

# МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

## «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

---

УТВЕРЖДАЮ

Зам. проректора-директора ИК по  
учебной работе, к.т.н., доцент

\_\_\_\_\_ С.А. Гайворонский

« \_\_ » \_\_\_\_\_ 2012 г.

**М.А. Сонькин, А.А. Шамин**

## МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

Методические указания к выполнению курсового проекта  
для студентов IV курса, обучающихся по направлению 230100  
«Информатика и вычислительная техника»

Издательство  
Томского политехнического университета  
2012

УДК 004.41  
ББК 32.973.2-04  
С00

**Сонькин М.А., Шамин А.А.**

С00 Микропроцессорные системы: Методические указания к выполнению курсового проекта для студентов IV курса, обучающихся по направлению 230100 «Информатика и вычислительная техника» / М.А. Сонькин, А.А. Шамин; Национальный исследовательский Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2012. – 29 с.

**УДК 004.41**  
**ББК 32.973.2-04**

Методические указания рассмотрены и рекомендованы  
к изданию методическим семинаром кафедры  
информатики и проектирования систем ИПС  
«\_\_»\_\_\_\_\_ 2012 г.

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

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

*Рецензент*

Доцент кафедры ОСУ ИК ТПУ, к.т.н. \_\_\_\_\_ Е.С. Чердынцев

© ФГБОУ ВПО НИ ТПУ, 2012  
© Сонькин М.А., Шамин А.А., 2012

## СОДЕРЖАНИЕ

1 Цели курсового проекта .....	4
2 Процедура выполнения курсового проекта .....	4
3 Требования к содержанию и оформлению пояснительной записки к курсовому проекту .....	5
4 Стенды для отладки микропроцессорной системы на базе ВИП-МК .5	
4.1 Многофункциональный микропроцессорный терминал ВИП-МК...6	
4.2 Построение стенда.....	6
4.3 Стенд для отладки не-коммуникационного ПО .....	6
4.4 Стенд для отладки ПО, использующего Ethernet для коммуникации между несколькими ВИП-МК .....	7
5 Загрузка и запуск ПО на ВИП-МК.....	8
6 Средства разработки .....	10
Список использованных источников .....	13
Приложение А. Титульный лист.....	14
Приложение Б. Оформление задания .....	15
Приложение В. Пример курсового проекта .....	17
Приложение Б. Календарный рейтинг-план выполнения курсового проекта .....	28

## **1 Цели курсового проекта**

Выполнение курсового проекта преследует следующие цели:

- закрепление и углубление теоретических знаний студентов по дисциплине “Микропроцессорные системы”;
- обучение студентов использованию приобретенных знаний для решения конкретных прикладных задач построения микропроцессорных систем;
- 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 студента, поэтому студент имеет право предложить собственную тему проекта по согласованию с преподавателем;
- согласование задания на курсовое проектирование. На этом этапе студент согласует 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, так как чем более детально проработано ТЗ, тем проще потом его реализовать;
- 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 (приложение А).
2. Задание (приложение Б).
3. Введение (краткое описание актуальности разработки, цель и задачи разработки, краткое описание содержания основных разделов пояснительной записки).
4. Описание структуры разрабатываемой системы.
5. Обоснование и разработка алгоритмов работы.
6. Выбор средств разработки.
7. Описание разработанных программных средств.
8. Тестирование системы
9. Заключение (краткие выводы по разделам 4-10 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 Стенды для отладки микропроцессорной системы на базе ВИП-МК**

Отладочные стенды предназначены для комплексной отладки аппаратного и программного обеспечения микропроцессорной системы на базе ВИП-МК. Структура и состав каждого отладочного стенда проектируется исходя из функций отлаживаемого программно-

аппаратного комплекса.

#### **4.1 Многофункциональный микропроцессорный терминал ВИП-МК**

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.

Подробную информацию об аппаратных и программных средствах ВИП-МК можно найти в [1,2,3,4,5].

#### **4.2 Построение стенда**

Комплект оборудования, входящего в состав стенда, зависит от конкретной системы.

В зависимости от конкретной системы в состав стенда входят:

- один или несколько ВИП-МК;
- персональный компьютер;
- кабели (витая пара) для включения ВИП-МК по интерфейсу Ethernet в локальную сеть;
- нуль-модемные кабели для соединения ВИП-МК по интерфейсу RS-232;
- дополнительное оборудование.

