от традиционных ОСРВ?**

Да, во многих случаях это именно так. Если задача в принципе может быть успешно решена без использования ОСРВ, то ее использовать не нужно. Расширение RTX уже позволило реализовать сотни тысяч проектов реального времени на базе самых обычных, не встраиваемых версий ОС Windows, которые значительно дешевле традиционных систем реального времени. Справедливости ради надо признать, что заменить классические профессиональные системы реального времени в любых приложениях комбинация Windows + RTX не сможет. Однако в силу развития Windows, продукта RTX и современных x86-платформ, количество приложений, где возможности Windows + RTX позволяют эффективно решать актуальные задачи, постоянно растет.

**В каких прикладных областях используется расширение RTX в России?**

Российские разработчики давно и успешно применяют расширение RTX в авиационных симуляторах, в контрольно-диагностических комплексах, при проведении научных экспериментов и во многих других приложениях, где, во-первых, требуется реальное время и, во-вторых, допустимо использование стандартных или встраиваемых версий Windows.

**Какие дополнительные аппаратные ресурсы необходимы для полноценной работы расширения RTX?**

Расширение RTX очень нетребовательно к ресурсам, и в любой реальной системе — настольной рабочей станции, пульте управления, мобильном устройстве либо промышленном контроллере — большая часть ресурсов будет затрачена на саму Windows, настольную либо встраиваемую.

**Говорят, что при использовании расширения RTX пользователь имеет прямой доступ к оборудованию как в реальном режиме MS-DOS?**

Да, это так. Простота работы с оборудованием является одним из основных факторов популярности расширения RTX у Windows-разработчиков.

**Нужно ли покупать встраиваемую версию Windows (Windows XP Embedded или Windows Embedded Standard) для работы с RTX?**

Нет, продукт RTX прекрасно работает с обычными «настольными» версиями Windows из коробки. Но использование встраиваемых версий Windows позволяет иметь больше возможностей

по созданию целевых систем с различной конфигурацией.

**Нужно ли приобретать для расширения RTX какие-то специальные средства разработки?**

Нет, ничего специального для разработки RTX-приложений приобретать не нужно, достаточно стандартного пакета Microsoft Visual Studio, который имеется у большинства разработчиков.

**Сколько стоит RTX?**

Расширение RTX стоит меньше классических систем реального времени: 350 долл. для одной уникальной системы, при этом НДС, согласно действующему российскому законодательству, не взимается.

**Насколько снижается стоимость расширения RTX с ростом числа установок?**

С версии RTX-2011 цены на единичные установки были снижены и приравнены к тем, которые раньше действовали лишь при покупке десяти и более лицензий. Дальнейшее снижение цен могут получить уже крупные производители оборудования с сотнями установок. Расширение RTX — это недорогой продукт, и теперь он стал еще более доступен для небольших проектов.

**Сложно ли написать драйвер устройства под систему с расширением RTX?**

Во многих случаях отсутствует необходимость в отдельном драйвере устройства для задач RTX, т.к. приложение может довольно просто работать с устройством напрямую. Но если драйвер нужен, написать его под RTX-систему в каком-то смысле даже проще, чем под стандартную ОС Windows. Если вы не располагаете собственными инженерными ресурсами для написания драйверов, можно обратиться в компанию «РТСофт» ([www.rtsoft.ru](http://www.rtsoft.ru)), специалисты которой имеют большой опыт создания RTX-драйверов для самого разного, в т.ч. уникального оборудования.

**Насколько сложен процесс интеграции RTX в различные версии Windows?**

Продукт RTX создавался для того, чтобы было легко и удобно работать, поэтому процесс интеграции этого расширения с базовыми ОС хорошо продуман и максимально облегчен. Применительно к настольным версиям Windows дело обстоит проще некуда: потребуется не более двух щелчков мышью. И в случае Windows Embedded интеграция происходит не намного сложнее.

