

*Московский ордена Ленина, ордена Октябрьской Революции
И ордена Трудового Красного Знамени
Государственный Технический Университет им. Н.Э. Баумана*

Кафедра «Гидромеханика, гидромашины и гидроневмоавтоматика» (Э-10)

**Отчёт перед ЗАО «СВД Софтвр»
об использовании операционной системы
QNX Neutrino 6.3 в учебном процессе
в 2006/2007 учебном году.**

Руководитель: Семёнов С.Е., к.т.н., доцент кафедры Э-10.
Исполнители: Кулаков Д. Б., старший преподаватель кафедры Э-10.
Водолажский В. В., аспирант кафедры Э-10.

МОСКВА
2007

Содержание:

1. Учебный курс.	3
2. Учебно-исследовательский стенд для исследования системы ЭГСП.....	5
2.1 Компоновка стенда.....	5
2.2 Задача управления системой приводов.	7
3. Принципы построения программного обеспечения системы управления(ПОСУ)..	8
3.1 Общая концепция построения ПОСУ.....	8
3.2 Практическая реализация ПОСУ лабораторного робота.	13
4. Заключение.	15

1. Учебный курс.

Кафедра «Гидромеханика, гидромашины и гидропневмоавтоматика» (Э-10) в 2006/2007 учебном году в рамках некоммерческой образовательной программы «QNX для вузов» использовала лицензионные комплекты операционной системы реального времени QNX Neutrino 6.3 компании QNX Software Systems при разработке учебного курса для обучения студентов специальности 121100 – «Гидромашины, гидроприводы и гидропневмоавтоматика» направления подготовки 657400 «Гидравлическая, вакуумная и компрессорная техника» по дисциплине «Основы мехатроники гидро и пневмосистем».

Описание учебного курса:

Лекции:

Современные тенденции развития электронных управляющих устройств гидропневмосистем (ГПС), особенности их проектирования; Промышленные компьютеры, однокристалльные и другие микроконтроллеры, микропроцессоры в ГПС; Структура микропроцессорной системы управления ГПС; Распространенные классы и типы микропроцессоров и микроконтроллеров; Современные операционные системы реального времени, среды и языки программирования; Программы реального времени для управления ГПС.

Лабораторные работы обеспечивают наглядность и конкретность излагаемого материала, создание физического образа полученных знаний по дисциплине.

1. Лекции.

1.1. Устройства ввода-вывода и преобразования электрических сигналов. Применение устройств согласования в ГПС. Использование сетевых технологий для управления ГПС.

1.2. Общие сведения об операционных системах реального времени, средах и языках программирования.

1.3 Варианты построения программ управления ГПС. Примеры приложений реального времени для ОС РВ.

1.4 Примеры построения аппаратуры и ПО типовых ГПС.

2. Лабораторные работы.

- Демонстрация цифровых устройств кодирования и передачи информации.
- Изучение микропроцессорной системы управления ГПС; общая шина, устройства ввода вывода и преобразования электрических сигналов.
- Демонстрация работы ГПС под управлением компьютера.
- Разработка и отладка управляющей программы под ОС РВ.

Лекции.

В лекции 1 студенты знакомятся с устройствами ввода-вывода и преобразования электрических сигналов (ЦАП/АЦП). Приводятся примеры применения устройств согласования в гидропневмосистемах. А так же рассказывается о использовании сетевых технологий для управления гидропневмосистемами (технология разделения на ЭВМ верхнего и нижнего уровня). Лекция служит теоретической основой для лабораторных работ №1 и №2.

В лекции 2 студенты получают общие сведения об операционных системах реального времени, средах и языках программирования. В качестве примеров ОСРВ

приводятся VxWorks, Windows CE.NET, QNX. Лекция в основном строится на материалах статьи «Операционные системы реального времени» книги «Практика работы с QNX» Москва, Издательский Дом «КомБук» 2004. Учитывая, что в дальнейшем основное внимание уделяется применению ОСРВ QNX в качестве языка программирования описывается C/C++, а в качестве сред программирования – Photon АВ и IDE. Для подготовки к лекции 3 и лабораторных работ №3 и №4 студентам выдаётся для ознакомления статья Олега Цирюлика «Построение приложений в PhAB» книги «Практика работы с QNX» Москва, Издательский Дом «КомБук» 2004 (статья также доступна на сайте qnx.org.ru).

В лекции 3 студенты знакомятся с примерами построения приложений реального времени. В качестве примеров приводятся программы **Input reader** (считывание данных с АЦП и вывод результатов в текстовом виде на экран), **Sliders** (считывание данных со слайдеров вывод информации в графическом виде), **Time meter** (программный «секундомер», с выводом на экран времени выполнения различных участков кода). Данные программы являются «кирпичиками» для построения управляющей программы (лабораторная работа №4).