Ниже приведены примеры стендов для отладки различных классов программного обеспечения.

#### **4.3 Стенд для отладки не-коммуникационного ПО**

Не-коммуникационное ПО предполагает, что ввод и вывод обрабатываемой информации будут осуществляться в пределах одного вычислительного устройства, в данном случае – терминала ВИП-МК.

Структурная схема стенда для отладки такого ПО представлена на Рис.1.

Персональный компьютер (ПК) и соединение по локальной сети необходимо для загрузки отлаживаемого ПО на ВИП-МК и

осуществления контроля за его работой. USB-аудиогарнитура, микрофон и колонки (наушники) необходимы только, если отлаживаемое ПО работает со звуком. Так же возможно подключение дополнительного оборудования, если программа предполагает работу с ним. В качестве дополнительного оборудования могут использоваться, например, светосигнальные устройства, подключаемые ко разъёму «управление».

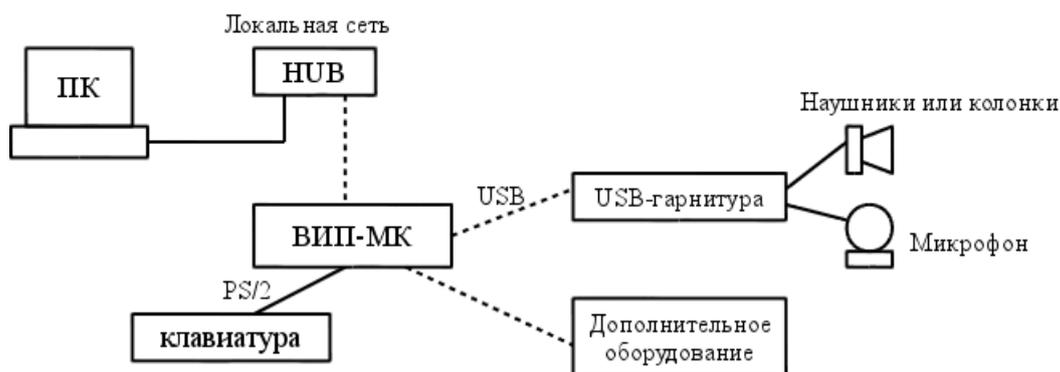


Рис. 1 Структурная схема стенда для отладки не-коммуникационного ПО

#### 4.4 Стенд для отладки ПО, использующего Ethernet для коммуникации между несколькими ВИП-МК

Структура стенда предполагает, что программное обеспечение включает функции взаимодействия между несколькими ВИП-МК и связь осуществляется по локальной сети с использованием интерфейса Ethernet.

Структурная схема стенда для отладки такого ПО представлена на Рис.2.

К каждому ВИП-МК, может подключаться дополнительное оборудование, USB-аудиогарнитура, клавиатура в зависимости от конкретного назначения отлаживаемого ПО.

Локальная сеть используем Персональный компьютер (ПК) и соединение по локальной сети необходимо для загрузки отлаживаемого ПО на ВИП-МК и осуществления контроля за его работой.

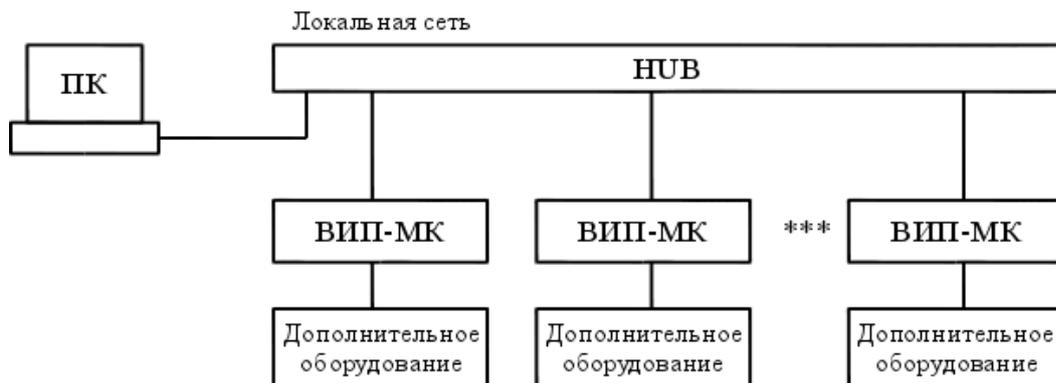


Рис. 2: Структурная схема стенда для отладки ПО, использующего Ethernet для коммуникации между несколькими ВИП-МК

## 5 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 [6].

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.

