

# **МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

## **«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

УТВЕРЖДАЮ

Заместитель директора по  
учебной работе, к.т.н., доцент

\_\_\_\_\_ С.А. Гайворонский

« \_\_ » \_\_\_\_\_ 2013 г.

**М.А. Сонькин, А.А. Шамин**

## **РАСПРЕДЕЛЁННЫЕ МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ**

Методические указания к выполнению курсового проекта  
для магистров II года обучения по направлению  
230100 «Информатика и вычислительная техника»

Издательство  
Томского политехнического университета  
2013

УДК 004.41  
ББК 32.973.2-04  
С00

**Сонькин М.А., Шамин А.А.**

С00      Распределённые микропроцессорные системы: Методические указания к выполнению курсового проекта для магистров II года обучения по направлению 230100 «Информатика и вычислительная техника» / М.А. Сонькин, А.А. Шамин; Национальный исследовательский Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2013. – 40 с.

**УДК 004.41**  
**ББК 32.973.2-04**

Методические указания рассмотрены и рекомендованы  
к изданию методическим семинаром кафедры  
информатики и проектирования систем ИПС  
«\_\_\_»\_\_\_\_\_ 2013 г.

Зам. директора по учебной работе,  
кандидат технических наук, доцент \_\_\_\_\_ С.А. Гайворонский

Председатель учебно-методической  
комиссии,  
доцент кафедры ОСУ ИК ТПУ, к.т.н. \_\_\_\_\_ О.В. Марухина

*Рецензент*

Профессор кафедры КИБЭВС ТУСУР,  
д.т.н, доцент \_\_\_\_\_ Р.В. Мещеряков

© ФГБОУ ВПО НИ ТПУ, 2013  
© Сонькин М.А., Шамин А.А., 2013

## **Список используемых сокращений**

**ТЗ**- техническое задание

**МПС** — микропроцессорная система

**РМПС** — распределённая микропроцессорная система

**МПУ** — микропроцессорное устройство

**ЦЗ** — целевая задача

**КС** — канал связи

**СПД** — системы передачи данных

**АСУ** — автоматизированные и автоматические системы управления

**ССД** — системы сбора данных

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| 1 Цели курсового проекта.....   | 6  |
| 2 Процедура выполнения курсового проекта.....   | 6  |
| 3 Требования к содержанию и оформлению пояснительной записки к курсовому проекту..... | 7  |
| 4 Анализ задания на курсовое проектирование.....                                      | 8  |
| 5 Особенности выполнения курсового проекта в группе.....                              | 9  |
| 6 Типовые задачи, решаемые при проектировании РМПС.....                               | 9  |
| 7 Понятие распределённой микропроцессорной системы РМПС.....                          | 10 |
| 7.1 Программная модель взаимодействия МПУ в составе РМПС.....                         | 11 |
| 7.2 РМПС с точки зрения системотехники.....   | 12 |
| 7.3 РМПС с точки зрения целевой задачи.....   | 13 |
| 7.4 Структура РМПС.....   | 15 |
| 7.5 Надёжность РМПС. Режимы работы с полным и ограниченным набором функций.....       | 16 |
| 8 МПУ в составе РМПС.....   | 17 |
| 9 Каналы связи КС.....  | 18 |
| 9.1 Особенности обмена данными между МПУ в РМПС.....                                  | 19 |
| 9.2 Классификация каналов связи.....  | 19 |
| 10 Программное обеспечение МПУ.....   | 21 |
| 10.1 Модульное построение программного обеспечения МПУ.....                           | 21 |
| 10.2 Межпроцессное взаимодействие.....  | 22 |
| 11 Стенды для выполнения курсового проекта.....                                       | 23 |
| 11.1 Стенд 1. Локальная сеть.....   | 24 |
| 11.2 Стенд 2. Разнородное резервирование.....   | 24 |
| 12 Загрузка и запуск ПО на ВИП-МК.....  | 25 |
| 13 Средства разработки.....   | 27 |
| Список использованных источников.....   | 30 |
| Приложение А. Титульный лист.....   | 33 |

|   |    |
|---|----|
| Приложение Б. Оформление задания.....                   | 34 |
| Приложение В. Варианты заданий к курсовому проекту..... | 36 |

## **1 Цели курсового проекта**

Выполнение курсового проекта преследует следующие цели:

- закрепление и углубление теоретических знаний студентов по дисциплине “Распределённые микропроцессорные системы”;
- обучение студентов использованию приобретённых знаний для решения конкретных прикладных задач построения распределённых микропроцессорных систем;
- обучение студентов навыкам группового проектирования программно-аппаратного обеспечения распределённой микропроцессорной системы;
- изучение архитектуры и средств разработки ПО операционной системы UNIX (Linux);
- расширение навыков работы со специальной литературой и другими современными источниками информации.

## **2 Процедура выполнения курсового проекта**

Независимо от избранной темы рекомендуется придерживаться следующей процедуры выполнения курсового проекта:

- выбор темы курсового проекта. Курсовой проект является самостоятельной творческой работой студентов, поэтому студенты имеют право предложить собственную тему проекта по согласованию с преподавателем;
- согласование задания на курсовое проектирование. На этом этапе студенты согласуют с преподавателем все параметры своего задания, как количественные, так и качественные. Типовые задания на курсовой проект так же предполагают творческий подход и даны в общем виде с целью самостоятельной проработки студентами всех деталей выбранной темы;
- составление технического задания (ТЗ) на курсовой проект согласно форме, приведённой в приложении Б. В техническом задании указываются все количественные и качественные параметры проектируемой системы. Составление ТЗ — очень важный этап выполнения курсового проекта, так как чем более детально проработано ТЗ, тем проще потом его реализовать. Без согласования ТЗ с преподавателем, курсовой проект не принимается;

- распределение задач между участниками проекта. Курсовой проект рассчитан на выполнение группой из 3-4 студентов. После составления ТЗ необходимо закрепить за студентами-участниками проекта решаемые ими задачи. Кроме этого, выбирается координатор проекта, основной задачей которого является общее руководство и обеспечение взаимодействия между всеми участниками проекта. В частности, координатор следит за соблюдением, принятых в группе, соглашений по интерфейсам обмена данными, протоколами межпрограммного и аппаратного взаимодействия;
- разработка алгоритма работы и структуры проектируемой системы. Результатами выполнения данного этапа работ являются: структура взаимодействия аппаратных и программных средств разрабатываемой системы, алгоритм, описывающий работу каждой части разрабатываемой системы по отдельности и взаимодействие этих частей;
- выбор средств разработки для реализации алгоритмов;
- написание программ, реализующих разработанные алгоритмы;
- тестирование системы.

### **3 Требования к содержанию и оформлению пояснительной записки к курсовому проекту**

Пояснительная записка к курсовому проекту должна содержать следующие разделы.

1. Титульный лист (см. приложение А).
2. Задание (см. приложение Б).
3. Введение (краткое описание актуальности разработки, цель и задачи разработки, краткое описание содержания основных разделов пояснительной записки).
4. Описание структуры разрабатываемой РМПС.
5. Обоснование и разработка алгоритмов работы (описание алгоритмов работы всей системы в целом и отдельного МПУ в частности).
6. Выбор средств разработки.
7. Описание разработанных программных средств. (Структура программных средств, описание каждого модуля с обязательным указанием ответственного за его разработку студента-участника проекта).
8. Тестирование системы. (Обоснование набора тестовых заданий и результаты их выполнения).

9. Заключение (краткие выводы по разделам 4-8 и общий вывод по работе).
10. Список использованных источников (в данный список включаются все источники информации, на которые имеются ссылки в курсовом проекте; источники в списке нумеруются в порядке их упоминания в тексте курсового проекта арабскими цифрами без точек).
11. Приложение (исходный текст программы с комментариями, возможно указание ссылки на него).

Пояснительная записка к курсовому проекту должна быть выполнена на белой бумаге формата А4.

