

# Применение механизма Persistent Publish/Subscribe во встраиваемых промышленных приложениях

Бен Вандельбельт, инженер компании QNX Software Systems

**В** статье рассматриваются возможности применения механизма QNX Persistent Publish/Subscribe для создания высоконадежных масштабируемых встраиваемых систем, которые способны поддерживать множество устройств и программных компонентов и обеспечивать их взаимодействие со сложным графическим пользовательским интерфейсом.

Применение механизма QNX Persistent Publish/Subscribe (PPS) показано на примере системы управления энергопотреблением QNX Smart Energy. Человеко-машинный интерфейс данной системы построен на основе технологий Adobe Flash, однако, мы полагаем, что механизм Persistent Publish/Subscribe может успешно использоваться при построении интерфейсов на основе OpenGL и других технологий.

При создании встраиваемых систем для рынка промышленной автоматизации нередко возникают задачи, которые невозможно решить за счет простого увеличения объема памяти и использования более мощного процессора. Встраиваемые промышленные приложения традиционно отличаются сложностью архитектуры. Они разрабатываются для решения большого объема задач, объединяют множество устройств и программных компонентов. Экономическая эффективность таких систем может достигаться за счет использования недорогого аппаратного обеспечения (менее мощных процессоров, минимальной памяти и т.д.). По требованию заказчика они могут быть оснащены сложным человеко-машинным интерфейсом (ЧМИ). В последние годы к промышленным системам все чаще предъявляют еще одно требование: масштабируемость, т.е. возможность расширения функциональности оборудования.

Для реализации всех этих требований в новую версию операционной системы реального времени QNX Neutrino была интегрирована новая служба: Persistent Publish/Subscribe (PPS). Это объектно-ориентированная служба, в которой серверы публикаций (publisher) и подписчики (subscriber) взаимодействуют в рамках слабосвязанной архитектуры. Компактная и расширяемая, служба PPS позволяет создавать недорогие, надежные, масштабируемые системы на базе асинхронной модели обработки сообщений.

Служба PPS была заложена в основу системы обработки сообщений отладочной платформы QNX Smart Energy (см. рис. 1), обеспечивающей интеллектуальное управление системами энергопотребления зданий. QNX Smart Energy объединяет многочисленные датчики, контроллеры, счетчики и другое оборудование под управлением единого ЧМИ на основе flash-графики.

Достигнутая прозрачность и масштабируемость системы (как на программном, так и на аппаратном уровне) позволяют сделать вывод о том, что использованная в QNX Smart Energy асинхронная модель обработки сообщений подходит для сложных промышленных систем лучше, чем более распространенная сегодня синхронная модель.

## СИНХРОННАЯ МОДЕЛЬ ОБРАБОТКИ СООБЩЕНИЙ

Синхронная модель обработки сообщений имеет особое значение для систем реального времени, в которых многие процессы, прежде чем продолжить работу, должны получить ответ на уже отправленные сообщения. Но она не всегда подходит для применения в системах, архитектура которых должна предусматривать возможность добавления новых приложений и функций.

Синхронная обработка сообщений предполагает сильные связи между отправителем и получателем сообщений: каждый сервер напрямую взаимодействует со своими клиентами, поэтому он должен точно знать, как отвечать на все их сообщения. При такой организации обработки сообщений по мере добавления новых или изменения существующих компонентов, сложность целевой системы быстро увеличивается. Она становится более запутанной и хрупкой, а процесс ее разработки и модернизации занимает все больше времени.

## АСИНХРОННАЯ МОДЕЛЬ ОБРАБОТКИ СООБЩЕНИЙ

Асинхронные коммуникации являются решением нижнего уровня (low-level solution). Такие задачи, как управление буферизацией, обработка ошибок и сквозных записей переносятся на уровень приложений. А это значит, что при использовании данной модели необходимо разработать такой протокол, который будет обеспечивать корректность обмена сообщениями между всеми приложениями и, гарантированно, будет предоставлять необходимые им ресурсы памяти. При создании таких протоколов для простых систем от инженеров не требуются особые усилия. Но при проектировании масштабных промышленных систем могут возникнуть часто неразрешимые проблемы.

