

Д.В.Пастушенков, ООО «Пролог»

С.В.Золотарев, ЗАО «РТСофт»

CoDeSys SP RTE: SoftPLC + возможности реального времени в среде Windows

Вкратце о CoDeSys

CoDeSys (*Controller Development System*) немецкой компании Smart Software Solutions GmbH – один из наиболее популярных в Европе комплексов программирования приложений реального времени в стандарте МЭК-61131-3. CoDeSys [1] состоит из двух основных частей: системы программирования и системы исполнения. Система программирования включает в себя редакторы для всех МЭК-языков (ST, SFC, IL, LD, FBD). Имеет развитую систему отладки, средства коллективной разработки проектов, встроенную SCADA-систему и много других полезных разработчику инструментов, то есть все, что нужно для быстрой разработки и внедрения приложений реального времени.

Система программирования содержит встроенные компиляторы для самых различных целевых платформ, таких как Motorola 68xxx, Intel 8086, Intel 8051, Infineon SAB80C16x, Tricore, Power PC, StrongARM, Motorola ColdFire.

Система исполнения – это программа, которая выполняется в целевой платформе и управляет сгенерированным системой программирования кодом. Система исполнения может выполняться на целевой платформе без операционной системы (ОС). Функциональные возможности систем исполнения с операционной системой шире, чем у систем исполнения, не использующих ОС. Разработаны системы исполнения под такие операционные системы как WinCE, Windows NT/2000/XP, VxWorks, QNX, RT-OS32 (RTKernel –32), Linux (RTLlinux).

Расширения реального времени для Windows NT и средства SoftPLC

В настоящее время на рынке предлагаются несколько коммерческих пакетов, обеспечивающих расширения реального времени для Windows NT/2000/XP [4]. Наиболее известным для российских программистов является пакет RTX фирмы Venturcom (сейчас Ardence, <http://www.vci.com>)[3]. RTX обеспечивает пользователя средствами и утилитами для построения и выполнения программ реального времени вместе со средствами для измерения и «тонкой» настройки производительности как аппаратных, так и программных средств. RTX глубоко интегрирован в ядро Windows и для обеспечения необходимых функций использует сервис Windows и WIN32 API. Некоторые фирмы-поставщики пакетов класса SoftPLC используют RTX как среду исполнения своих систем. Например, Siemens в своем пакете WinAC или Infoteam в пакете OpenPCS. Достоинством такого подхода является то, что RTX – это универсальное расширение реального времени для Windows NT, применимое для широкого круга приложений. Но с другой стороны, оно само по себе гораздо слабее привязано к специфике задач SoftPLC (например, к интерактивному планированию задач реального времени в терминах объектов SoftPLC-среды). Другим недостатком универсальных расширений реального времени (по сравнению с CoDeSys SP RTE) является более высокая цена как на среду разработки, так и на среду исполнения (стоимость run-time лицензии на RTX+стоимость run-time лицензии на SoftPLC).

CoDeSys SP RTE

CoDeSys SP RTE – включает в себя расширение Windows NT/2000/XP, позволяющее выполнять приложения реального времени, созданные на языках МЭК. Эта возможность обеспечивается подменой стандартного обработчика прерываний от системного таймера. Этот обработчик вызывается периодически через заданный период (по умолчанию, 1мс) времени. Благодаря этому, CoDeSys SP RTE периодически получает управление, причем это никоим образом не нарушает работу приложений Windows NT, а только приостанавливает ее на один или несколько тактов системного таймера. Этот механизм продемонстрирован на рис.1.

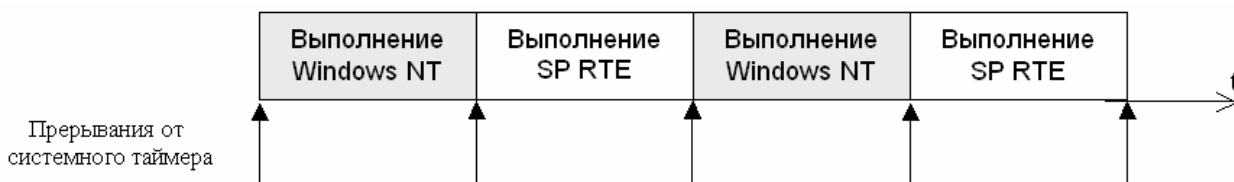


Рис. 1. Принцип выполнения CoDeSys SP RTE (длительность владения процессорным временем 50%)