#### **4 Анализ задания на курсовое проектирование**

Задание включает в себя разработку программного обеспечения распределённой микропроцессорной системы на основе интеллектуальных микропроцессорных терминалов ВИП-МК [1,16,17,18].

Исходные данные для выполнения проекта:

- 1) Целевые задачи РМПС.
- 2) Структура РМПС, заданная в виде каналов связи, соединяющих МПУ.
- 3) Характеристики каждого типа каналов связи.
- 4) Характеристики МПУ.
- 5) Виды передаваемой информации.
- 6) Для каждого вида передаваемой информации оговариваются требования:
  - 6.1) Необходимость подтверждения о доставке.
  - 6.2) Требования к срокам доставки.
  - 6.3) Объём передаваемой информации данного типа в единицу времени.

Анализ задания на курсовое проектирование должен дать ответ на следующие вопросы:

- Какие основные аппаратные и программные компоненты, необходимые для решения целевой задачи?
- Необходима ли передача информации между несвязными компонентами системы?
- Каким образом строится передача данных между связными и несвязными МПУ?
- Какие ограничения функциональности системы повлечёт за собой выход тех или иных каналов связи?



Следует помнить, что анализ задания — это очень важный этап успешного выполнения проекта, поскольку именно результаты анализа определяют такие важные параметры проекта как то структуру аппаратного обеспечения, структуру программного обеспечения и способы взаимодействия всех компонентов системы.

## **5 Особенности выполнения курсового проекта в группе**

Одной из особенностей группового проектирования является необходимость распределить роли участников проекта.

Самой важной является роль координатора проекта, который в дальнейшем осуществляет постановку задач и следит за тем, чтобы все участники проекта соблюдали требования к аппаратным и программным интерфейсам — использовали одни и те же протоколы обмена данными, выполняли соглашения по использованию общих библиотек и т. п.

Следующим этапом выполнения курсового проекта в группе является распределение решаемых задач между участниками группы.

При распределении задач, координатору необходимо учитывать навыки каждого из участников, чтобы все они были равномерно заняты в проекте.

Так же координатор должен перераспределять задачи между участниками проекта, если возникает такая необходимость. Следует помнить, что трудоёмкость той или иной задачи не всегда возможно достаточно точно оценить заранее. Поэтому, если какая-либо задача требует больше или меньше времени для своего решения, чем было на неё отведено при распределении задач между участниками проекта, то координатор должен перераспределить задачи.

Координатор проекта должен принимать решения о распределении и перераспределении задач только после обсуждения с другими участниками проекта.

В пояснительной записке необходимо указывать — кто из участников проекта отвечает за решение каждой задачи.

## **6 Типовые задачи, решаемые при проектировании РМПС**

Имеется ряд типовых задач, которые приходится решать при разработке РМПС:

- Разработка алгоритма передачи каждого из требуемых типов информации между связными и несвязными МПУ. На основе данных, полученных в ходе анализа задания на проектирование составляется

список всех типов информации, циркулирующих в РМПС и требования к обмену информацией каждого типа. После чего разрабатывается в общем виде алгоритм передачи-приёма для каждого типа информации отдельно.

- Разработка программных компонентов, реализующих передачу информации между МПУ по каждому их заданных каналов связи, выполняющихся на каждом МПУ, входящем в РМПС.

- Разработка программных компонентов, реализующих целевые задачи.

## 7 Понятие распределённой микропроцессорной системы РМПС

Следует отметить, что общепринятого определения распределённой микропроцессорной системы не существует [2], поэтому мы будем исходить из тех определений, которые являются наиболее удобными для наших целей.

Несколько определений, сформулированных в терминах распределённых микропроцессорных систем приведены ниже:

**РМПС** — это МПС, которая приобрела специфику территориально рассредоточенной системы [3];

**РМПС** — это совокупность независимых МПУ, взаимодействующих с целью решения задач, нерешаемых одним МПУ индивидуально [4];

**РМПС** — это совокупность независимых МПУ, представляющаяся пользователям единой системой [2].

Исходя приведённых определений, неотъемлемыми частями любой РМПС являются:

**ЦЗ** - целевые задачи, для решения которых предназначена данная РМПС и которые решаются всей системой в целом, но не её компонентами в отдельности;

**МПУ** - микропроцессорные устройства, каждое из которых взаимодействует с другими МПУ, входящими в РМПС с целью решения определённой задачи, нерешаемой одним МПУ индивидуально;

**КС** - каналы связи, которые обеспечивают обмен информацией между МПУ, входящими в РМПС.

Таким образом, считаем, что мы имеем дело с РМПС, если в неё входят все три компонента: ЦЗ, МПУ и КС.

По назначению к РМПС относятся:

СПД — системы передачи данных;

АСУ — автоматизированные и автоматические системы управления;

ССД — системы сбора данных;  
и т. п.

## 7.1 Программная модель взаимодействия МПУ в составе РМПС

Существует несколько программных моделей, позволяющих рассматривать взаимодействие МПУ в составе РМПС на различных уровнях представления. Наиболее распространённые из указанных моделей — семиуровневая модель OSI [5] и четырёхуровневая модель TCP/IP (DOD) [6]. Программная модель DOD создана для реализации стека протоколов TCP/IP и является по сути упрощённой моделью OSI.

Мы будем рассматривать РМПС исходя из программной модели TCP/IP (DOD), представленной на рис.1, поскольку модель OSI избыточна для наших целей.

Модель DOD состоит из четырёх уровней (сверху вниз):

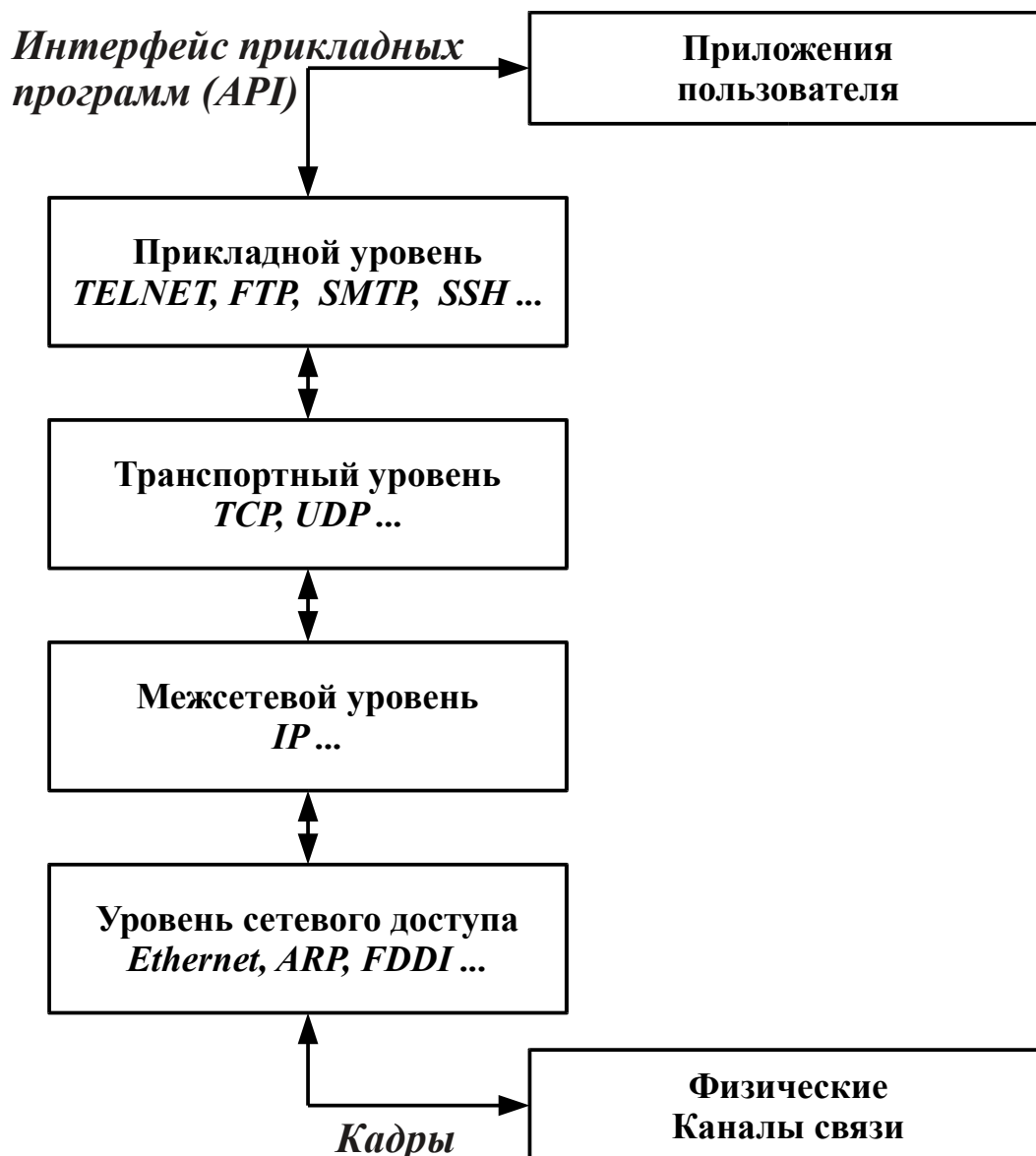
**Прикладного уровня или уровня приложений (англ. Process/Application)**, обеспечивающего программный интерфейс для взаимодействия с приложениями пользователя. На этот же уровень возлагаются задачи преобразования протоколов, шифрования/дешифрования данных, поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время, создание и завершение сеанса связи.