## PERSISTENT PUBLISH/SUBSCRIBE

Persistent Publish/Subscribe (PPS) — новый сервис операционной системы QNX Neutrino, позволяющий создавать недорогие, надежные, масштабируемые системы на базе асинхронной модели и слабосвязанной архитектуры системы обмена сообщениями.



рис. 1. Экран отладочной платформы QNX Smart Energy

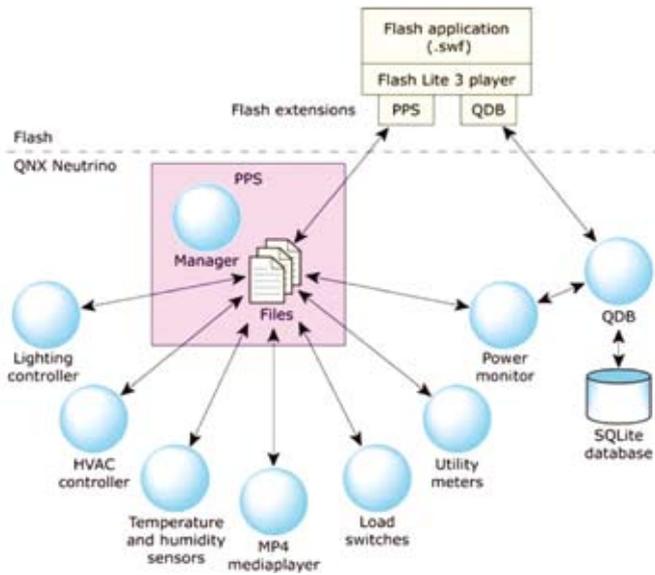


рис. 2. Конфигурация отладочного приложения QNX Smart Energy, использующего PPS-модель обработки сообщений. Обратите внимание на то, что для обеспечения обмена между NMI-интерфейсом и всеми программными компонентами системы требуется всего одно Flash-расширение. Второе Flash-расширение обеспечивает прямое взаимодействие между NMI-интерфейсом и базой данных QDB

### Двоичные или воспринимаемые человеком объекты?

Службу PPS можно организовать так, чтобы она читала либо двоичные объекты, либо объекты в удобной для восприятия человеком форме.

Отладку удобных для восприятия человеком объектов можно производить с использованием командной строки и простых утилит файловой системы. Например, cat для случая подписки и echo для случая публикации, или с помощью простой программы, которая подписывается на объект и распечатывает отладочную информацию, в том числе, сам PPS-объект и его атрибуты.

В общем виде принцип работы системы может быть описан следующим образом:

- серверы публикаций (publisher) модифицируют объекты и их атрибуты и фиксируют изменения в файловой системе;
- когда любой из серверов публикации (publisher) изменяет объект, служба PPS оповещает всех клиентов (subscriber), подписанных на отслеживание изменений этого объекта;
- клиенты могут подписаться на несколько объектов, а объекты могут обладать множеством серверов публикации и подписчиков.

Такой способ обмена сообщениями позволяет серверам публикаций использовать один и тот же объект для передачи своей информации всем его подписчикам. При этом, если клиент является сервером публикаций, он должен знать лишь то, какую информацию необходимо опубликовать и когда. Если он является подписчиком, то необходимо определить информацию об изменении каких атрибутов каких объектов он будет получать.

### Механизмы Push или Pull?

При реализации по умолчанию служба Persistent Publish/Subscribe работает как система публикации типа *push* (вытолкнуть), то есть, серверы публикации «выталкивают» данные в объекты, а подписчики считывают данные при получении уведомления или по собственному усмотрению.