В лекции 4 приводятся примеры построения аппаратуры и ПО типовых ГПС. Лекция является практическим обобщением предыдущих.

Лабораторные работы.

Лабораторные работы №1 и №2 служат для практического ознакомления студентов с устройствами ввода-вывода и преобразования электрических сигналов (ЦАП/АЦП). В качестве лабораторного образца используется 12-ти разрядный ЦАП/АЦП – плата cio-das08/jr-ao фирмы “Measurement computing”. Из-за недостаточности мощности сигнала для непосредственного управления возникает необходимость использования дополнительного электрического усилителя мощности. Лабораторные работы служат для практического закрепления материалов лекции 1.

Лабораторная работа №3 носит демонстрационный характер. На данной лабораторной работе студентам демонстрируется работа ГПС под управлением компьютера в различных режимах (ручное управление, автоматическое, отработка заданной последовательности действий и т.п.).

Лабораторная работа №4 служит для практического закрепления полученных знаний и создания первичных навыков практического написания программ в PhAB.

Виды учебных работ	Объём работ, в часах	11 семестр, 13 недель
Выделено на дисциплину	52	52
Аудиторная работа:	39	39
Лекции	26	26
Семинары	0	0
Лабораторные работы	13	13
Самостоятельная работа	13	13
Домашнее задание	10(15)	10(15)
Контроль знаний		экзамен

Данный учебный курс в указанном объёме был прочитан группе из 20 студентов в осеннем семестре 2006 года.

2. Учебно-исследовательский стенд для исследования системы ЭГСП.

Особенностью учебного процесса в МГТУ им. Н. Э. Баумана является постоянное сочетание теории и практики. Совместно с курсом лекций проводятся лабораторные занятия, чтобы студенты не просто заучивали преподаваемый материал, а могли разобраться в сути происходящих явлений.

В рамках некоммерческой образовательной программы «QNX для вузов», реализуемой компанией SWD Software Ltd., официальным дистрибьютором QNX в России и на территории стран бывшего СССР, для студентов специальности 121100 – «Гидромашины, гидроприводы и гидропневмоавтоматика» проводится обучение в курсе дисциплины «Основы мехатроники гидро и пневмосистем» операционной системе реального времени QNX Neutrino компании QNX Software Systems.

2.1 Компоновка стенда.

В учебном процессе необходим современный стенд для изучения системы электрогидравлических следящих приводов (ЭГСП). Стенд был реализован на базе 3-х степенного робота (фото 1).



Фото 1.

На роботе установлена система из 3 гидравлических приводов с обратными связями по положению (кинематическая схема приводится на рис.1). Стенд также включает в себя насосную станцию.

Кинематическая схема МЕХАНИЗМА

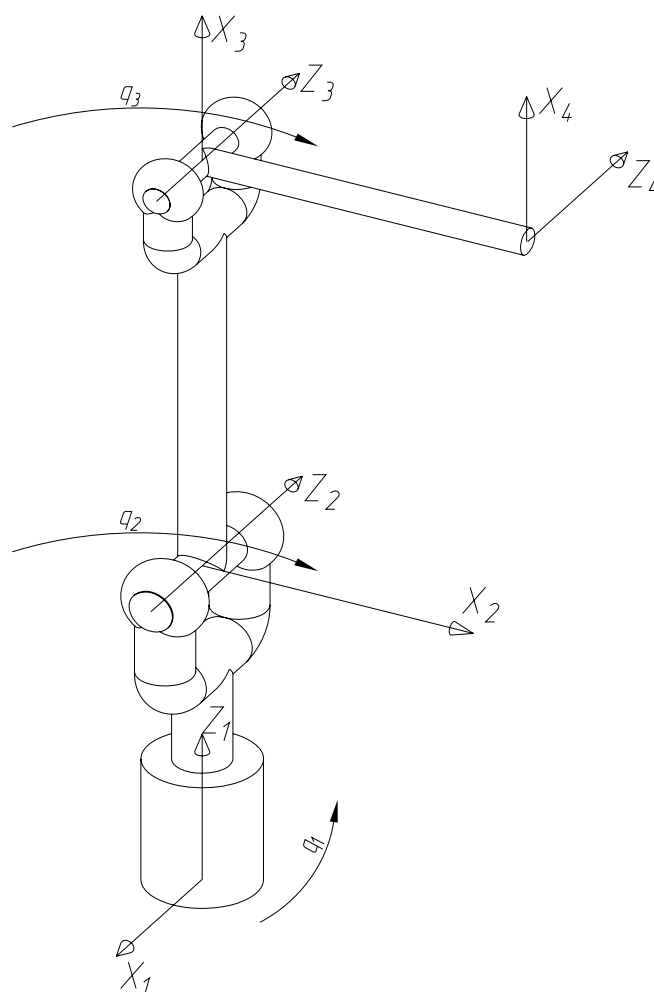


Рис. 1

Архитектура управляющей системы представляет собой одно или двухмашинную конфигурацию (для иллюстрации примеров реализации).