**Транспортного уровня (англ. Transport)** предназначен для обеспечения надёжной передачи данных от отправителя к получателю.

**Межсетевое уровня (англ. Internet)** отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок в сети.

**Уровня сетевого доступа (англ. Network Access)** предназначенного для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками. На данном уровне данные упаковываются в кадры и осуществляется передача-приём кадров по физическим каналам связи, а так же контроль целостности кадров.

Следует помнить, что обе рассмотренные программные модели взаимодействия между МПУ — ISO и DOD — достаточно универсальны, но в то же время не описывают некоторых ограничений, возникающих при проектировании реальных РМПС, например скоростных ограничений физических каналов связи, ограниченного быстродействия МПУ.



**Рис. 1: Программная модель DOD**

Для того, чтобы учесть эти ограничения, необходимо рассмотреть РМПС с точки зрения системотехники, на различных уровнях абстракции.

### **7.2 РМПС с точки зрения системотехники**

С точки зрения системотехники [7], РМПС должна рассматриваться на нескольких уровнях абстракции, последовательно от уровня «чёрный ящик» до структуры программного и аппаратного обеспечения используемых МПУ.

Требуемый уровень абстракции зависит от конкретной ситуации. Например, если даны готовые МПУ с известными характеристиками, то достаточно дойти до уровня «программная модель МПУ», которая описывает как запрограммировать ту или иную функцию данного МПУ известными средствами разработки.

Если МПУ с требуемыми характеристиками не существует — то требуется разработка нового или доработка существующего МПУ до требуемой функциональности. Это приводит к тому, что уровень абстракции понижается до уровня отдельных электронных компонентов и схемотехнических решений.

Для выполнения курсового проекта достаточно уровня абстракции «программная модель МПУ».

На этом уровне МПУ (микропроцессорный терминал ВИП-МК) рассматривается как совокупность устройств, управляемых программой пользователя, работающей под управлением операционной системы Linux [8] дополненной программными средствами для работы по различным каналам связи — радиоканалу, прямому модемному соединению и т.п..

### 7.3 РМПС с точки зрения целевой задачи

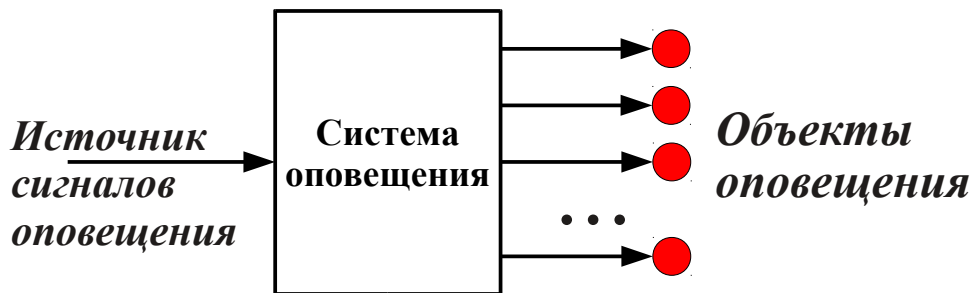
На самом верхнем уровне — уровне целевой задачи — РМПС рассматривается как «чёрный ящик», т. е. некий объект, внутренняя структура которого неизвестна, но известны входные воздействия и ответная реакция на них. Приведём несколько примеров различных РМПС и рассмотрим их с точки зрения целевой задачи.

**Пример 1.** Система оповещения населения о чрезвычайных ситуациях [9,10].

Данная система (рис.2) включает источник сигналов оповещения и объекты оповещения. Задачей системы является доведение сигналов оповещения до объектов оповещения. На целевой задачи определяется:

- максимальное количество сигналов оповещения;
- максимальное количество объектов оповещения;
- способ доведения сигналов - звуковой (сирена, речь и т.п.), визуальный («мигалка», дисплей, бегущая строка и т. п.);
- сроки доведения сигнала от источника до каждого объекта;
- способы подтверждения объектом получения сигнала оповещения;
- дополнительные функции системы, не относящиеся непосредственно к решению основной задачи оповещения;

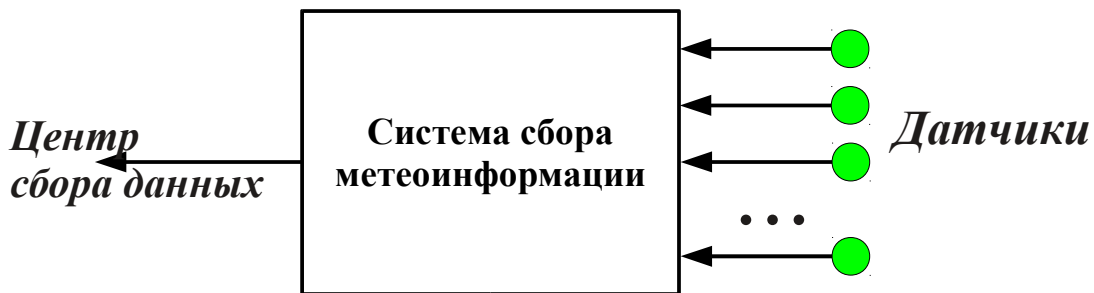
- другие характеристики, важные для функционирования системы в целом.



*Рис. 2: Система оповещения на верхнем уровне представления (чёрный ящик)*

**Пример 2.** Система автоматического сбора метеорологической информации [11,12].

Данная система (рис.3) включает в себя центр сбора данных и датчики, поставляющие метеорологическую информацию.



*Рис. 3: Система сбора метеорологической информации на верхнем уровне представления (чёрный ящик)*

Задачей системы является доставка информации от датчиков в центр сбора данных. На уровне целевой задачи определяется:

- максимальное количество датчиков;
- тип и количество информации для каждого датчика;
- сроки сбора информации (например: непрерывный сбор, период сбора и проч.) для каждого датчика и типа информации;
- способ передачи информации для каждого датчика — по инициативе датчика или центра;

- дополнительные функции системы, не относящиеся непосредственно к решению основной задачи сбора метеорологической информации;
- другие характеристики, важные для функционирования системы в целом.