Как видно из рисунка работа CoDeSys SP RTE периодически прерывается. При этом выполнение задач реального времени приостанавливается на один такт системного таймера, а затем снова возобновляется в соответствии с алгоритмом планирования. На рис.1 показан случай, когда Windows NT и CoDeSys SP RTE делят процессорное время пополам. Однако CoDeSys SP RTE позволяет разработчику задать произвольное соотношение времени владения процессором, т.е. возможна такая ситуация (рис. 2):

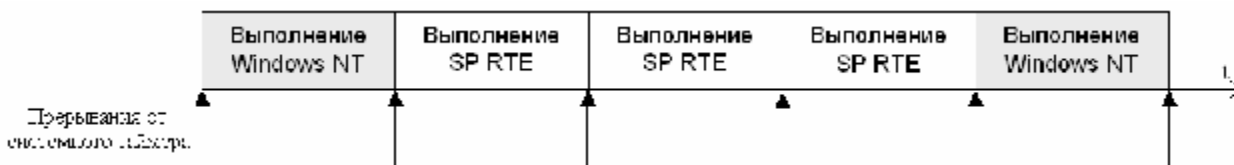


Рис. 2. Принцип выполнения CoDeSys SP RTE (длительность владения процессорным временем 80%)

Разработчик выбирает это соотношение самостоятельно, в соответствии с решаемой им задачей.

Для уменьшения времени реакции системы на внешние события период вызова SP RTE можно уменьшить. Для этого нужно уменьшить период вызова обработчика прерывания от системного таймера. Он может составлять 200, 100, 50 и 25мкс. Эта величина называется базовым квантом времени.

CoDeSys SP RTE имеет структуру, представленную на рис.3.

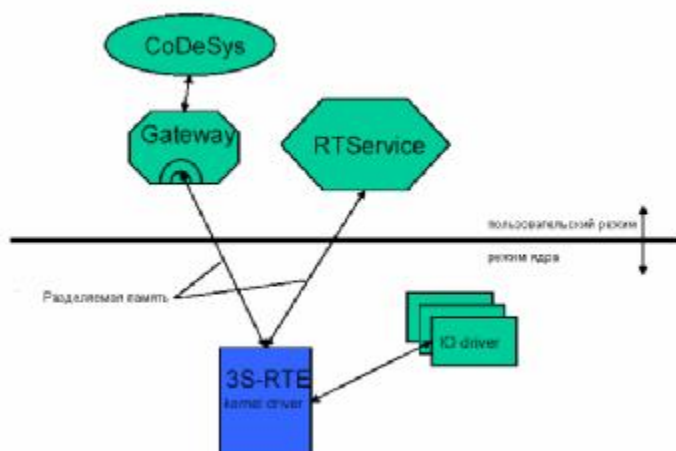


Рис.3. Структура CoDeSys SP RTE

SP RTE состоит из двух основных компонентов: системного сервиса RTService и драйвера режима ядра RTDrv.sys. RTService выполняется в пользовательском режиме и предназначен для конфигурирования и мониторинга работы системы исполнения, а также асинхронного вызова функций, выполняющихся в пользовательском режиме.

RTDrv.sys выполняется в режиме ядра. Именно этот драйвер вызывает планировщик задач и прикладные задачи реального времени, то есть задачи реального времени также выполняются в режиме ядра, что позволяет им обращаться к любым ресурсам целевой платформы.

Драйверы ввода/вывода (IO Driver) предназначены для управления различными внешними устройствами, например платами ввода/вывода. Они содержат функции инициализации аппаратуры, чтения входов, записи выходов, обработки прерываний и т.д. Шаблон для создания драйвера поставляется вместе с RTE в виде исходных текстов на языке C. Кроме того, вместе с RTE поставляются драйверы для наиболее распространенных устройств.

Gateway-сервер обеспечивает связь системы исполнения и системы программирования. Gateway с одной стороны связан с системой исполнения через разделяемую память, а с другой стороны с системой программирования. В стандартной поставке CoDeSys связь с системой программирования может осуществляться по протоколу TCP/IP или через разделяемую память (в этом случае система исполнения и система программирования должны выполняться на одной целевой платформе). Кроме того, Gateway имеет открытый интерфейс для разработки драйверов связи, благодаря чему можно поддерживать различные протоколы связи между системой программирования и системой исполнения, например, CAN или RS-232.