Двухмашинная конфигурация реализована на основе «бортовой» ЭВМ (машина низкого уровня), устройств сопряжения и машины оператора (машина высокого уровня).

Машина оператора реализована на базе процессорной платы Intel (Intel® Communications ICH Value Appliance Universal Reference Platform) полученной в рамках некоммерческой образовательной программы «QNX для вузов».

В одно и двухмашинной системе в качестве операционной системы используется ОС PV QNX Neutrino 6.3 компании QNX Software Systems, предоставленная компанией SWD Software Ltd. в рамках некоммерческой образовательной программы «QNX для вузов».

Задачи, решаемые в процессе управления, по функциональному назначению можно разделить на несколько групп. В частности, выделим следующие три:

- Формирование интерфейса взаимодействия оператора с роботом.
- Задачи, связанные с использованием типичных для робототехники (и сложных в вычислительном отношении) алгоритмов управления движением многозвенных механизмов.
- Управление системой приводов робота.

Все эти задачи требуют периодического выполнения определенных алгоритмов, причем для первой группы задач типичны частоты повторения порядка 10 Гц, для второй - 100 Гц, а для третьей - 1000 Гц. На третью группу приходится и основная доля взаимодействия с различным внешним по отношению к системе управления оборудованием.

Подробнее подход к построению и реализации программного обеспечения системы управления приводится в разделе 3 данного отчёта.

2.2 Задача управления системой приводов.

Задача управления сводится в выработке управляющего воздействия на каждый привод с учётом взаимного влияния приводов, обусловленного кинематикой системы.

Имея координаты схвата манипулятора (желаемые), определяем углы в шарнирах механизма.

Требуемые координаты схвата: $\mathbf{M}(x_0, y_0, z_0)$

Для угла в первом шарнире:

$$q_1 = \arctg\left(\frac{x_0}{y_0}\right)$$

Для угла во втором шарнире:

$$q_3 = \arcsin\left(\frac{x_0^2 + y_0^2 + z_0^2 - a_2^2 - a_3^2}{2 \cdot a_2 \cdot a_3}\right)$$

Для угла в третьем шарнире:

$$q_2 = \arcsin\left(\frac{x_0^2 + y_0^2}{\sqrt{a_3^2 \cdot \cos^2(q_3) + (a_3 \cdot \sin(q_3) + a_2)^2}}\right) - \arcsin\frac{a_3 \cdot \cos(q_3)}{\sqrt{a_3^2 \cdot \cos^2(q_3) + (a_3 \cdot \sin(q_3) + a_2)^2}}$$

Обозначения в формулах:

q_1 – координата в первом сочленении

q_2 – координата во втором сочленении

q_3 – координата в третьем сочленении

a_2 – расстояние между началами второй и третьей системы координат

a_3 – расстояние между началами третьей и четвертой системами координат.

После вычисления требуемых углов (q_1 , q_2 , q_3) для каждого привода на основании данных датчика положения вычисляется требуемое управляющее воздействие.

Внешний вид управляющей программы представлен на рис. 2.

На сегодняшний день материальная часть стенда позволяет в рамках курса «Основы мехатроники гидро и пневмосистем» проводить лабораторные работы и даёт возможность исследовать различные алгоритмы управления системой приводов, разрабатывать методики исследования электрогидравлических приводов различного назначения.