Исходя из рассмотренного на уровне целевой задачи, можно сделать следующие заключения о работе систем, приведённых в примерах:

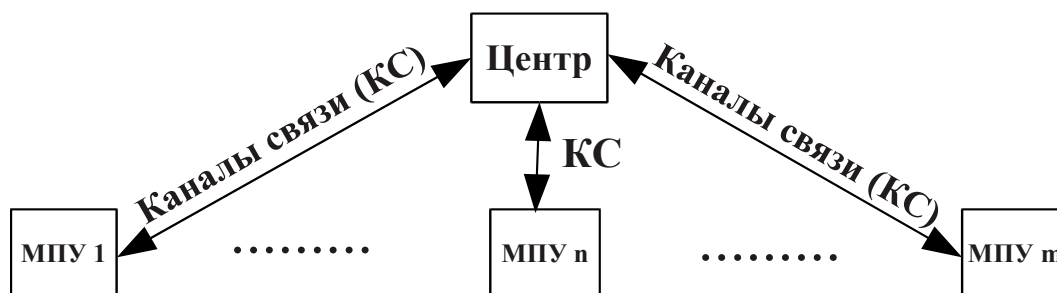
- все, рассмотренные в примерах, РМПС имеют явно выраженную иерархическую структуру — один «центр» и множество подчинённых ему объектов;
- основные потоки данных в системах — противоположны по направлению: в примере 1 основной поток данных идёт от «центра» к подчинённым объектам; в примере 2 основной поток данных идёт напротив — от подчинённых объектов к «центру».

#### 7.4 Структура РМПС

Структура РМПС определяется решаемыми задачами и доступными средствами и является более низким уровнем абстракции, чем рассмотренный ранее «чёрный ящик».

На уровне структуры РМПС в качестве элементов выступают МПУ, а в качестве связующих звеньев системы — каналы связи.

Несмотря на то, что системы в обоих рассмотренных примерах различны по своему назначению, решаемым задачам и направлению основного потока данных, общая структура у обоих систем — одинакова (рис. 4).



*Рис. 4: Общая структура систем из примеров 1 и 2*

Для **примера 1** — системы оповещения — «центр» является источником сигналов оповещения и сбора подтверждения о их

получении, а МПУ — устройства, непосредственно управляющие доведением сигналов — сиренами, дисплеями, громкоговорителями.

Для **примера 2** — системы сбора метеорологической информации — МПУ снимают информации с датчиков и отправляют её по каналам связи в центр сбора данных.

На уровне структуры РМПС производится определение типов данных, циркулирующих по каналам связи и выбор каналов связи по их характеристикам. Определяющим критерием выбора канала связи является тип и количество информации, передаваемой системой по данному каналу связи.

Например, предположим, что в системе оповещения (**пример 1**) имеется требование передавать мультимедийную (звуковую) информацию в реальном времени. Также предположим, что в системе сбора метеорологической информации (**пример 2**) — такого требования нет.

Поэтому, в **примере 1** приходится выбирать более скоростные и дорогие каналы, например ADSL, спутниковые каналы связи, со скоростями не менее 24 кбит/сек. В **примере 2** достаточно каналов со скоростями 50-100 бит/сек (телеграф, радиоканал).

### **7.5 Надёжность РМПС. Режимы работы с полным и ограниченным набором функций**

Как правило, распределённые микропроцессорные системы требуют резервирования каналов связи с целью повышения надёжности доставки данных. Не всегда удаётся обеспечить резервные каналы связи с теми же скоростными характеристиками, как и основные. Поэтому при построении такой системы необходимо, чтобы каждое из МПУ работало в условиях ограниченной функциональности при выходе из строя основного канала связи.

Например, если каждое МПУ в **примере 1** связано с центром двумя КС — высокоскоростным спутниковым каналом и низкоскоростным радиоканалом, то необходимо разработать такой алгоритм работы каждого МПУ и всей системы в целом, чтоб при выходе из строя высокоскоростного спутникового канала — система выполняла основную задачу — доведение сигналов оповещения объектов оповещения.

Таким образом необходимо различать два режима работы РМПС, использующих резервированные каналы связи с различными характеристиками:



- **полнофункциональный режим работы.** Это такой режим в котором РМПС обеспечивает как выполнение ЦЗ, так и всех дополнительных функций, не связанных напрямую с ЦЗ;

- **режим ограниченного функционала.** В данном режиме работы РМПС РМПС обеспечивает выполнение ЦЗ, но не гарантирует выполнения дополнительных функций, не связанных напрямую с ЦЗ.

## 8 МПУ в составе РМПС

**Микропроцессорное устройство (МПУ)** представляет собой функционально и конструктивно законченное изделие, состоящее из нескольких микросхем в состав которых входит микропроцессор; оно предназначено для выполнения определённого набора функций: получение, обработка, передача, преобразование информации и управление.

Применительно к РМПС, МПУ обязательно содержит в своём составе либо средства для передачи данных по каналам связи, либо интерфейсы для подключения таких средств (например — модемов, оптоволоконных линий, спутниковых терминалов и проч.).

Как правило, МПУ в составе РМПС должно сохранять ограниченную функциональность при отсутствии соединений по каналам связи.

Например, в системах сбора данных и системах документированной связи [13], а так же в большинстве иных РМПС существуют требования доставки всех введённых в МПУ данных без потерь. Но практически в любой РМПС имеются периоды времени, когда связь с центром и всеми другими МПУ на некоторое время прерывается по тем или иным причинам.

С целью выполнения требования доставки всех введённых в МПУ данных без потерь, обычно применяется достаточно простой алгоритм: в отсутствие связи по всем доступным каналам, МПУ запоминает сформированные сообщения и хранит их до появления связи, а затем передаёт их по назначению.

Отметим, что данный алгоритм накладывает на протокол передачи сообщений требование о наличии в каждом сообщении времени его формирования. Иначе «центру» невозможно будет узнать какому моменту времени какое сообщение соответствует. Кроме этого, очевидно, что все МПУ, входящие в состав РМПС должны быть синхронизированы по времени.

Итак, сформулируем общие требования, предъявляемые к МПУ в составе РМПС:

- МПУ должно содержать в своём составе либо средства для передачи данных по каналам связи, либо интерфейсы для подключения таких средств;

- МПУ должно обеспечивать ограниченную функциональность в отсутствие связи с другими МПУ;

- все МПУ в составе РМПС должны иметь часы, синхронизированные по времени. Причём часы должны корректно отсчитывать время и при отсутствии связи с «центром» и другими МПУ данной системы;

- иные требования к МПУ, накладываемые решаемыми задачами. Основные требования накладывает ЦЗ. Но кроме этого, возможны и иные требования к МПУ, не связанные с ЦЗ, но значительно повышающие функциональные возможности МПУ — например, комфорт оператора при работе с МПУ, удобство диагностирования неполадок в МПУ и т.п.

## 9 Каналы связи КС

**Канал связи** [14]— система технических средств и среда распространения сигналов для передачи сообщений от источника к получателю (и наоборот). Применительно к РМПС под каналом связи (**КС**) понимают **только** цифровые каналы связи, то есть такие, которые на входе и выходе используют двоичный цифровой код. Преобразование цифрового кода в аналоговый сигнал, передача-приём его по физическим линиям, преобразование форматов и прочее — не рассматривается. С точки зрения МПУ — вход и выход любого канала связи — цифровой.

Для обеспечения передачи цифровой информации в виде аналогового сигнала по различным физическим средам, как правило, используются различные **модемы** — телефонные, сотовые, спутниковые, радиомодемы и проч. **Модем** — устройство, применяющееся в системах связи для физического сопряжения информационного сигнала со средой его распространения, где он не может существовать без адаптации.

Любой модем, применяемый в составе (или подключаемый) к МПУ имеет цифровой интерфейс для сопряжения с МПУ (обычно, RS232, Ethernet или SPI) и аналоговый интерфейс для сопряжения со средой передачи сигнала (антенна для радио-, сотовых и спутниковых модемов, разъём для подключения к телефонной линии для телефонных модемов и т. п.).

Со стороны МПУ — модем есть устройство, которое передаёт и принимает цифровые сообщения. В общем случае модем **не гарантирует** целостность пакетов данных, передаваемых и принимаемых по каналу связи, хотя большинство современных модемов имеют средства обнаружения и устранения ошибок, автоматического установления соединения и проч.

