



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ  
КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной тех-  
ники и автоматизированных систем»

## **Учебно-методическое пособие** по дисциплине

# **«СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»**

Автор  
Кузин А. П.

Ростов-на-Дону, 2019



## Аннотация

Учебно-методическое пособие предназначено для студентов очной формы обучения направления 02.03.03 «Математическое обеспечение и администрирование информационных систем».

## Автор

Старший преподаватель  
кафедры ПОВТиАС  
Кузин А.П.



## Оглавление

<b>1. Общие указания к выполнению лабораторных работ</b>	<b>5</b>
1.1.    Цель лабораторного практикума.....	<b>Ошибка! Залка не определена.</b>
1.2.    Требования к лабораторному оборудованию .....	<b>Ошибка! Залка не определена.</b>
1.3.    Требования к отчетам студентов.....	<b>Ошибка! Залка не определена.</b>
1.4.    Литература .....	6
<b>2. Лабораторная работа №1: Знакомство с ОС QNX</b>	<b>6</b>
2.1.    Цель работы.....	6
2.2.    Общие сведения.....	6
2.3.    Ход работы.....	7
2.4.    Литература .....	7
2.5.    Контрольные вопросы .....	8
<b>3. Лабораторная работа №2: Знакомство с микроконтроллерами</b>	<b>8</b>
3.1.    Цель работы .....	8
3.2.    Общие сведения .....	8
3.3.    Литература .....	11
3.4.    Контрольные вопросы .....	11
<b>4. Лабораторная работа №3: Программа с основным циклом</b>	<b>11</b>
4.1.    Цель работы... .....	11
4.2.    Задание .....	12
4.3.    Варианты заданий .....	12
4.4.    Литература .....	13
4.5.    Контрольные вопросы .....	13
<b>5. Лабораторная работа №4: Программа, управляемая прерываниями</b>	<b>13</b>
5.1.    Цель работы... .....	13
5.2.    Задание .....	13
5.3.    Варианты .....	14
5.4.    Литература .....	14
5.5.    Контрольные вопросы	<b>1Ошибка! Залка не</b>

<b>определена.</b>	
<b>6. Лабораторная работа №5: Работа с таймером .....</b>	<b>15</b>
6.1.    Цель работы... ..	1 <b>Ошибка! Залкадка не определена.</b>
6.2.    Задание .....	1 <b>Ошибка! Залкадка не определена.</b>
6.3.    Варианты	1 <b>Ошибка! Залкадка не определена.</b>
6.4.    Литература .....	1 <b>Ошибка! Залкадка не определена.</b>
6.5.    Контрольные вопросы .....	16
<b>7. Лабораторная работа №6: Корпоративная многозадачность. ....</b>	<b>16</b>
7.1.    Цель работы... ..	16
7.2.    Задание .....	16
7.3.    Варианты.....	17
7.4.    Литература.....	18
7.5.    Контрольные вопросы.....	18
<b>Список литературы .....</b>	<b>19</b>

## 1. ОБЩИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

### 1.1. Цель лабораторного практикума

Основные задачи практикума:

- выработать у студентов навыки работы с СРВ;
- изучить особенности микроконтроллеров;
- отработать методы программирования микроконтроллеров;
- обеспечить студентов навыками построения систем реального времени с использованием микроконтроллеров;
- закрепить на практике основные приемы написания программ для микроконтроллеров.

Лабораторный практикум является достаточно самостоятельной частью данного курса. При проведении лабораторных работ следует уделять внимание:

- тонкостям построения СРВ;
- особенностям работы микроконтроллеров;
- написанию программ для микроконтроллеров;
- построению систем реального времени на микроконтроллерах.

### 1.2. Требования к лабораторному оборудованию

Для выполнения лабораторной работы №1 необходимо наличие на компьютерах класса любой современной виртуальной машины (напр. VirtualBox или VM Ware).

Для выполнения лабораторных работ необходим компьютерный класс, работающий под управлением ОС Windows с установленными на все компьютеры системой программирования AVR Studio версии не ниже 5.0.

При наличии возможности желательна установка ПО Proteus, позволяющего моделировать работу схем с применением микроконтроллеров.