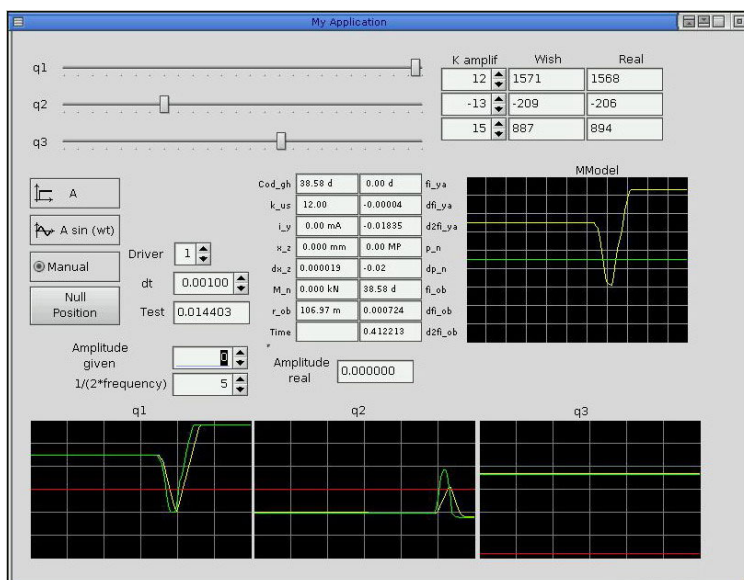


Рис. 2

Выполненная работа способствовала созданию современного учебного оборудования, что позволит студентам при изучении вопросов управления гидросистемами, не только ознакомиться с устройством и особенностями работы средств цифрового управления в составе ЭГСП, но и освоить методику построения управляющих программ под ОС РВ QNX Neutrino.

3. Принципы построения программного обеспечения системы управления(ПОСУ).

3.1 Общая концепция построения ПОСУ.

Как было ранее упомянуто, задачи, решаемые в процессе управления, по функциональному назначению можно разделить на несколько групп. В частности, выделим следующие три:

- Формирование интерфейса взаимодействия оператора с роботом.
- Задачи, связанные с использованием типичных для робототехники (и сложных в вычислительном отношении) алгоритмов управления движением многозвенных механизмов.
- Управление системой приводов робота.

Все эти задачи требуют периодического выполнения определенных алгоритмов, причем для первой группы задач типичны частоты повторения порядка 10 Гц, для второй - 100 Гц, а для третьей - 1000 Гц. На третью группу приходится и основная доля взаимодействия с различным внешним по отношению к системе управления оборудованием.

Очевидно, что система управления роботом, призванная решать настолько разнообразные и сложные задачи, должна строиться на основе сети компьютеров с соответствующим программным обеспечением системы управления (ПОСУ).

Накопленный опыт позволяет из множества решаемых проблем выделить следующие:

- Необходимость эффективной сетевой организации.
- Требование точно выдерживать временные интервалы и обеспечивать гарантированно малое время реакции на внешние события, т.е. реализовать работу системы в жестком реальном времени.

- Необходимость синхронизировать решение различных задач между собой, организовать эффективный обмен данными.
- Проблема неконтролируемого взаимовлияния различных задач.
- Затрудненность совместной работы нескольких разработчиков – специалистов разных областях, как правило не являющихся профессиональными программистами.

Эти проблемы практически блокируют дальнейшее развитие ПОСУ при достижении определенного уровня сложности, однако они могут быть эффективно решены в рамках следующей концепции построения ПОСУ:

- Рациональный выбор операционной системы реального времени (ОСРВ) с развитыми сетевыми средствами и средствами взаимодействия процессов между собой.
- Оформление логически обособленных задач управления роботом в виде отдельных процессорных задач (ПЗ), которые ОС выполняет как независимые процессы.
- Возложение функции обеспечения взаимодействия отдельных ПЗ (обмен данными, командами, синхронизация работы) в рамках всей сети на специальный процесстранспорт (ПТ). Он также является независимым процессом, выполняемым ОС.
- Введение единых для всех ПЗ системы протоколов обмена данными и командами (событиями).
- Использование специальных заготовок ПЗ, содержащих значительную часть кода, а также единой методики написания текста программы ПЗ и подключения ПЗ к системе.

В качестве основы для построения ПОСУ работа применена ОС жесткого реального времени QNX. В настоящее время это надежная многозадачная UNIX-подобная ОС, поддерживающая различные, в том числе много-процессорные, аппаратные платформы, а также многопоточность. QNX обладает уникальными сетевыми свойствами. Средства взаимодействия компьютеров, объединенных в сеть, настолько развиты, что сеть превращается практически в единый компьютер. Это позволяет процессам, выполняемым на разных компьютерах сети, взаимодействовать почти столь же эффективно, что и выполняемым на одном компьютере.

Кроме того, в QNX имеется хотя и не POSIX-совместимое, но очень эффективное в системах жесткого реального времени средство взаимодействия процессов – сообщение. Различают блокирующие (обеспечиваются функциями ОС `MsgSend()`, `MsgReceive()`, `MsgReply()` и др.) и неблокирующие (обеспечиваются функциями `MsgReceivePulse()`, `MsgSendPulse()`) сообщения. Они позволяют с минимальными накладными расходами синхронизировать выполнение процессов и обеспечивают передачу данных.

Одно из важнейших свойств QNX – малое и гарантированно не превышающее определенный предел время реакции на прерывание. Именно это свойство делает данную ОС системой жесткого реального времени.