Функциональные возможности CoDeSys SP RTE

Многозадачность

Наличие многозадачности является одним из основных требований к системам реального времени, поэтому начнем именно с нее.

Основными элементами проекта в CoDeSys являются программы – объекты, предназначенные для отображения множества значений входных параметров на множество выходных [2], т.е. это некий программный код, который выполняется для достижения поставленных целей управления в соответствии с текущим состоянием объекта управления.

Проект может состоять из нескольких программ. Правила совместного выполнения программ определяются задачами (tasks), т.е. задача является единицей планирования.

Задача может состоять как из одной, так и нескольких последовательно выполняющихся программ.

Основные этапы выполнения задачи представлены на рис.4

1	Чтения состояния входов
2	Выполнение кода программы 1 Выполнение кода программы 2 · · Выполнение кода программы N
3	Запись состояния выходов

Рис.4. Основные этапы выполнения задачи

Под чтением состояния входов понимается копирование значения физических входов ПЛК (например, значение из АЦП) в специальную область памяти, называемую памятью входов. Программа на протяжении своего выполнения работает именно с этой областью памяти. Работа с выходами обратна работе с входами - прикладная программа пишет значения выходов в память выходов, а в физические выходы ПЛК устанавливаются на этапе 3.

В CoDeSys SP RTE реализована вытесняющая многоприоритетная многозадачность. Каждая задача имеет приоритет, заданный числом от 0 до 31, причем 0 – соответствует самому высокому приоритету. Планировщик задач вызывается в начале каждого кванта времени выделенного SP RTE (см. рис. 1). Если сразу несколько задач претендуют на выполнение, то сначала будет выполнена задача, приоритет которой выше. Выполнение задачи с меньшим приоритетом будет отложено до того момента, когда все задачи с более высоким приоритетом будут выполнены. Если в момент перепланировки процессором владеет задача, приоритет которой ниже, чем у задачи, претендующей на выполнение, то она будет вытеснена. При вытеснении задачи ее контекст (информация о текущем состоянии задачи) сохраняется. Это позволяет планировщику возобновить выполнение задачи с той же инструкции, на которой она была прервана, когда ее приоритет вновь станет самым высоким. Если прикладная задача не успела выполниться в течение базового кванта времени, то ее контекст также сохранится и будет восстановлен при следующей перепланировке (если конечно не появится задача с более высоким приоритетом)

В CoDeSys нет механизма диспетчеризации, то есть нельзя создавать несколько задач с одинаковыми приоритетами. Однако в одну задачу можно включать несколько программ.

Отсутствие механизма диспетчеризации приводит к тому, что разработчику приходится более тщательно выбирать приоритеты задач, а это, в свою очередь, повышает надежность создаваемых систем.

Для обмена данными между задачами можно использовать глобальные переменные.

В RTE используются такие механизмы синхронизации задач, как семафоры и события.

Типы и свойства задач SP RTE

В SP RTE поддерживаются следующие типы задач:

- Циклические (cyclic) - вызываются через определенные периоды времени. Значение этого периода должно быть кратно базовому кванту времени. Например, если базовый квант времени равен 50мкс, то в

качестве периода вызова циклических задач можно задать 100, 150, 200 мкс и т.д.

- По событию (triggered by event) – задача вызывается, когда значение событийной переменной (event variable) равно истинно.
- По внешнему событию (triggered by event) – аналогичны предыдущему типу, только событие определяется конфигурацией целевой платформы.
- По системному событию – задачи вызываются при возникновении системного события. К системным событиям относятся такие события как возникновение прерывания, исключения, старт/стоп ПЛК. Такие задачи не управляются планировщиком и вызываются непосредственно при возникновении системного события. Так что такая задача в любом случае прерывает выполнение остальных прикладных задач.
- Свободно выполняемые задачи (freewheeling) – этот тип задач аналог задач idle в ОС Windows и QNX. Эта задача имеет самый низкий приоритет и выполняется в последнюю очередь, если процессор не занят выполнением остальных задач.

На каждую задачу можно установить сторожевой таймер. Он определяет максимальное время выполнения задачи. Если значение установленного порога будет превышено, то система будет либо остановлена, либо, если задан обработчик системного события `excpt_watchdog`, будет вызван обработчик этого события. Требования к устанавливаемым временным порогам такие же, как и к периоду вызова циклических задач.