**Где можно получить специальные знания для эффективной работы с расширением RTX?**

Весь нужный объем специальных знаний можно получить, например, в учебном центре компании «РТСофт» ([www.rtsoft.ru](http://www.rtsoft.ru)). Но, как показывает практика, большой необходимости в этом нет. Любой специалист, способный писать приложения для Win32 API, разберется с RTX буквально за пару дней, ведь к его услугам будет полноценная документация, описывающая все аспекты работы RTX и включающая руководство по программированию. Другими словами, при использовании расширения RTX инвестировать дополнительные средства в обучение разработчиков можно, но не обязательно. Это одно из фундаментальных отличий RTX от традиционных систем реального времени, для эффективного использования которых необходимы весьма квалифицированные специалисты с соответствующим уровнем зарплаты.

**Насколько быстро работает комбинация Windows + RTX по сравнению с «чистой» ОС Windows?**

Прежде всего, по сравнению с ОС Windows комбинация Windows + RTX в любом состоянии имеет предсказуемое время отклика для задач реального времени. При этом RTX работает с интервалом до 1 мкс, в то время как Windows оперирует в лучшем случае 1 мс. Говорить о времени реакции самой ОС Windows нельзя, поскольку она не является ОСРВ и не характеризуется в силу этого детерминированным поведением. Можно сравнить время реакции на одно и то же событие в Windows и в RTX на одной аппаратной платформе. Например, даже на такой платформе как Intel Atom, расширение RTX обеспечивает время реакции на срабатывание таймера от 1 до 11 мкс, а Windows от 600 до 1500 мкс.

**Влияет ли RTX на характеристики и поведение Windows?**

Нет, RTX интегрируется с Windows путем добавления компонентов, не изменяя ничего в самой ОС. Система остается в исходном виде, с такой же полноценной функциональностью и работает точно так же, как и без RTX. RTX не вносит никаких дополнительных рисков в работу Windows и ее приложений.

**Много ли места занимает расширение RTX?**

Нет. В том, что касается потребления памяти и дискового пространства, расширение RTX отличается предельной скромностью: RTX в целом образе может занимать не более 4 Мбайт. Идет ли речь о персональном



компьютере или же о каком-нибудь компактном устройстве, докупать оперативную память, флэш-память и/или дисковые накопители для RTX точно не придется. Общий принцип таков: расширение RTX легко адаптируется к любой аппаратной конфигурации, где работает стандартная Windows. Львиная доля потребляемых ресурсов всегда приходится именно на Windows, вклад же RTX пренебрежимо мал.

**Есть ли какие-либо особенности у аппаратных платформ для расширения RTX?**

Никаких! Там, где работает Windows — будь то решение типа PC Box, мобильный компьютер, система VME или CompactPCI, настольная персональная система, сервер или

что-то еще — сможет работать и расширение RTX. Единственное жесткое аппаратное ограничение со стороны RTX — это наличие x86-совместимого центрального процессора, но данное требование выполняется автоматически, поскольку стандартные версии ОС Windows рассчитаны только на архитектуру x86, а все версии Windows Embedded для этой архитектуры доступны. Продукт RTX совместим с любыми x86-процессорами, от самых современных до самых устаревших.

**Можно ли использовать расширение RTX как дополнительное средство, повышающее надежность системы?**

Да, RTX можно использовать в таком качестве, поскольку во многих случа-

ях после того, как Windows зависает или даже появляется «синий экран», подсистема RTX продолжает работать и, как минимум, способна обеспечить мягкую безболезненную перезагрузку системы.

**Можно ли использовать расширение RTX в бездисковых конфигурациях?**

Не только можно, но и очень полезно. Расширение RTX может работать как со стандартной Windows, так и со встраиваемыми ОС Windows Embedded. При использовании с Windows Embedded расширение RTX способно улучшить качество конечного изделия и обеспечить дополнительные преимущества за счет повышения надежности и поддержки реального времени.

## ИНЖЕНЕР — ИНЖЕНЕРУ

**| ПОЧЕМУ ШУМИТ СХЕМА? |** Одна из главных причин шума — пульсации на шинах питания и земли от импульсного источника питания. Рецепты уменьшения таких шумов хорошо известны. Это — использование параллельно включенных электролитических и керамических фильтрующих конденсаторов, устанавливаемых в непосредственной близости от выводов разъема с шинами питания и установка байпасных керамических конденсаторов как можно ближе к выводам питания микросхемы с высокочастотными сигналами. Не стоит экономить на этих конденсаторах, лучше выбирать их с минимальными ESR и ESL, иначе пользы от них будет немного, особенно, если частота импульсного источника питания или преобразователя составляет сотни килогерц.