В ОС QNX каждый процесс выполняется в своем, не доступном другим процессам, адресном пространстве и имеет назначаемый пользователем приоритет, который может превышать приоритет системных процессов. Оформление каждой задачи или группы задач в виде отдельного процесса позволяет уменьшить степень неконтролируемого влияния решаемых задач управления друг на друга. Программировать такие ПЗ могут разные люди – специалисты в решении конкретной задачи. При этом они меньше мешают друг другу. Раньше же, если новый разработчик включал участок своего кода в программу, разработанную другим человеком, даже под его контролем, часто возникали трудно выявляемые ошибки. Иногда ПОСУ полностью разрушалось, и большой объем работы приходилось делать заново.

Средства QNX позволяют одновременно выполняющимся процессам обмениваться сообщениями без посредников. Если система не очень сложная, такой подход позволяет избежать накладных расходов. Однако, по мере усложнения системы и увеличения срока работы над ней, непосредственные связи становятся запутанными. Если

учесть, что ПОСУ робота – это динамическая система, такая структура может работать неустойчиво, часто возникают сбои.

Упорядочить связи позволяет специальный процесс-транспорт (ПТ) (или при использовании сети – группа ПТ). При использовании ПТ отдельные ПЗ взаимодействуют посредством сообщений только с ним и никогда - между собой. Совместно используемые данные в этом случае размещаются в общей области памяти, доступ к которой контролирует ПТ. Это позволяет избежать несанкционированной модификации общих данных, некорректной передачи массивов динамически изменяемых данных (например смысл матрицы положения звена робота будет полностью потерян, если попытаться прочитать ее в то время, когда она обновлена лишь частично).

ПТ, работающие на каждом компьютере сети, берут на себя обмен сообщениями по сети, своевременное обновление массивов данных в общей памяти, если ПЗ-источник и ПЗ-потребитель данных расположены на разных компьютерах. Таким образом, формально, с точки зрения написания кода, конкретному ПЗ не важно, где расположены ПЗ-источники и ПЗ-потребители данных, с которыми он работает.

Существенно повышает скорость разработки, модернизации и отладки ПОСУ использование единых для всех ПЗ протоколов взаимодействия с ПТ. Это позволяет быстро изменять код конкретных ПЗ и их общее количество в системе. Кроме того, это упрощает подключение к работе над проектом новых разработчиков, что особенно важно именно в нашем случае. Напомним, что данная работа ведется в вузе с участием студентов.

Большим подспорьем стало использование единой программы заготовки, а также оформленной методики разработки ПЗ и подключения его к системе. Такая возможность появилась именно благодаря использованию единых протоколов взаимодействия ПЗ и ПТ.

В качестве основного структурообразующего принципа построения ПОСУ принята схема клиент-сервер. Сервером является многопоточный ПТ. ПЗ являются клиентами. Для взаимодействия с каждым ПЗ в процессе инициализации системы ПТ создает по два отдельных потока-сервера. Они, в свою очередь - нумерованные каналы для связи ПЗ с ПТ. Одна из пар поток-канал предназначена для обеспечения обмена данными, другая - для синхронизации во времени работы ПЗ между собой.

Механизм обмена данными между двумя ПЗ проиллюстрирован на рис. 3.