Однако, некоторые данные, например, счетчики пакетов на интерфейсе, изменяются слишком быстро, чтобы процесс их публикации через службу PPS был эффективным, если используется принятый по умолчанию тип *push*.

Поэтому в службе PPS предлагается дополнительная возможность, при которой подписчик может изменить тип системы публикации в службе PPS на *pull* (вытащить). При открытии объекта в таком режиме выдается команда *read()*, и все серверы публикации этого объекта получают уведомление о необходимости записать в объект текущие данные. Подписчики не считывают блоки, пока данные объекта не обновятся, после чего принимаются новые данные.

При использовании механизма *pull* подписчик службы PPS получает данные от сервера публикаций с необходимым ему темпом — фактически, это есть публикация по запросу.

Все остальные операции, не связанные с обработкой POSIX-вызовы *open()*, *read()* и *write()*, выполняются службой PPS.

Аналогично этому, поскольку PPS-подписчики используют вызовы *read()* для получения объектов, то нет необходимости управлять буферами для этих объектов. Теоретически, подписчик мог бы использовать однобайтовый буфер и просто продолжать чтение, пока не будет прочитано все, что нужно.

### СОХРАНЯЕМОСТЬ ДАННЫХ

Служба Persistent Publish/Subscribe сохраняет данные при перезагрузках. Во время работы она сохраняет объекты в оперативной памяти, а также, по требованию, в постоянном хранилище. Служба восстанавливает объект немедленно при запуске, либо при первом запросе на доступ (отложенная загрузка). Естественно, базовое постоянное хранилище находится в определенной зависимости от надежности файловой системы, будь то жесткий диск, NAND-Flash, NOR-Flash, либо иная специализированная файловая система.

Наряду с обеспечением гарантированной сохранности данных во время перезагрузок, служба Persistent Publish/Subscribe может ускорять запуск системы. Например, обычно, если клиент запускается после запуска сервера, он должен запросить с сервера последние данные, на случай, если между запуском сервера и клиента что-то изменялось. Это требование остается справедливым, если клиент теряет связь с сервером.

При использовании PPS-модели, служба восстанавливает объекты при запуске системы, и сохраняет их по мере того, как они изменяются. Любой клиент — независимо от того запускается он или подключается повторно — должен всего лишь прочитать эти объекты, чтобы получить актуальные данные.

### МАСШТАБИРУЕМОСТЬ СИСТЕМЫ

При работе службы Persistent Publish/Subscribe сервер публикаций и подписчик не знают друг друга. Единственное средство их связи — объект, являющийся целью и результатом. Такая модель дает большую гибкость при построении систем: появляется возможность задерживать решения по точкам соединения модуля и по потокам данных до этапа исполнения программы. Поскольку такие решения не являются жестко закодированными, не имеют непосредственных связей, то их можно адаптировать по мере эволюции требований или ситуаций. Их можно даже динамически изменять во время работы системы.

Слабосвязанная модель обмена сообщениями, на основе механизма Persistent Publish/Subscribe, упрощает также интеграцию новых программных компонентов. Поскольку сервер публикаций и подписчик не должны иметь сведений друг о друге, разработчикам, при добавлении новых компонентов, следует лишь определить, *что* эти компоненты должны публиковать, и *какие данные* нужны для публикации другим PPS-клиентам. Не нужна перенастройка API, а, при добавлении компонента, сложность системы не возрастает.

### ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ СЛУЖБЫ QNX PERSISTENT PUBLISH/SUBSCRIBE

Описанные выше технологии были заложены в основу отладочной платформы QNX Smart Energy для систем интеллектуального управления энергопотреблением, работающих в среде ОСРВ QNX Neutrino (рис. 2). Приложение разработано с учетом возможностей его использования в средах с ограниченными вычислительными ресурсами, например, во встраиваемых промышленных приложениях. QNX Smart Energy объединяет множество устройств: датчиков, видеокамер, программных компонентов под управлением единого человеко-машинного интерфейса (ЧМИ) на основе flash-графики. Есть возможность интеграции новых устройств и компонентов, а также проведения многочисленных модернизаций на всем протяжении эксплуатации.