Для управления задачами используется встроенный в систему программирования CoDeSys конфигуратор задач (рис.5).

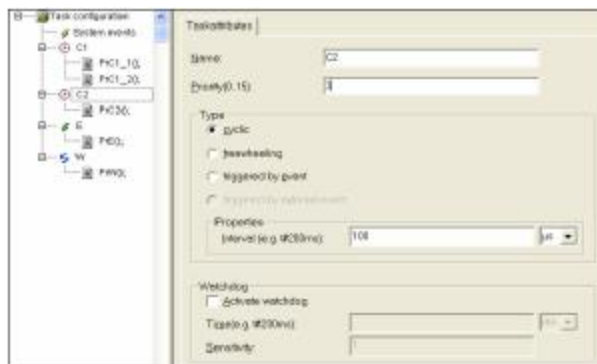


Рис.5. Конфигуратор задач CoDeSys

Рассмотрим пример планирования задач. Пусть базовый квант времени равен 50мкс, т.е. планировщик задач RTE вызывается с периодом 100мкс.

Создадим в конфигураторе задач 5 задач:

- циклическая задача С1 с приоритетом 2 и периодом вызова 200мкс
- циклическая задача С2 с приоритетом 3 и периодом вызова 100мкс
- выполняемая по событию bEvent задача Е с приоритетом 1
- задача-обработчик некоторого системного события S
- Свободно выполняемая задача W

Процесс планирования и выполнения этих задач представлен на рис.6.

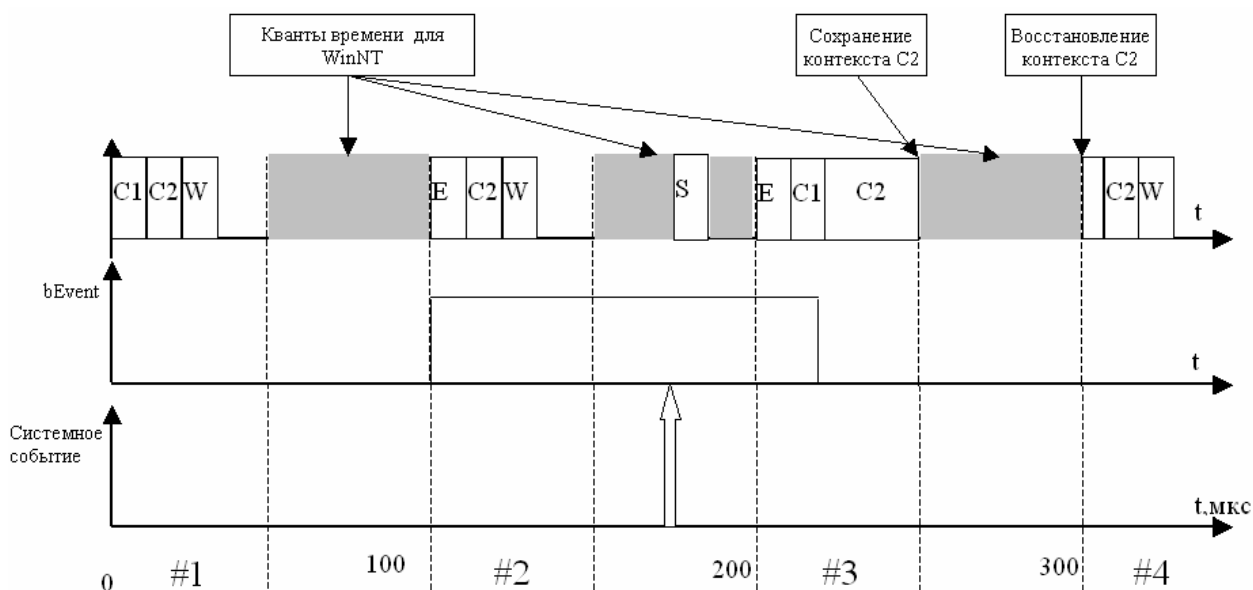


Рис.6. Пример планирования и выполнения задач

Задача C2 выполняется в каждом интервале времени, выделенного для RTE, так как период вызова этой задачи равен периоду вызова планировщика SP RTE. Задача C1 вызывается в два раза реже. Задача E вызывается в случае, когда переменная bEvent имеет значение TRUE.

Если сразу несколько задач претендуют на выполнение, то они будут выполнены последовательно согласно их приоритету. Задача W выполняется в каждом кванте времени, если остальные задачи закончили свое выполнение до истечения текущего кванта времени.