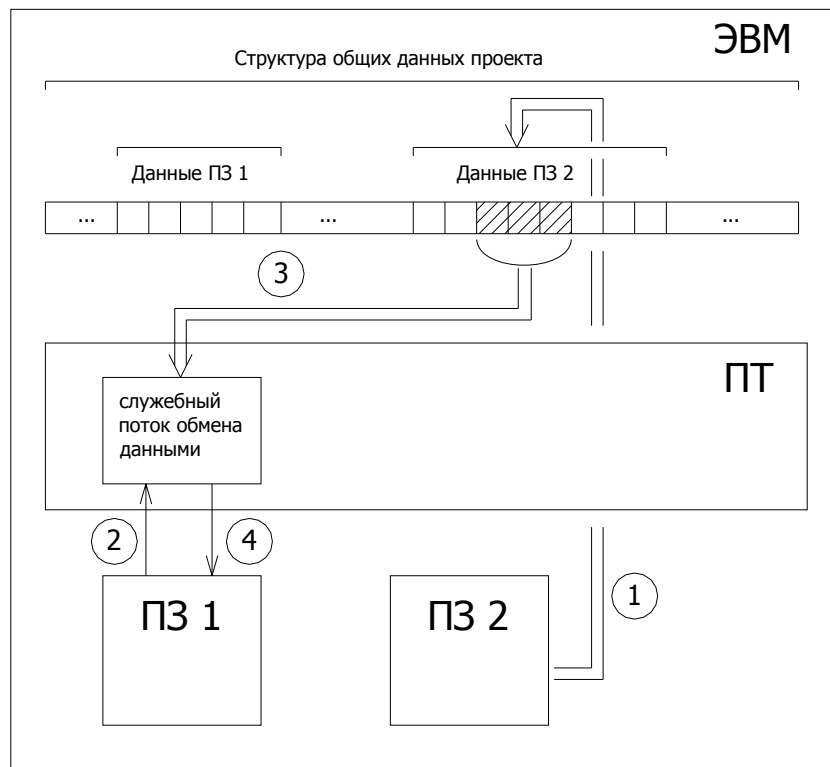


Рис. 3. Схема обмена данными между ПЗ, выполняющимися на одной ЭВМ.

Последовательность действий выполняемых ПОСУ при обмене данными между ПЗ выполняющимися на одной ЭВМ:

1. ПЗ 2 изменяет предназначенные для обмена данные в отведенной ей части общей области памяти;
2. ПЗ 1 посылает служебному потоку ПТ запрос на требуемые данные;
3. служебный поток связи с ПЗ определяет, что владельцем запрашиваемых данных является ПЗ 2, выполняющийся на локальной ЭВМ, и производит их чтение из структуры общей памяти на локальной ЭВМ;
4. в ответном сообщении ПЗ 1 получает запрошенные данные.

Таким образом исключается нарушение целостности данных, передаваемых от одного ПЗ другому данные передаются асинхронно по отношению к ПЗ – владельцу данных.

Если вычислительные мощности ПК не позволяют реализовать работу ПОСУ на одной ЭВМ, то структура построения ПОСУ позволяет распределить выполнение программных модулей из которых состоит ПОСУ между несколькими ЭВМ. При этом несколько ЭВМ объединяются в локальную вычислительную сеть и для каждого ПЗ указывается номер ЭВМ, на которой данный ПЗ будет выполняться. Таким образом структура созданного ПОСУ позволяет распределять уже написанные ПЗ между несколькими ЭВМ без изменений в текстах их программ. Единственно что при этом нужно учитывать – это влияние ограничений, возникающие при взаимодействии ПЗ, выполняющихся на разных ЭВМ, на работу ПОСУ в целом.