## 6 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 [7], краткие примеры программ с описанием – в [8,9].

Так же настоятельно рекомендуется ознакомиться с утилитой **make**, позволяющей автоматизировать сборку программ. Примеры использования утилиты **make** имеются в [7], а подробное описание (оригинал) – в [10].

Предположим, имеется программа, состоящая из одного файла с исходными текстами на языке С – **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 С++ 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 С – **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 можно получить в [7,10].

## 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. Структура и функции ПО ВИП-МК. Краткое описание. [Электронный ресурс] . – Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
3. Файловая система ВИП-МК [Электронный ресурс]. – Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
4. Последовательные порты ВИП-МК [Электронный ресурс]. – Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
5. Терминал ВИП-МК. Справочные данные [Электронный ресурс]. – Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
6. Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes. SSH, The Secure Shell: The Definitive Guide O'Reilly Media // ISBN 0596008953; 2005 г.
7. Гриффитс Артур. GCC. Настольная книга пользователей, программистов и системных администраторов / К.: ООО ТИД "ДС", 2004, 624с.
8. Знакомство с компилятором GCC [Электронный ресурс]. - Режим доступа: <http://www.linuxcenter.ru/lib/books/linuxdev/linuxdev1.phtml> свободный. – Загл. с экрана.
9. Компилятор G++ [Электронный ресурс]. – Режим доступа: <http://www.linuxcenter.ru/lib/books/linuxdev/linuxdev4.phtml> свободный. – Загл. с экрана.
10. 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  
"Микропроцессорные системы"

Студент гр. \_\_\_\_\_

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

Руководитель: ассистент .каф.

ИПС, канд.техн.наук

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

2012

## 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

По курсовому проектированию студенту Иванову Александру Борисовичу

группа 8В82

1 Тема проекта: Многофункциональный будильник с микропроцессорным управлением

2 Срок сдачи студентом законченного проекта \_\_\_\_\_

3 Исходные данные к проекту:

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. Выбор средств разработки.
4. Описание разработанных программных средств.
5. Тестирование системы.
6. Заключение (краткие выводы по разделам 2-5 и общий вывод по работе).
7. Список использованных источников.
8. Приложение (исходный текст программы с комментариями, возможно указание ссылки на него, возможно в электронном виде).

Задание принял к исполнению \_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_

\_\_\_\_\_

дата

\_\_\_\_\_

подпись студента

**Приложение В. Пример курсового проекта**

**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)

**ПРОГРАММИРУЕМЫЙ БУДИЛЬНИК СО ЗВУКОВОЙ  
ИНДИКАЦИЕЙ**

Пояснительная записка к курсовому проекту по дисциплине  
"Микропроцессорные системы"

Студент гр. \_\_\_\_\_

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

Руководитель: ассистент .каф.  
ИПС,

канд.техн.наук

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

2012

# 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

По курсовому проектированию студенту Иванову Александру  
Борисовичу  
группа 8В82

1 Тема проекта: Программируемый будильник со звуковой  
индикацией

2 Срок сдачи студентом законченного проекта 15.12.2012

3 Исходные данные к проекту:  
учебно–методическая литература;  
требования к содержанию и оформлению проекта.

4 Техническое задание на курсовой проект.

Требуется разработать программное обеспечение для многофункционального терминала ВИП-МК, осуществляющее функции программируемого будильника со звуковой индикацией.

#### Функции ПО.

- ▲ Вывод на экран текущего времени.
- ▲ Вывод на экран состояния будильника (отключен, однократное срабатывание, ежедневное срабатывание).
- ▲ Вывод на экран времени срабатывания будильника.
- ▲ Установка с клавиатуры текущего времени.
- ▲ Установка с клавиатуры состояния будильника (отключен, однократное срабатывание, ежедневное срабатывание).
- ▲ Установка с клавиатуры времени срабатывания будильника.
- ▲ При срабатывании будильника, звуковой сигнал отключается при нажатии на любую клавишу клавиатуры или по истечении двух минут со времени срабатывания.
- ▲ Точность установки текущего времени – 1 минута.
- ▲ Точность времени срабатывания будильника – 1 минута.