### **9.1 Особенности обмена данными между МПУ в РМПС**

В общем случае, МПУ должно обеспечивать контроль целостности принимаемых пакетов и подтверждение об их получении.

Чаще всего, МПУ работают по алгоритму - «посылка-подтверждение». То есть на каждое отправленное МПУ-передатчиком сообщение, МПУ-приёмник должен ответить подтверждением. Это гарантирует, что передатчик информирован о том, дошло его сообщение до приёмника или нет. Особенная ситуация возникает, в случае, когда сообщение доставлено успешно, а подтверждение не доставлено. В этом случае передатчик считает, что сообщение не доставлено и обычно повторяет его передачу. Это может вызвать, например, выполнение двух одинаковых команд, вместо одной и последствия для работы системы станут непредсказуемыми.

Отсюда следует, что каждое сообщение должно иметь некий уникальный идентификатор, позволяющий приёмнику сообщений знать — обрабатывал он данное сообщение или нет с целью избежать повторной обработки.

Таким образом, ненадёжность каналов связи накладывает свои требования на протокол передачи данных:

- МПУ в РМПС должны обмениваться данными по алгоритму «посылка-подтверждение»;
- каждый пакет должен иметь уникальный идентификатор, чтобы избежать повторной обработки.

### **9.2 Классификация каналов связи**

Каналы связи различны по своим свойствам. Различны их скоростные характеристики. Различны алгоритмы передачи данных по каналам связи различной природы. Поэтому, выбор каналов связи и оценка пригодности их для решения задач, поставленных перед РМПС — важный этап проектирования РМПС и МПУ. Один из вариантов классификации КС, показан на рис.5.



*Рис. 5: Классификация каналов связи*

Классификация произведена по нескольким параметрам — скоростным характеристикам, способу установления соединения и способу обмена данными.

**Скоростные характеристики.**

С точки зрения скоростных характеристик, КС подразделяются на высокоскоростные и низкоскоростные.

Считаем, что **высокоскоростные КС** — это КС, позволяющие реализовать полнофункциональный режим работы РМПС.

**Низкоскоростные КС** — те КС, которые позволяют реализовать функции РМПС в режиме ограниченного функционала.

Таким образом, понятия «низкоскоростной» и «высокоскоростной» канал связи — зависят от конкретной ЦЗ и требований к конкретной РМПС.

КС, не обеспечивающие работу конкретной РМПС даже в режиме ограниченного функционала считаем непригодными для использования и не рассматриваем.

**Способ установки соединения.**

По способу установки соединения КС классифицируются следующим образом:

- КС с постоянным соединением. Это однократно устанавливаемое и постоянно контролируемое соединение. При обнаружении разрыва — производится восстановление соединения. Постоянное соединение характерно для каналов связи, использующих протокол TCP/IP;

- КС с соединением на время сеанса связи. В этом случае при передаче сообщения производится процедура установления соединения, затем обмен данными, после чего соединение разрывается. Соединение на время сеанса характерно для прямых модемных соединений (PSTN, GSM);

- КС без установления соединения. Обмен данными производится сразу, без процедуры установки соединения. Такой способ характерен, например, для радиоканала.

## **10 Программное обеспечение МПУ**

Выделяется ряд общих программных компонентов, присутствующих в любом ПО, которое исполняется на МПУ в составе РМПС.

Прежде всего — это ПО, обеспечивающее обмен данными между МПУ. Функции данного ПО — реализация алгоритмов обмена данными по каждому из КС, подключённому к МПУ.

Другая составляющая ПО — это программные модули, реализующие целевую функцию.

При разработке ПО, студентам-участникам проекта, следует определить следующее:

- а) общий алгоритм работы всей системы в целом и каждого МПУ в отдельности. Алгоритм составляется на основании задания на проектирование;
- б) функции и алгоритм работы МПУ в составе РМПС. Функции МПУ обосновываются исходя из задания на проектирование. Алгоритм работы каждого МПУ описывается исходя из реализуемых им функций и алгоритма работы системы в целом.
- в) функции и алгоритм работы каждого программного модуля;
- г) способ реализации программных модулей;
- д) механизм взаимодействия между различными программными модулями.

### **10.1 Модульное построение программного обеспечения МПУ**

Создание программного обеспечения для МПУ в составе РМПС представляет собой сложную задачу, которую можно удовлетворительно решить только используя современные подходы к созданию ПО.

В настоящее время целесообразным считается разбиение программного обеспечения на модули, каждый из которых реализует задачу, достаточно простую для кодирования и тестирования.

Программное обеспечение, разрабатываемое в настоящем курсовом проекте, выполняется под ОС Linux, имеющей широкий набор средств проектирования и реализации ПО.

Прежде всего, следует определиться что, с точки зрения ОС, будут представлять собой программные модули, разрабатываемые каждым

участником проекта: модули-процедуры, модули-классы, модули-процессы или модули-потoki. Возможно, что часть участников будет разрабатывать программные модули одного типа, а часть — другого.

Поскольку разработка ведётся под ОС Linux, то наиболее рациональным представляется, что каждый участник проекта разрабатывает модули в виде отдельных процессов, реализующих взаимодействие одним из возможных в ОС Linux способов. Но по согласованию с руководителем, организация ПО может быть изменена.

Независимо от решения необходимо чётко и однозначно описать интерфейс каждого программного модуля, а именно: способ взаимодействия с другими программными модулями, входные и выходные данные, поведение (алгоритм работы) модуля.

В отчёте по курсовому проектированию **обязательно** должно содержаться:

- обоснование типа выбранной модульной структуры;
- описание способа взаимодействия между программными модулями;
- документация на каждый разработанный программный модуль.

В документации на программный модуль должны указываться следующие сведения:

- автор программного модуля (один из участников проекта);
- цель разработки данного программного модуля;
- алгоритм работы модуля;
- входные и выходные данные модуля (описание структур данных, способов их передачи модулю, способы получения выходных данных и проч.).

## 10.2 Межпроцессное взаимодействие

Достоинства организации ПО МПУ, при которой каждый законченный программный модуль является отдельным процессом очевидны:

во-первых — это удобство отладки модулей по отдельности. Для отладки нет необходимости дожидаться пока будут закончены все программные модули, а достаточно лишь симитировать входные данные конкретного модуля и получить результат в виде выходных данных;

во-вторых, при отладке системы целиком, гораздо проще определить причину краха достаточно простой программы-модуля, чем искать эту причину в монолитной большой программе;

в-третьих, в случае какой-либо модификации системы, нет необходимости переписывать и компилировать всю систему целиком, а достаточно лишь заменить отдельные один или несколько модулей.

В ОС Linux существует широкий выбор способов взаимодействия между процессами [19]:

- полудуплексные каналы;
- FIFO (именованные каналы);
- очереди сообщений в стиле;
- множества семафоров в стиле;
- разделяемые сегменты памяти в стиле;
- сетевые сокеты;
- полнодуплексные каналы (каналы потоков).

Выбор наиболее подходящих способов межпрограммного взаимодействия в проектируемом ПО требует от студентов изучения дополнительных источников, например [19,20].

## **11 Стенды для выполнения курсового проекта**

Для выполнения курсового проекта используются стенды с различной конфигурацией МПУ и подключённой к ним периферии.

В качестве МПУ для построения РМПС используются интеллектуальные микропроцессорные терминалы ВИП-МК и персональные компьютеры (ПК).

Базовым элементом стенда является многофункциональный микропроцессорный терминал ВИП-МК, построенный на базе микропроцессора с архитектурой ARM9T и имеющего различные интерфейсы ввода-вывода информации — клавиатуру PS/2, дисплей 2x40 символов, последовательные порты, USB, интерфейс Ethernet для подключения по локальной сети.

Терминалы ВИП-МК работают под управлением ОС Linux версии не менее 2.6.32, что облегчает написание и отладку программ пользователя.

Подробную информацию об аппаратных и программных средствах ВИП-МК можно найти в [1,16,17,18].