**ADOBE FLASH LITE**

Человеко-машинный интерфейс системы реализован на базе технологии Adobe Flash Lite, позволяющей использовать преимуществами инструментального набора Flash toolbox. Технология Flash Lite предоставляет все необходимые элементы для построения понятных и эффективных интеллектуальных экранов, соответствующих самым взыскательным требованиям разработки, в том числе: компоновку элементов, разбиение на слои и работу с мультимедиа.

**УПРОЩЕННАЯ АРХИТЕКТУРА**

Одним из ключевых преимуществ использования службы Persistent Publish/Subscribe, при построении системы обработки сообщений отладочного приложения QNX Smart Energy, является лаконичность решения. Для обеспечения связи человеко-машинного интерфейса с другими элементами системы необходимо всего два Flash-расширения: для службы PPS и для QDB-сервера базы данных QNX. В то время как при использовании другой модели обмена сообщениями, необходимо писать Flash-расширения для всех программных компонентов.

Аналогичным образом, механизм PPS позволяет добавлять в систему новые компоненты без необходимости написания новых Flash- и QDB-расширений и без пересмотра логики системы. На рисунке внизу показана одна из возможных конфигураций

отладочного приложения для интеллектуального управления энергопотреблением.

В системе, использующей другую модель передачи сообщений, датчики и компоненты были бы сильно связаны друг с другом и с человеко-машинным интерфейсом. Каждому компоненту было бы известно о любом другом компоненте, с которым нужно обмениваться данными — получается архитектура, в которую трудно вносить изменения или что-то добавлять, что сопряжено с рисками и требует больших временных затрат.

При использовании службы Persistent Publish/Subscribe, компонентам не надо даже иметь сведения о существовании друг друга. Например, если необходимо добавить в систему новое устройство/приложение, достаточно единожды определить, какими данными оно будет делиться с другими устройствами, выступая в качестве издателя, и какие данные оно будет от них получать, являясь клиентом.

Другим преимуществом архитектуры обмена сообщениями через службу PPS является упрощение тестирования, потому что добавление новых компонентов не требует пересмотра структуры обмена сообщениями между всеми другими компонентами.

Таким образом, служба Persistent Publish/Subscribe позволяет упростить и ускорить процесс создания сложных промышленных систем с использованием асинхронных коммуникаций. Она упрощает задачу разработки и модернизации систем, в которых требуется объединять разнородные устройства и программные компоненты под управлением сложного ЧМИ; создает условия для параллельной работы дизайнеров и программистов, и, соответственно, для оперативного вывода продукта на рынок.

**Расширения для Flash**

Flash-расширения для QNX PPS и QDB можно бесплатно загрузить с портала Foundry27 сайта QNX по адресу: <http://community.qnx.com>

**НОВОСТИ РЫНКА | КОМПАНИЯ QNX SOFTWARE SYSTEMS ОБЪЯВИЛА ОБ ОФИЦИАЛЬНОМ ВЫХОДЕ НОВОЙ ВЕРСИИ ОСРВ QNX NEUTRINO 6.5**

Компания QNX Software Systems, крупнейший производитель операционных систем, связующего программного обеспечения и инструментов разработки для рынка встраиваемых систем, объявила об официальном выходе ОСРВ QNX Neutrino 6.5.0 и комплекта разработчика QNX Momentics Tool Suite 6.5.0. Эти версии ПО созданы специально для разработчиков встраиваемых систем для автомобильного, промышленного, медицинского, сетевого и оборонного рынков.

В ОСРВ QNX Neutrino 6.5.0 и комплекте разработчика QNX Momentics 6.5.0 обеспечена расширенная поддержка многоядерных процессоров, введена служба Persistent Publish/Subscribe (PPS), усовершенствован набор инструментальных средств разработки, оптимизированы многочисленные процессы, связанные с производительностью системы.