Если задача не успевает выполниться в течение выделенного кванта времени, как это произошло с задачей C2 в кванте #3), то она вытесняется и ее выполнение возобновляется в следующем кванте. Причем в кванте #4 выполнение задачи C2 будет завершено и возобновлено заново.

Задача S вызывается по системному событию независимо от того, когда оно произошло – во время выполнения RTE или во время выполнения Windows NT.

Доступ к устройствам ввода/вывода

Так как SP RTE используется для решения задач управления, то она должна предоставлять удобные механизмы для доступа к различным устройствам ввода/вывода.

В SP RTE есть несколько таких механизмов. Прежде всего, это интерфейс драйверов ввода/вывода. Каждый драйвер управляет отдельным устройством ввода/вывода. Он содержит функции работы с входами/выходами, которые вызываются на этапах 1 и 3 выполнения программы (см. рис. 4).

Штатный комплект поставки SP RTE уже содержит драйвера для наиболее распространенных устройств. Кроме того, в состав поставки RTE входит пакет разработчика драйверов ввода/вывода, который содержит исходные тексты шаблона драйвера (с комментариями TODO) и документацию. Это позволяет разработчику самому создать драйвер для любого устройства.

Конфигурирование внешних устройств в этом случае производится с помощью конфигулятора ПЛК(PLC Configuration) – объекта, встроенного в систему программирования (см. рис. 7).

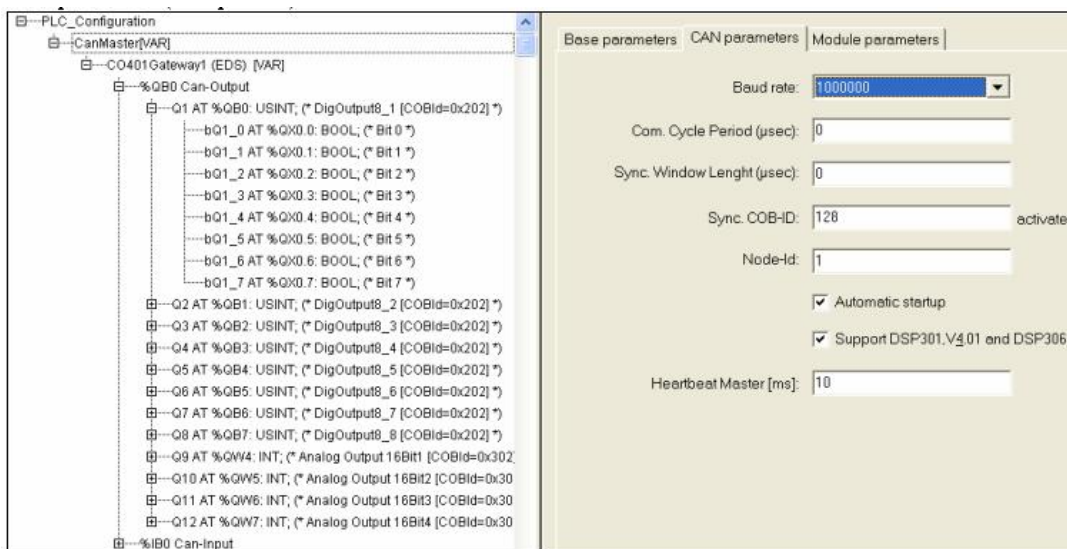


Рис.7. Конфигуратор ПЛК

Более простым способом для работы с внешними устройствами является использование внешних библиотек. Функции внешних библиотек используются в проекте как обычные МЭК-функции, но их особенность в том, что реализованы они во внешнем объектном файле, т.е. их можно писать языке С. Эти функции работают в режиме ядра и могут обращаться ко всем ресурсам целевой платформы. Это удобно при реализации низкоуровневых функций доступа к внешним устройствам, например функции чтения данных из регистра периферийного устройства.

Кроме того, в штатную поставку CoDeSys входят системные библиотеки SysLibPorts.lib и SysLibCom.lib. Библиотека SysLibPorts.lib позволяет работать с портами ввода/вывода, что дает возможность управлять устройствами, подключенными к этим портам. SysLibCom.lib используется для работы с последовательными интерфейсами.