Каждый стенд разрабатывается с учётом задания. Примеры построения стендов, построенных на базе терминалов ВИП-МК приведены ниже.

### 11.1 Стенд 1. Локальная сеть.

На рис.6 представлена схема наиболее простого стенда для моделирования РМПС. Данный стенд имеет только один физический канал связи между МПУ — локальную сеть Ethernet.

На данном стенде можно реализовать различные протоколы обмена данными между МПУ, моделировать передачу различных, в том числе и мультимедийных (например звуковых) данных в реальном времени. ОС Linux позволяет создать несколько независимых виртуальных сетевых интерфейсов, что позволяет моделировать также многоканальные системы, используя один физический Ethernet-канал.



**Рис. 6: Стенд 1**

Персональный компьютер ПК служит для оперативного мониторинга состояния МПУ ВИП-МК по локальной сети.

При моделировании иерархических систем, как правило, удобно на ПК запускать ПО центрального МПУ.

### 11.2 Стенд 2. Разнородное резервирование.

На рис.7 изображён стенд с двумя разнородными каналами связи — основным и резервным.

Данный стенд позволяет моделировать особенности поведения РМПС при наличии двух разнородных каналов связи — высокоскоростного Ethernet и низкоскоростного радиоканала.



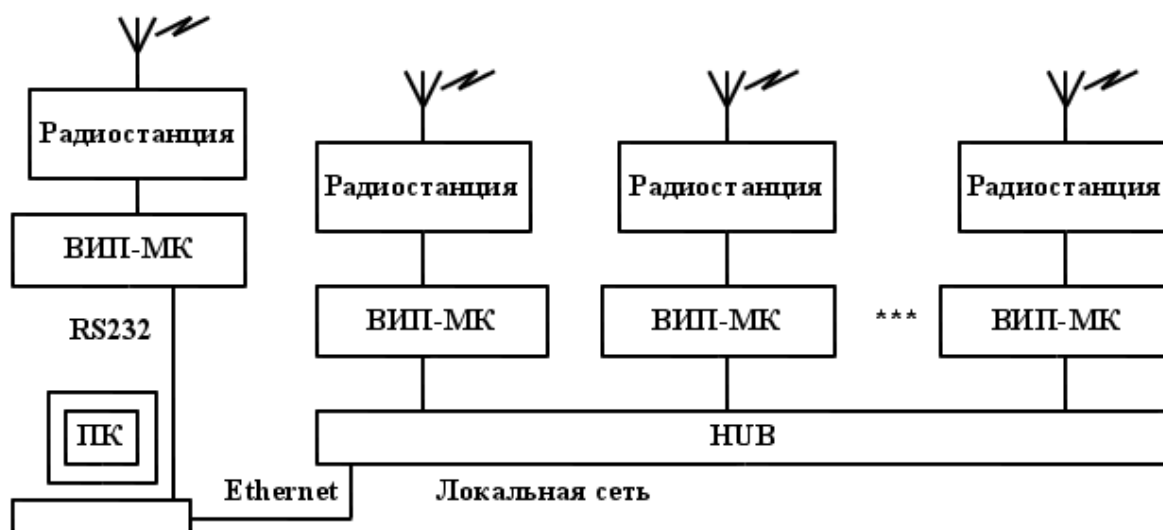


Рис. 7: Стенд 2

## 12 Загрузка и запуск ПО на ВИП-МК

Системное ПО ВИП-МК имеет в своём составе **ssh**-сервер (реализация *dropbear*), позволяющий по локальной сети подключаться к ВИП-МК, используя в качестве терминала персональный компьютер.

**SSH** - это протокол для удаленного доступа на другой компьютер через сеть, удаленного исполнения команд, например копирование файлов между компьютерами, а так же другие задачи. "SSH" расшифровывается как "Secure Shell". SSH обеспечивает надежную авторизацию и безопасную передачу данных по открытым каналам связи [21].

Для загрузки и запуска отлаживаемого ПО на ВИП-МК необходим персональный компьютер, имеющий интерфейс Ethernet и установленную программу, называемую **ssh**-клиент (программа **ssh** или, например, **putty**).

Для доступа на ВИП-МК необходимо знать IP-адрес ВИП-МК, имя пользователя, под правами которого осуществляется доступ и пароль.

**Пример 1.** Выполнение команд на терминале ВИП-МК с использованием программы **ssh**.

Программа **ssh** входит в пакет программ с аналогичным названием и предназначена для организации удалённого доступа на другую систему. Программа **ssh** предоставляет пользователю интерфейс командной строки, позволяющий выполнять команды и программы на удалённой системе.

Предположим, что IP-адрес ВИП-МК *192.168.3.1* и необходимо зайти с правами пользователя *root*. В общем виде команда **ssh** имеет формат **ssh <имя>@<адрес>**, где **имя** — имя пользователя, **адрес** — адрес удалённого ssh-сервера. Тогда для доступа достаточно набрать команду *ssh root@192.168.3.1*, где *root* — имя пользователя:

```
student@localhost$ ssh root@192.168.3.1
```

Если адрес указан верно и на ВИП-МК запущен **ssh**-сервер, то система запросит пароль:

```
root@87.251.123.34's password:
```

При вводе пароля он никак не отображается на экране терминала. Если пароль верен, то произойдёт вход в систему ВИП-МК и на экран терминала выведется подсказка оболочки терминала ВИП-МК:

```
localhost ~ #
```

Далее можно вводить любые доступные команды, они будут выполняться на терминале ВИП-МК. Предположим, что на ВИП-МК имеется программа *hello*, находящаяся в каталоге */opt/bin* и выводящая в стандартный поток вывода фразу «Hello world!». Для запуска программы *hello* достаточно ввести:

```
localhost ~ # /opt/bin/hello
```

```
Hello world!
```

Для завершения сеанса работы достаточно выйти из оболочки командой *exit*:

```
localhost ~ # exit
```

После выхода из оболочки ВИП-МК, произойдёт завершение ssh-соединения.

Более подробную информацию по команде **ssh** можно получить, набрав команду **man ssh** в терминале ОС Linux.

**Пример 2.** Копирование файлов с использованием протокола **SSH** (команда **scp**).

Для копирования файлов между ВИП-МК и персональным компьютером используется команда **scp**, входящая в пакет программ **ssh**. Для использования этой команды необходимо знать IP адрес ВИП-МК

имя пользователя, под правами которого осуществляется доступ и пароль.

Например, имеется программа `hello`, скомпилированная для ВИП-МК и находящаяся в каталоге `/home/student/work/` и её необходимо скопировать на ВИП-МК в каталог `/opt/bin/` для тестирования.

Предположим, что IP-адрес ВИП-МК **192.168.3.1** и необходимо зайти с правами пользователя **root**. В общем виде команда **scp** имеет формат **scp <источник> <приёмник>**, где источник и приёмник могут быть как на локальной так и на удалённой машине.

Для этого необходимо ввести следующую команду:

```
student@localhost$ scp /home/student/work/hello \  
root@192.168.3.1:/opt/bin/
```

Если адрес указан верно и на ВИП-МК запущен **ssh**-сервер, то система запросит пароль:

```
root@87.251.123.34's password:
```

При вводе пароля он никак не отображается на экране терминала. Если пароль верен, то произойдёт копирование файла.

Более подробную информацию по команде **scp** можно получить, набрав команду **man scp** в терминале.

### 13 Средства разработки

В качестве средств разработки для ВИП-МК рекомендуется применять кросс-компиляторы языков C и C++ - **arm-incom-linux-gnueabi-gcc** и **arm-incom-linux-gnueabi-g++**.

Кросс-компиляторы — компилятор, производящий исполняемый код для платформы, отличной от той, на которой исполняется сам кросс-компилятор. В данном случае — сам компилятор исполняется на платформе x86 или x86\_64 (персональный компьютер), а исполняемый код, полученный в процессе компиляции — на платформе ARM9T (ВИП-МК). Подробная информация о применении компилятора `gcc/g++` приведена в [22], краткие примеры программ с описанием — в [23].