Иногда рекомендуют RC-фильтр по питанию: в разрез шины питания включают небольшой резистор (0,1...2 Ом), а после него — емкость на землю. Этот метод неплохо работает с АЦП, ЦАП и компараторами с внутренними ИОН, но вряд ли пригоден для операционного усилителя (ОУ).

Но иногда встречаются довольно экзотические источники шумов. Автору известен случай, когда были приняты все возможные меры, но схема продолжала шуметь — величина шума изменялась непредсказуемым образом, иногда шум и вовсе исчезал. Высказывались разные предположения, в том числе предполагались наводки по эфиру — в схеме стояли электрометрические ОУ с очень большими входными сопротивлениями. Когда при наладке опытных экземпляров перерезали дорожки на плате и поставили прямые перемычки из экранированных проводов, шум исчез.

Не будем описывать подробности схемы и всю последовательность действий — ларчик открылся просто и неожиданным образом. На входе ОУ стояли диоды в остеклованном корпусе, и свет от солнца и люминесцентных ламп, попадая на них, очевидно, вызывал фотоэффект — строго говоря, p-n-переход можно рассматривать как фотодетектор. Во всяком случае после замены диодов шум исчез.

**АКТИВНЫЙ ОДНОПУЛПЕРИОДНЫЙ ВЫПРЯМИТЕЛЬ С МАЛЫМ ПАДЕНИЕМ НАПРЯЖЕНИЯ** Суть предлагаемой схемы заключается в уменьшении падения напряжения на выпрямителе. Схема интересна еще и тем, что на ее основе можно построить ряд других решений, а не только те, какие имел в виду автор идеи.

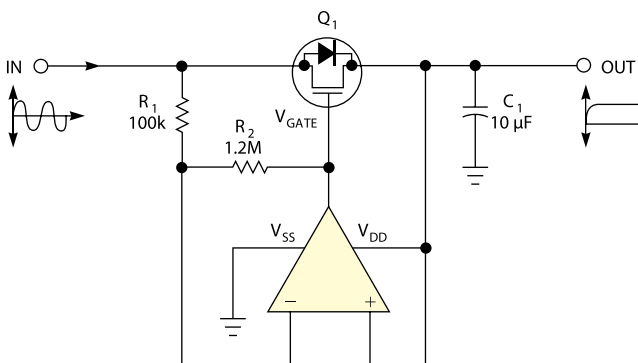


Рис. 1. Схема активного выпрямителя

Если взять, например,  $(R2/R1) = 12$ , то падение на канале открытого ключа составит примерно 40 мВ. Можно увеличить и усиление ОУ, но в этом случае следует учитывать погрешность, вызванную входным смещением напряжения. Учитывая, что напряжение питания подается на неинвертирующий вывод, а выходное напряжение должно быть достаточным для управления затвором и при низком напряжении питания, ОУ должен быть rail-to-rail по входу и выходу. Вместо источника переменного входного напряжения могут использоваться аккумуляторные или солнечные батареи.

Принцип действия выпрямителя очевиден из рисунка 1. Схема с самозапуском — операционный усилитель (ОУ) питается от выпрямленного напряжения. В принципе можно выбрать ОУ с током потребления меньшим, чем обратный ток диода Шоттки, но в этом случае полоса пропускания ОУ окажется крайне малой — не более 400–600 Гц, соответственно и частота входного напряжения не должна превышать эту величину, а лучше, чтобы она была в несколько раз меньше.

Выходное напряжение ОУ ( $V_{GATE}$ ) определяется соотношением:

$$V_{GATE} = V_{OUT} - (R2/R1)(V_{IN} - V_{OUT}).$$

Напряжения сток-исток  $V_{DS}$  и затвор-исток  $V_{GS}$  определяются из уравнений:

$$V_{DS} = V_{IN} - V_{OUT} \text{ и } V_{GS} = V_{GATE} - V_{OUT} \text{ соответственно.}$$

Отсюда получаем:

$$V_{GS} = -(R2/R1)V_{DS}.$$