Сетевой обмен

CoDeSys можно использовать для создания распределенных систем управления благодаря тому, что он поддерживает работу с такими полевыми сетями, как ProfiBus, DeviceNet и CANOpen. Для работы с этими сетями используется конфигуратор ПЛК (см. рис. 7). Он делает работу с удаленными узлами сети такой же простой, как с обычными входами/выходами.

Другим способом сетевого обмена в CoDeSys SP RTE является использование сетевых переменных. Это переменные, доступные всем узлам сети, то есть один узел сети может устанавливать эти переменные, а другие узлы считывать. Механизм сетевых переменных реализован для протоколов CAN и UDP.

Для работы с протоколами UDP и TCP можно пользоваться стандартным механизмом – сокетами. Функции работы с сокетами реализованы в библиотеке SysLibSockets.lib и аналогичны функциям системной библиотеки Windows Winsock.dll.

Взаимодействие с Windows

Как уже было сказано выше, одним из главных преимуществ расширений реального времени является использование возможности взаимодействия с обычными приложениями Windows.

Например, необходимо создать приложение реального времени, которое, наряду с остальными, будет использовать устройства, работа с которыми не критична по времени. Большинство таких устройств имеет драйверы только под Windows. В этом случае логично для работы с этими устройствами использовать именно Windows, а в проекте

CoDeSys обеспечить только логику их функционирования. Для этих целей SP RTE имеет специальный механизм асинхронного вызова функции из DLL. Эти функции выполняются под управлением Windows и могут получить доступ ко всем ее ресурсам - файловой системе, драйверам и т.д.

Данный механизм реализуется с помощью библиотеки DllCall, содержащей одноименный функциональный блок (см. рис. 8).

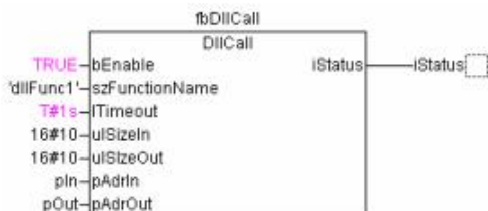


Рис.8. Функциональный блок DllCall

На вход функционального блока поступает имя функции, которая должна быть выполнена и эта функция выполняется асинхронно по отношению к задачам CoDeSys в течение времени, выделенного для выполнения Windows NT. Состояние выполнения функции можно определить, вызвав данный функциональный блок еще раз. Если выходной параметр iStatus равен STATUS_PROCESSING, то функция еще находится в процессе выполнения, а если STATUS_READY, то функция была успешно выполнена ОС Windows. Выходной параметр iStatus может принимать еще ряд значений, которые позволяют контролировать процесс выполнения функции, например, если он равен STATUS_EXCEPTION, то функция в процессе своего выполнения вызвала исключение. Функциональный блок DllCall также обеспечивает обмен данными между RTE и Windows NT.

Еще одна возможность взаимодействия с Windows NT заключается в том, что из DLL можно асинхронно получить доступ к адресному пространству, в котором выполняются прикладные задачи RTE. В некоторых случаях это очень удобно.

Заметим, что для работы со SCADA-системами, в том числе установленными на целевой платформе, достаточно лишь настроить OPC-сервер, который поставляется вместе с системой программирования бесплатно.

Заключение

CoDeSys SP RTE, как интегрированная среда, обеспечивает полную функциональную поддержку для решения задач пользователя, в том числе интеграцию со SCADA-системами (и даже с установленными на целевой платформе) путем настройки OPC-сервера, который бесплатно поставляется вместе с системой программирования. Другим применением CoDeSys SP RTE является возможность использования его в рамках CoDeSys SoftMotion - функциональным набором средств управления движением, встроенным в среду SoftPLC. Систему программирования CoDeSys и демо-версию CoDeSys SP RTE можно скачать на сайте <http://www.3s-software.ru>

Литература

1. С.В.Золотарев, И.В.Петров, CoDeSys – интегрированный комплекс МЭК 61131-3 программирования, Приборы и системы управления, N3, 2005.
2. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования /Под ред. проф. В.П. Дьяконова. – М.: СОЛОН-Пресс, 2004. – 256 с.: ил. - (Серия “Библиотека инженера”)
3. П.Кирюхин, RTX – расширение реального времени для Windows NT, http://citforum.cn.ua/operating_systems/rtx/index.shtml
4. Martin Timmerman, Jean-Christophe Monfret, Windows NT as Real-Time OS ? Real-Time Magazine, 97Q2