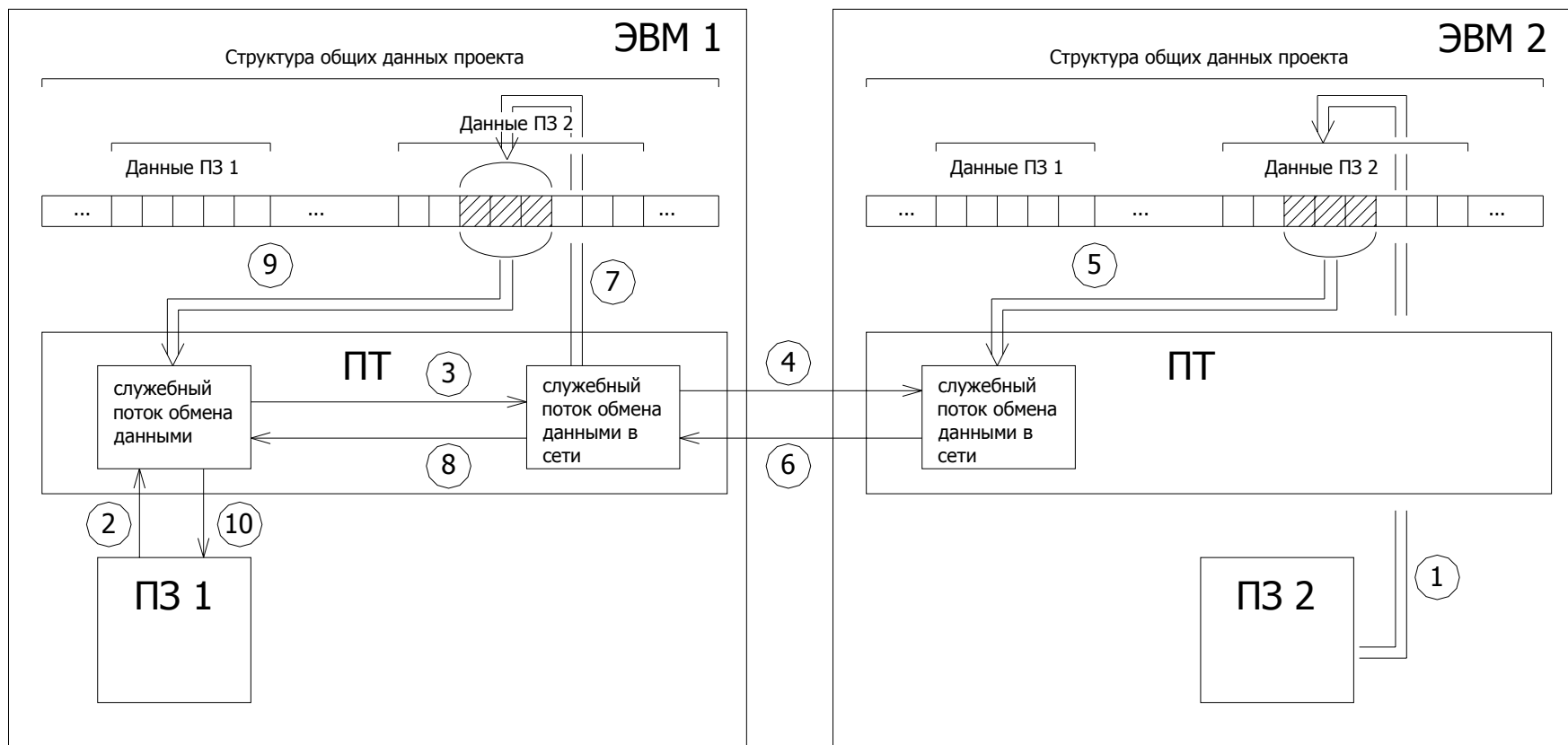


Рис. 4. Схема обмена данными между ПЗ, выполняющимися на разных ЭВМ.

Последовательность действий выполняемых программным комплексом при обмене данных между ПЗ выполняющимися на разных ЭВМ (рис. 4):

1. ПЗ 2 изменяет предназначенные для обмена данные в отведенной ей части общей области памяти;
2. ПЗ 1 посылает служебному потоку ПТ запрос на требуемые данные;
3. служебный поток связи с ПЗ определяет, что владельцем запрашиваемых данных является ПЗ 2, выполняющийся на удаленной ЭВМ, и посылает служебному потоку обмена данными в сети сообщение-запрос на требуемые данные;
4. служебный поток ПТ обмена данными в сети на локальной ЭВМ посылает сообщение-запрос на требуемые данные служебному потоку обмена данными в сети на удаленной ЭВМ;
5. служебный поток ПТ обмена данными в сети на удаленной ЭВМ производит чтение требуемых данных из структуры общей памяти;
6. служебный поток ПТ обмена данными в сети на удаленной ЭВМ в ответном сообщении пересылает требуемые данные;
7. служебный поток ПТ обмена данными в сети на локальной ЭВМ записывает полученные данные в структуру общей памяти на локальной ЭВМ;
8. служебный поток ПТ обмена данными в сети на локальной ЭВМ в ответном сообщении служебному потоку связи с ПЗ указывает, что данные получены;
9. служебный поток связи с ПЗ производит чтение требуемых данных из структуры общей памяти на локальной ЭВМ;
10. в ответном сообщении ПЗ 1 получает запрошенные данные.

Если ПЗ запрашивающий данные, целенаправленно не нарушает принятый в системе протокол обмена данными, то он получает через ПТ копии данных, находящихся области общей памяти. Это защищает данные, формируемые ПЗ владельцем от их случайного изменения. Кроме того, этот механизм позволяет ПЗ получать данные с любой ЭВМ системы используя единый протокол.

Предложенная концепция построения программного обеспечения системы управления позволяет решить изначально поставленные задачи, ускорить процесс создания нового ПОСУ, а также расширения функциональности имеющихся ПОСУ как одним разработчиком, так и группой разработчиков.

3.2 Практическая реализация ПОСУ лабораторного робота.

После анализа обмена данными между ПЗ, выполняющимися на разных ЭВМ, и использования ими информации о происшедших событиях, в рамках выполняемых ими задач в процессе работы системы управления роботом, была разработана схема сетевого взаимодействия ПТ, показанная на рис. 5.