#### Требования к аппаратной части и системному ПО.

- ▲ Многофункциональный терминал ВИП-МК.
- ▲ Звуковая карта с интерфейсом USB.
- ▲ Клавиатура PS/2.
- ▲ Колонки.
- ▲ ОС Linux версии не ниже 2.6.16.
- ▲ Звуковая подсистема OSS.

5. Содержание пояснительной записки (перечень подлежащих разработке вопросов)

1. Введение
2. Обоснование и разработка алгоритмов работы

3. Выбор средств разработки
4. Описание разработанных программных средств.
5. Тестирование системы.
6. Заключение
7. Список использованных источников
8. Приложение (исходный текст программы с комментариями, возможно указание ссылки на него, возможно в электронном виде).

Задание принял к исполнению \_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_

\_\_\_\_\_

дата

\_\_\_\_\_

подпись студента

## **Введение**

В настоящее время для решения различных задач управления, сбора и передачи данных широко применяются микропроцессорные устройства, представляющие собой специализированные микрокомпьютеры. Одним из таких микрокомпьютеров является многофункциональный микропроцессорный терминал ВИП-МК. Платформа ВИП-МК представляет собой микропроцессорный терминал на базе процессора AT91RM9200, работающий под управлением ОС Linux.

В данном курсовом проекте на основе платформы ВИП-МК реализовано устройство «Программируемый будильник со звуковой индикацией». Для реализации данного устройства потребовалось решить следующие задачи:

- разработать и согласовать техническое задания на проект;
- обосновать и разработать алгоритмы работы устройства;
- осуществить выбор средств разработки;
- на основе созданных алгоритмов разработать программное обеспечение, используя выбранные средства разработки;
- разработать и опробовать алгоритм тестирования системы;
- сделать общие выводы по проделанной работе.

## **Обоснование и разработка алгоритмов работы**

Алгоритмы разрабатываемой программы основаны на трёх составляющих: функциях ПО, описанных в ТЗ; аппаратной составляющей, требования к которой выставлены в ТЗ и средствах, предоставляемых операционной системой для работы с устройствами. Для реализации функций, описанных в ТЗ, необходимо обеспечить взаимодействие со следующими устройствами:

- символьный дисплей 2x40 символов, входящий в состав ВИП-МК. Используется для индикации текущего времени, отображения состояния и времени срабатывания будильника;
- клавиатура PS/2, подключаемая к ВИП-МК. Используется для установки текущего времени, времени срабатывания будильника,

состояния будильника, отключения звукового сигнала при срабатывании будильника;

- звуковая карта с интерфейсом USB. Используется для вывода звука.

ОС Linux предоставляет следующие средства для работы с перечисленными устройствами:

- символьный дисплей 2x40 символов, входящий в состав ВПП-МК, представлен как устройство вывода «последовательный порт» /dev/ttyS0 со скоростью 19200бит/сек;
- клавиатура PS/2, подключаемая к ВПП-МК представлена как устройство ввода «последовательный порт» /dev/ttyS0 со скоростью 19200бит/сек;
- звуковая карта с интерфейсом USB представлена как устройство ввода-вывода /dev/dsp.