Так же настоятельно рекомендуется ознакомиться с утилитой **make**, позволяющей автоматизировать сборку программ. Примеры использования утилиты **make** имеются в [24], а подробное описание (оригинал) — в [25].

Предположим, имеется программа, состоящая из одного файла с исходными текстами на языке C — **prog.c**.

Для получения исполняемого файла необходимо выполнить команду:

```
localhost ~ # arm-incom-linux-gnueabi-gcc prog.c -o prog
```

Данная команда откомпилирует файл **prog.c**, результатом чего будет исполняемый файл **prog** в формате ARM9.

Для программ на языке C++ компиляция выполняется аналогично:

```
localhost ~ # arm-incom-linux-gnueabi-g++ prog.cxx -o prog
```

Если программа состоит из нескольких файлов с исходным текстом, то применяется отдельная компиляция. Т.е. каждый файл с исходным текстом преобразуется в объектный файл с расширением **\*.o**, а затем все объектные файлы преобразуются в единый исполняемый файл (линкуются).

Предположим, имеется программа, состоящая из трёх файлов с исходными текстами на языке C — **prog.c**, **mod1.c** и **mod2.c**. Необходимо получить исполняемый файл **myprog**.

Компилируем исходный код в объектные файлы:

```
localhost ~ # arm-incom-linux-gnueabi-gcc -c prog.c
```

```
localhost ~ # arm-incom-linux-gnueabi-gcc -c mod1.c
```

```
localhost ~ # arm-incom-linux-gnueabi-gcc -c mod2.c
```

Результатом являются объектные файлы **prog.o**, **mod1.o** и **mod2.o**.

Линкуем объектные файлы **prog.o**, **mod1.o** и **mod2.o** в исполняемый файл **myprog**:

```
localhost ~ # arm-incom-linux-gnueabi-gc prog.o, mod1.o и mod2.o  
-o myprog
```

Раздельная компиляция программ на языке C++ осуществляется полностью аналогично, но используется компилятор **arm-incom-linux-gnueabi-g++**.

Более подробную информацию о компиляции программ в ОС Linux можно получить в [20,21,23,24].

## Список использованных источников

1. Краткое описание структуры аппаратного обеспечения ВИП-МК. [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
2. Таненбаум Э., Стеен М. Распределенные системы. Принципы и парадигмы. - СПб.: Питер, 2003. - 877 с.: ил.
3. Олзоева С.И. Распределенное моделирование в задачах разработки АСУ. – Улан-Удэ: изд-во ВСГТУ, 2005. – 219 с.
4. Kshemkalyani D., Singhal M. Distributed Computing: Principles, Algorithms, and Systems. - New York: Cambridge University Press, 2008. - 756 с.
5. IEEE STANDARDS ASSOCIATION. [Электронный ресурс].- Режим доступа: <http://standards.ieee.org/getieee802/portfolio.html> свободный. – Загл. с экрана.
6. Модель DOD. [Электронный ресурс].- Режим доступа: [http://ru.wikipedia.org/wiki/Модель\\_DOD](http://ru.wikipedia.org/wiki/Модель_DOD) свободный. – Загл. с экрана.
7. Николаев, В.И. Системотехника: методы и приложения / В.И. Николаев, В.М. Брук. – Л.: Машиностроение, Ленингр. отд-ние, 1985. – 199 с.
8. Linux. [Электронный ресурс].- Режим доступа: <https://ru.wikipedia.org/wiki/Linux> свободный. – Загл. с экрана.
9. В.В. Гринемаер, А.А. Шамин Технические средства П-166 ИТК ОС . - Приборы №12 2012.
10. В.В. Гринемаер, А.А. Шамин Мониторинг работоспособности системы оповещения и контроль параметров удаленных объектов. - Приборы №12 2012.
11. Сонькин М.А., Гринемаер В.В., Печерская Е.И., Лиханов М.Г., Киреев М.В., Диденко С.В. Опыт создания интегрированной системы сбора метеоданных с сети труднодоступных станций на основе спутниковых и радиоканалов // Сборник «Кибернетика и вуз» Выпуск № 30. Издательство ТПУ. с. 87-95.

12. Багдасарова Е.П. Применение современных технологий сбора данных с наблюдательной сети // Метеоспектроскопия. – 2005. – № 2. – С. 89–93.
13. Бунин С.В., Сонькин М.А., Харламов А.М., Ямпольский В.З. Новые функциональные возможности системы оповещения и резервной документированной связи внутренних войск МВД России. - Известия ТПУ 2008 Том: 312, выпуск 2.
14. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. — Пер. с англ. — М.: Издательский дом «Вильямс», 2003. — 1104 с. — ISBN 5-8459-0497-8
15. Структура и функции ПО ВИП-МК. Краткое описание. [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
16. Файловая система ВИП-МК [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
17. Последовательные порты ВИП-МК [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
18. Терминал ВИП-МК. Справочные данные [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
19. Эрик Рэймонд. Искусство программирования для Unix. - Вильямс, 2005 // ISBN: 5-8459-0791-8, 0-13-142901-9
20. Стивене Р., Раго С. UNIX. Профессиональное программирование, 2-е издание. - СПб.: Символ-Плюс, 2007. - 1040 с, ил. ISBN 5-93286-089-8
21. Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes. SSH, The Secure Shell: The Definitive Guide O'Reilly Media // ISBN 0596008953; 2005 г.
22. Гриффитс Артур. GCC. Настольная книга пользователей, программистов и системных администраторов / К.: ООО ТИД "ДС", 2004, 624с.

23. Знакомство с компилятором GCC [Электронный ресурс] .- Режим доступа:  
<http://www.linuxcenter.ru/lib/books/linuxdev/linuxdev1.phtml>  
свободный. – Загл. с экрана.
24. Компилятор G++ [Электронный ресурс] .- Режим доступа:  
<http://www.linuxcenter.ru/lib/books/linuxdev/linuxdev4.phtml>  
свободный. – Загл. с экрана.
25. Managing Projects with GNU Make, 3.Xth Edition / O'Reilly & Associates, Inc., 2004 [Электронный ресурс] .- Режим доступа:  
<http://oreilly.com/catalog/make3/book/> свободный. – Загл. с экрана.



## Приложение А. Титульный лист

### МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

### «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

---

Институт кибернетики (ИК)

Кафедра информатики и проектирования систем (ИПС)

### ТЕМА КУРСОВОГО ПРОЕКТА

Пояснительная записка к курсовому проекту по дисциплине  
"Распределённые микропроцессорные системы"

Студенты гр. \_\_\_\_\_

А.Б.Иванов "\_\_\_" \_\_\_\_\_ 200\_ г.

Студенты гр. \_\_\_\_\_

В.Г.Петров "\_\_\_" \_\_\_\_\_ 200\_ г.

Руководитель: ассистент .каф.  
ИПС, канд.техн.наук

А.А. Шамин "\_\_\_" \_\_\_\_\_ 200\_ г.

2013

## Приложение Б. Оформление задания

### МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

### «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

---

Институт кибернетики (ИК)

Кафедра информатики и проектирования систем (ИПС)

#### ЗАДАНИЕ

на курсовое проектирование студенту Иванову Александру  
Борисовичу  
группа 8М82

- 1 Тема проекта: \_\_\_\_
- 2 Срок сдачи студентом законченного проекта \_\_\_\_\_
- 3 Исходные данные к проекту:
  - тема проекта
  - учебно–методическая литература;
  - требования к содержанию и оформлению проекта.
- 4 Техническое задание на курсовой проект.
- 5 Содержание пояснительной записки (перечень подлежащих разработке вопросов).
  1. Введение.
  2. Обоснование и разработка алгоритмов работы.

3. Распределение задач между участниками проекта.
4. Выбор средств разработки.
5. Описание разработанных программных средств.
6. Тестирование системы.
7. Заключение (краткие выводы по разделам и общий вывод по работе).
8. Список использованных источников.
9. Приложение (исходный текст программы с комментариями, возможно указание ссылки на него, возможно в электронном виде).