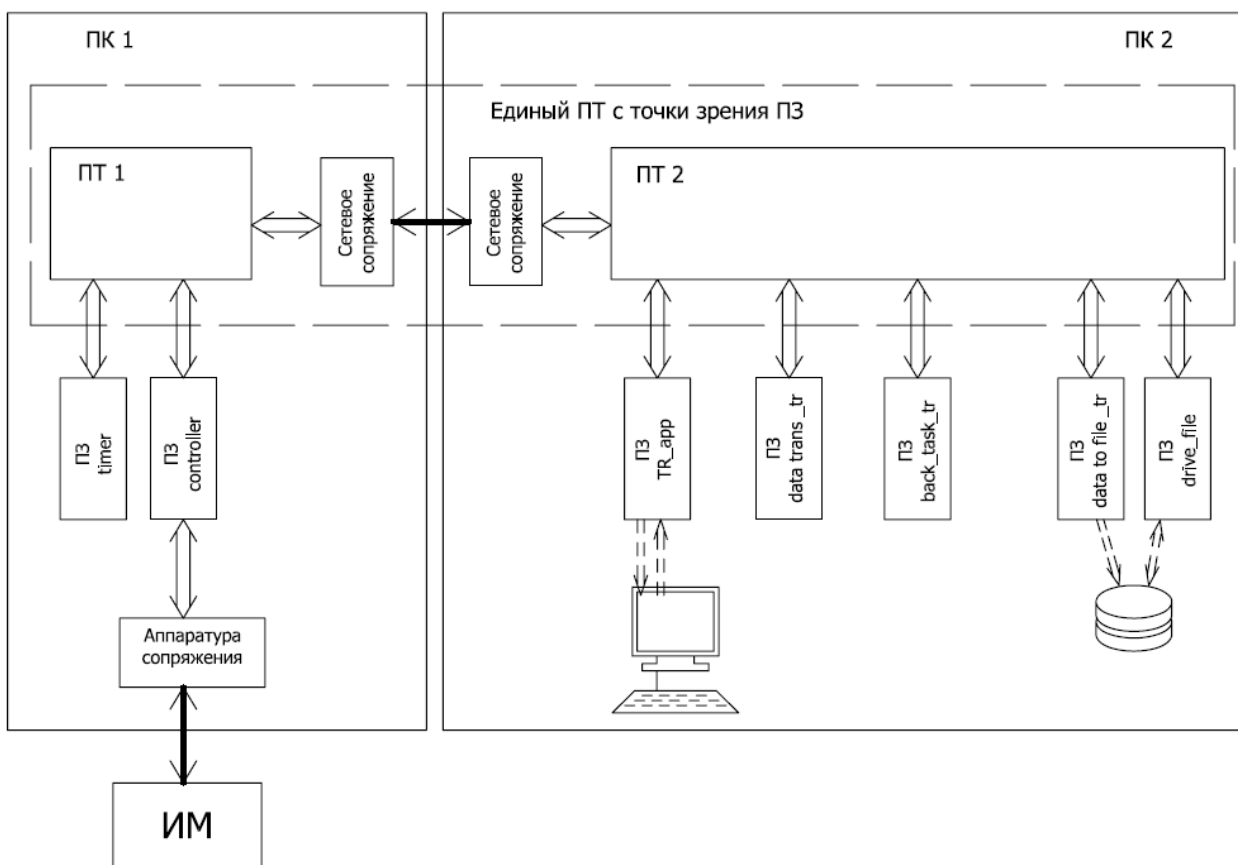


Рис. 5 Структура ПОСУ робота.

- ПТ1, ПТ2 – процессы-транспорты, выполняющиеся на ЭВМ1 и ЭВМ2 соответственно;
- ПЗ-timer – процесс-задача синхронизирующий ПОСУ в реальном времени;
- ПЗ-controller – ПЗ, в котором происходит обмен информацией с роботом через платы сопряжения и ПЗ с помощью которого происходит управление системой приводов робота (с частотой 1000 Гц);
- ПЗ TR_app – ПЗ интерфейса оператора (выполняется в фоновом режиме);
- ПЗ data_trans_tr – ПЗ с помощью которого осуществляется передача данных в ПЗ интерфейса для их отображения в виде осциллограмм на экране ЭВМ оператора (с использованием кольцевых буферов);
- ПЗ back_task_tr – ПЗ в котором решаются задачи верхнего уровня управления роботом;
- ПЗ data_to_file_tr – ПЗ с помощью которого осуществляется запись экспериментальных данных на жёсткий диск ЭВМ;
- ПЗ drive_file – ПЗ с помощью которого осуществляется сохранение настроечных коэффициентов.

4. Заключение.

В заключение хотелось бы отметить об очень полезных моментах некоммерческой образовательной программы проводимой компанией SWD Software, а именно:

- возможность студентов участвовать в крупнейших мероприятиях (выставка ПТА; конференция - QNX-Russia; различные семинары);
- возможность получения оборудования;
- предоставление технической литературы;
- проведение бесплатного обучения преподавателей в сертифицированном Учебном центре компании SWD Software по программам компании QNX Software systems и компании SWD Software;
- публикации в научно-технических журналах, в том числе журналах одобренных ВАК.

Мы благодарим компании SWD Software и QNX Software systems за поддержку и надеемся на дальнейшее продолжение сотрудничества.