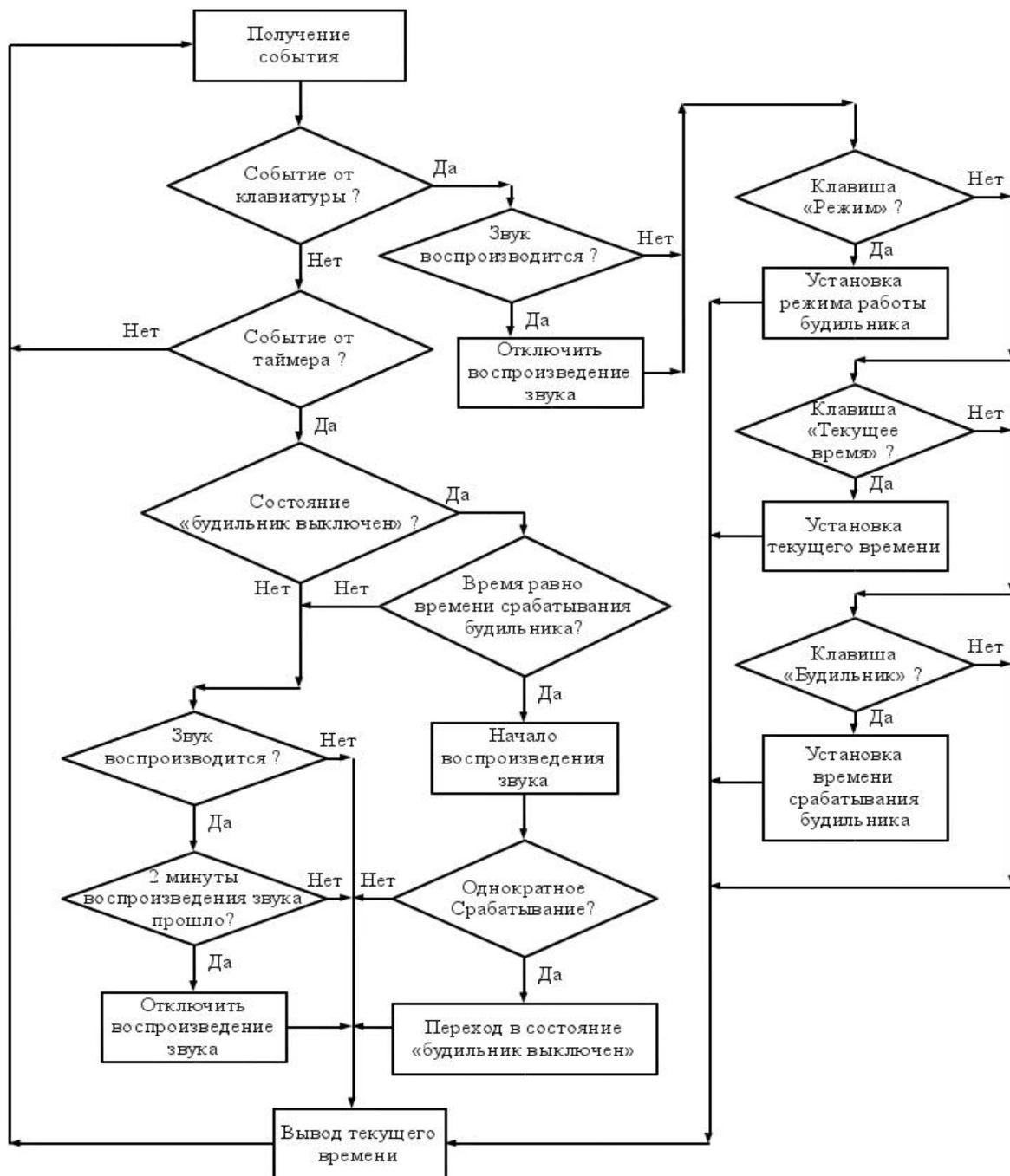
На основании ТЗ, определено, что имеется четыре режима работы программы – «основной» и три режима настройки: «установка режима работы будильника», «установка текущего времени» и «установка времени срабатывания будильника».

Для перехода из основного режима работы программы в один из режимов настройки используются клавиши, названные условно «Режим», «Текущее время», «Будильник».

После завершения настройки осуществляется переход в «основной» режим работы программы.

«Основной» режим работы программы представляет собой бесконечный цикл, начинающийся с получения события. Возможно событие одного из двух типов – от таймера (периодическое ежеминутное событие) и от клавиатуры (по нажатию одной из клавиш).

По событию от таймера – программа выполняет ветвь основного режима, по событию от клавиатуры – ветвь одного из режимов настройки. Затем происходит вывод текущего времени и программа возвращается к ожиданию события. При нажатии любой клавиши – прекращается воспроизведение звука будильника, если он воспроизводился. Общий алгоритм работы программы представлен на Рис.1.



1. Общий алгоритм программы «Программируемый будильник со звуковой индикацией»

## Выбор средств разработки

Для реализации разработанного алгоритма выбран язык C++, входящий в состав пакета компиляторов GCC и утилита make для автоматизации сборки программ.