Одно из ключевых нововведений версии 6.5.0 - поддержка симметричной многоядерности (SMP) для процессоров ARM Cortex-A9 и процессорных ядер Freescale Power e500MC. ОСРВ QNX Neutrino 6.5 поддерживает технологию SMP на многоядерных версиях этих процессоров (до 32 ядер), обеспечивая возможности для широкого масштабирования приложений, выполняющих большой объем вычислительных операций.

Эта служба упрощает разработку и техническое обслуживание сложных систем, в которых интегрировано много аппаратных устройств и программных компонентов. PPS позволяет заменять или обновлять системные компоненты без необходимости внесения изменений в человеко-машинный интерфейс (HMI).

В ОСРВ QNX Neutrino 6.5 повышена производительность ядра при более интенсивном использовании памяти и значительном повышении пропускной способности системы на многих платформах.

В комплекте разработчика QNX Momentics Tool Suite предлагается полнофункциональная среда IDE на основе платформы Eclipse с самыми современными инструментами профилирования, которые помогут разработчикам максимально проникнуть в поведение внутренних системных процессов. В версии 6.5 комплекта представлена поддержка платформ Eclipse 3.5.2, Eclipse CDT 6.0 и компилятора GNU (GCC) 4.4.2. В компиляторе предлагается оптимизированная динамическая компоновка, в том числе, связывание по запросу (lazy linking) и GNU-хеширование.

В ОСРВ QNX Neutrino 6.5 обеспечивается поддержка усовершенствованных программируемых контроллеров прерываний (Intel APIC), прерываний, инициируемых сообщениями (message signaled interrupts, Intel MSI) и расширенная поддержка плат архитектуры x86 от компаний Advantech, Intel и Kontron.

[www.qnx.com](http://www.qnx.com)

**НОВОСТИ РЫНКА | QNX БУДЕТ ПОДДЕРЖИВАТЬ ПРОЦЕССОРЫ INTEL ATOM СЕРИИ E6XX**

Компания QNX Software Systems объявила о намерениях обеспечить поддержку новых процессоров Intel Atom серии E6xx. Поддержка в ОСРВ QNX Neutrino процессоров серии E6xx, ранее известной под кодовым названием Tunnel Creek, позволит разработчикам встраиваемых систем для промышленного и медицинского оборудования объединить гибкость систем ввода/вывода и энергоэффективность процессоров Intel Atom E6xx с надежностью ОСРВ QNX Neutrino и ее высокой производительностью.

Поддержка процессоров Intel Atom E6xx со стороны технологий QNX будет включать комплект человеко-машинных интерфейсов (HMI) на базе Adobe Flash Lite, аппаратную поддержку 2D/3D-графики на основе OpenGL ES и технологию быстрой загрузки QNX. При использовании этой технологии отпадает необходимость в BIOS, что уменьшает стоимость систем и существенно улучшает работу функции мгновенного запуска (instant-on). В 2008 году технология быстрой загрузки QNX получила награду Альянса встраиваемых систем компании Intel «За отличное качество и за самое передовое программное обеспечение для процессоров Intel Atom» («Award of Excellence, Most Innovative Software for the Intel Atom Processor»).

Процессоры Intel Atom E6xx обеспечивают новый уровень гибкости систем ввода/вывода благодаря использованию промышленного стандарта PCI-Express в качестве интерфейса между процессором и чипсетом. Процессоры могут работать с Intel Platform Controller Hub EG20T, который объединяет несколько общих блоков обмена данными (например, SATA, USB, SD/SDIO/MMC и Gigabit Ethernet MAC) и включает такие интерфейсы, как CAN, IEEE1588, SPI, I<sup>2</sup>C, UART и GPIO.

[http://www.qnx.com/news/pr\\_4326\\_1.html](http://www.qnx.com/news/pr_4326_1.html)