Задание приняли к исполнению

---

(фамилия, имя, отчество)

---

дата

---

подпись студента

---

(фамилия, имя, отчество)

---

дата

---

подпись студента

## Приложение В. Варианты заданий к курсовому проекту

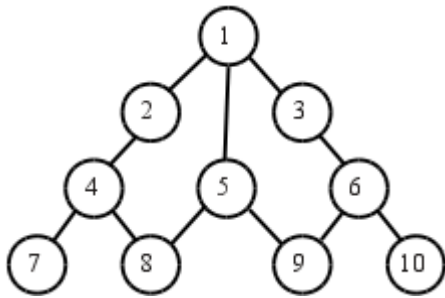
**Задача курсового проекта** — организовать передачу данных от любого из абонентов к любому в рамках заданной схемы. Варианты заданий курсового проекта приведены в таблице 1.

Таблица 1: Варианты заданий курсового проекта

| №  | Схема связей<br>(см. рис. 8-12) | Кол-во КС | КС1                               | КС2                  | КС3                              | КС4                  |
|----|---------------------------------|-----------|-----------------------------------|----------------------|----------------------------------|----------------------|
| 1  | 1                               | 2         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    |                                  |                      |
| 2  | 2                               | 3         | Прямое соединение, 115200 бит/сек | Телеграф, 50 бит/сек | Интерент, 9600 бит/сек           |                      |
| 3  | 3                               | 4         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    | Прямое соединение, 19200 бит/сек | Телеграф, 50 бит/сек |
| 4  | 4                               | 2         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    |                                  |                      |
| 5  | 5                               | 3         | Прямое соединение, 115200 бит/сек | Телеграф, 50 бит/сек | Интерент, 9600 бит/сек           |                      |
| 6  | 1                               | 4         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    | Прямое соединение, 19200 бит/сек | Телеграф, 50 бит/сек |
| 7  | 2                               | 2         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    |                                  |                      |
| 8  | 3                               | 3         | Прямое соединение, 115200 бит/сек | Телеграф, 50 бит/сек | Интерент, 9600 бит/сек           |                      |
| 9  | 4                               | 4         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    | Прямое соединение, 19200 бит/сек | Телеграф, 50 бит/сек |
| 10 | 5                               | 2         | Интерент, 115200 бит/сек          | Радио, 50 бит/сек    |                                  |                      |

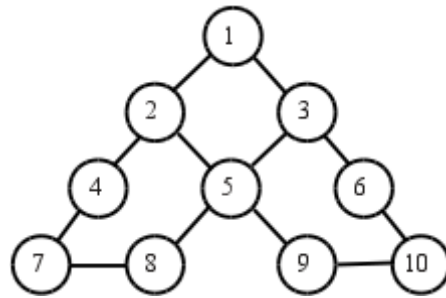
В ходе выполнения курсового проекта следует:

1. Распределить задачи и роли между участниками проекта.
2. Провести анализ схемы связей (топологии) предложенной РМПС.
3. Построить модель системы с заданной схемой связей и заданными каналами связи.
4. Построить алгоритм, определения того, возможно ли в данный момент передать информацию от одного устройства к другому, основываясь на построенной модели, типе передаваемой информации и текущем состоянии каналов связи.



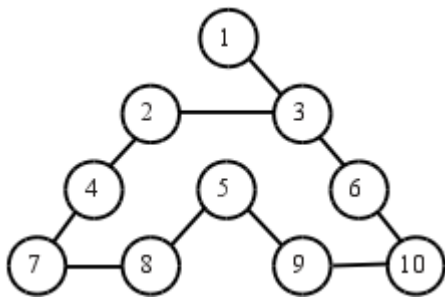
*Рис. 8: Схема связей*

*РМПС, вар. 1*



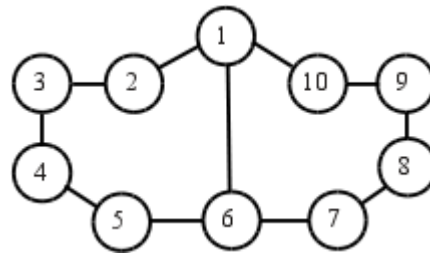
*Рис. 9: Схема связей*

*РМПС, вар. 2*



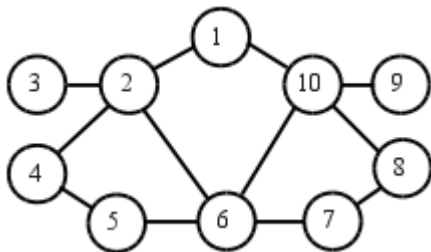
*Рис. 10: Схема связей*

*РМПС, вар. 3*



*Рис. 11: Схема связей*

*РМПС, вар. 4*



*Рис. 12: Схема связей*

*РМПС, вар. 5*

5. Построить алгоритм передачи звуковых сообщений между двумя произвольными устройствами РМПС.
6. Построить алгоритм передачи информационных сообщений и подтверждений между двумя произвольными устройствами РМПС.

7. Разработать программное обеспечение, реализующее построенные алгоритмы.

### **Схема связей РМПС.**

На каждой схеме представлена РМПС, имеющая 10 устройств-абонентов. Каждая линия представляет собой каналы связи между устройствами. Количество каналов и их тип определяется заданием.

### **Обмен данными.**

Считаем, что устройства обмениваются между собой данными двух типов:

1. Звуковыми сообщениями в реальном времени — постоянный поток 2150байт/сек, разбитые на пакеты по 43 байт без подтверждения.
2. Информационные сообщения длиной до 2 Кбайт, некритичные ко времени доставки. На каждое сообщение должно иметься подтверждение.

### **Каналы связи.**

Каждый из каналов связи может быть в двух состояниях — рабочем и неисправном. В неисправном состоянии — канал связи фактически отсутствует.

Краткая характеристика каналов связи:

**Интернет** — поддержка множественного соединения. То есть обмен информацией может осуществляться одновременно с несколькими устройствами. Однако, скорость обмена падает пропорционально кол-ву устройств, с которыми ведётся обмен данными.

**Прямое соединение** — обеспечивает соединение точка-точка. То есть одновременно по такому каналу связи могут обмениваться данными только два устройства.

**Радио** — отличается низкой скоростью передачи данных. Работает по принципу - «точка-многоточка» или «один передаёт — все принимают». То есть любая передача по радиоканалу может быть принята любым количеством устройств. Но в каждый конкретный момент времени передавать может только одно из устройств.

**Телеграф** — прямое соединение с низкой скоростью.

### **Дополнительные требования.**

Дополнительные требования к разрабатываемой РМПС помимо, указанных в таблице 1, согласовываются с преподавателем.

### **Самостоятельный выбор задания.**

В случае, если группа студентов предлагает свой вариант задания, отличный от типового, оно должно быть чётко изложено в **письменном** виде и предоставлено на согласование преподавателю.

Учебное издание

СОНЬКИН Михаил Аркадьевич  
ШАМИН Алексей Алексеевич

## РАСПРЕДЕЛЁННЫЕ МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

Методические указания к выполнению курсового проекта для  
магистров II года обучения по направлению  
230100 «Информатика и вычислительная техника»


**Отпечатано в Издательстве ТПУ в полном  
соответствии  
с качеством предоставленного оригинал-макет**

Подписано к печати [REDACTED]. Формат 60x84/16. Бумага «Снегурочка».  
Печать XEROX. Усл.печ.л. 9,01. Уч.-изд.л. 8,16.  
Заказ . Тираж 100 экз.



Национальный исследовательский Томский политехнический университет  
Система менеджмента качества  
Издательства Томского политехнического университета сертифицирована  
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008



ИЗДАТЕЛЬСТВО  ТПУ . 634050, г. Томск, пр. Ленина, 30  
Тел./факс: 8(3822)56-35-35, [www.tpu.ru](http://www.tpu.ru)