Данные средства разработки являются стандартными для ОС Linux и позволяют получить оптимизированный или отладочный код, в зависимости от настроек.

## Описание разработанных программных средств.

В результате реализации разработанного алгоритма на языке C++, получены следующие файлы с исходным текстом:

main.cxx – головной файл;

base\_mode.cxx – реализация ветки алгоритма «основного» режима;

input\_mode.cxx – ввод режима работы будильника;

input\_time.cxx – ввод и установка текущего времени;

input\_alarm.cxx – ввод и установка времени срабатывания будильника;

sound.cxx – управление выводом звука;

tinout.cxx – терминальный ввод-вывод. Ввод с клавиатуры и вывод на дисплей объединены в один программный модуль, поскольку для взаимодействия с клавиатурой и дисплеем используется одно устройство – последовательный порт /dev/ttyS0.

Каждый файл с исходным текстом на C++ имеет соответствующий ему заголовочный файл с расширением \*.h.

## Тестирование системы

Тестирование системы включает в себя серию проверок (тестов), позволяющих определить, насколько функциональность разработанного программного обеспечения соответствует ТЗ. В случае, если какой-либо из тестов не прошёл, то система считается неработоспособной.

## **Заключение**

В результате проделанной работы спроектировано и реализовано устройство «Программируемый будильник со звуковой индикацией» на платформе многофункционально микропроцессорного терминала ВИП-МК.

## 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. Структура и функции ПО ВИП-МК. Краткое описание. [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
3. Файловая система ВИП-МК [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
4. Последовательные порты ВИП-МК [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
5. Терминал ВИП-МК. Справочные данные [Электронный ресурс] .- Режим доступа: <http://portal.tpu.ru/SHARED/s/SALEX> свободный. – Загл. с экрана.
6. Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes. SSH, The Secure Shell: The Definitive Guide O'Reilly Media // ISBN 0596008953; 2005 г.
7. Гриффитс Артур. GCC. Настольная книга пользователей, программистов и системных администраторов / К.: ООО ТИД "ДС", 2004, 624с.
8. Знакомство с компилятором GCC [Электронный ресурс] .- Режим доступа: <http://www.linuxcenter.ru/lib/books/linuxdev/linuxdev1.phtml> свободный. – Загл. с экрана.
9. Компилятор G++ [Электронный ресурс] .- Режим доступа: <http://www.linuxcenter.ru/lib/books/linuxdev/linuxdev4.phtml> свободный. – Загл. с экрана.
10. 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**

Код программы (гиперссылка, носитель).

**Приложение Б. Календарный рейтинг-план выполнения курсового проекта**

по дисциплине	Микропроцессорные системы
ООП подготовки	бакалавров
направления	
на период	весенний семестр 2011/2012 учебного года

Дата рубежного контроля*	Вид работы	Максимальный балл
	<i>Выбор темы курсового проекта</i>	<b>5</b>
	<i>Обоснование и разработка алгоритмов работы</i>	<b>10</b>
	<i>Выбор средств разработки, аппаратного обеспечения, операционной системы</i>	<b>5</b>
	<b>Контрольная точка 1</b>	<b>Всего по КТ 1</b>
	<i>Разработка и отладка программного обеспечения</i>	<b>10</b>
	<i>Тестирование разработанного программного обеспечения</i>	<b>10</b>
	<b>Контрольная точка 2</b>	<b>Всего по КТ 2</b>
<b>Итого по результатам текущего контроля в семестре</b>		<b>40</b>
<b>По результатам защиты курсового проекта</b>		<b>60</b>
<b>Итого баллов по дисциплине</b>		<b>100</b>

Составил: \_\_\_\_\_ (\_\_\_\_\_)

«\_\_\_» \_\_\_\_\_ 201\_\_ г.

Согласовано:

Заведующий кафедрой \_\_\_\_\_ (\_\_\_\_\_)

«\_\_\_» \_\_\_\_\_ 201\_\_ г.

Учебное издание

СОНЬКИН Михаил Аркадьевич  
ШАМИН Алексей Алексеевич

## МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

Методические указания к выполнению курсового проекта  
для студентов IV курса, обучающихся 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. 1,69. Уч.-изд.л. 1,81.  
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