### 1.3. Требования к отчетам студентов

В качестве отчета выступает протокол хода работы, либо исходный код полученных программ. В последнем случае исходный код должен быть оформлен согласно стандартным

правилам форматирования исходных кодов, например так:

[http://www.opennet.ru/docs/RUS/coding\\_standard/standard-8.html](http://www.opennet.ru/docs/RUS/coding_standard/standard-8.html).

Для приведения исходных кодов к общему виду поощряется использование утилит `indend`, `astyle`, `uncrustify`.

Исходный код может снабжаться комментариями, однако студент должен быть готов к тому, что при сдаче лабораторной работы преподаватель изменит кодировку для невозможности чтения подсказок из комментариев.

При защите лабораторной работы исходные коды должны иметься у студента на электронном носителе. На передачу желательно иметь также твердую копию (распечатку) исходных кодов без комментариев.

Обязательным является разбор студентами контрольных вопросов к лабораторной работе.

#### 1.4. Литература

В конце каждой лабораторной работы приведена рекомендуемая литература. Ссылки даны на открытые электронные ресурсы «не запятнанные жёсткими запретами на воспроизведение и использование» и таким образом, чтобы студенты могли ознакомиться с этой литературой дома во время подготовки к лабораторной работе. Во время выполнения работы часть материалов будет доступна из компьютерных аудиторий университета.

## 2. ЛАБОРАТОРНАЯ РАБОТА №1: ЗНАКОМСТВО С ОС QNX

### 2.1. Цель работы

Установка QNX на виртуальную машину, знакомство с интерфейсом пользователя.

**Цель работы:** получение практических навыков установки ОС QNX и знакомство с её интерфейсом.

### 2.2. Общие сведения

QNX - "UNIX system-like" микроядерная операционная система реального времени. Обладает чрезвычайно низкой требовательностью к аппаратным ресурсам. Ядро этой ОС, называемое `Neutrino`, имеет размер всего 32 кб и выполняет только самые базовые функции, остальные же сервисы реализованы в динамически подключаемых модулях. В минимальной комплектации она умещается на обычной дискете. Главная область ее применения

- сложное и ответственное производство с серьезными требованиями к надежности и непрерывности управления, а также ко времени реакции в аварийных ситуациях.

Более подробную информацию можно получить в документации на оф. сайте <http://www.qnx.com/> и в Википедии. Относительно небольшое количество материала по этой ОС в интернете еще более подстегнуло любопытство посмотреть не нее "вживую".

### 2.3. Ход работы

1. Получить у преподавателя iso-образ, с которого будет производиться установка системы и ключ активации.

2. Создать виртуальную машину, все параметры можно оставить по-умолчанию, разме жесткого диска установить в 1GB.

3. Подключить iso-образ.

4. Запустить виртуальную машину и начать установку.

- В процессе установки будет произведен запрос ключа активации. Необходимо ввести пару e-mail/ключ, полученные у преподавателя.
- При выборе раздела для установки можно соглашаться с предлагаемыми инсталлятором настройками (F1 Use all available space) , остальные настройки тоже можно оставить по умолчанию, однако желательно разобраться, что же оно хочет.
- Запрос на установку доп. программного обеспечения можно проигнорировать, кроме графической оболочки.

5. После завершения установки отключить iso-образ и перезапустить виртуальную машину..

6. Ознакомиться с интерфейсом и набором программ ОС QNX.

### 2.4. Литература

1. Установка QNX в картинках.

<http://www.cyberforum.ru/unix/thread173415.html>

2. Операционная система реального времени QNX:  
Знакомство

<http://habrahabr.ru/post/124656/>

### 2.5. Контрольные вопросы

1. Какие опции разбиения жесткого диска поддерживаются?
2. Какие пакеты доступны для установки при выборе дополнительного ПО?
3. Какие отличия имеются у интерфейса пользователя QNX?

## 3. ЛАБОРАТОРНАЯ РАБОТА №2: ЗНАКОМСТВО С МИКРОКОНТРОЛЛЕРАМИ

### 3.1. Цель работы

Изучение примеров отладочных плат различных производителей и микроконтроллеров в корпусах различного исполнения.

**Цель работы:** обеспечить студентов представлением о внешнем виде и особенностях современных микроконтроллеров.

### 3.2. Общие сведения

Микроконтроллер (англ. Micro Controller Unit, MCU) — микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ и (или) ПЗУ. Фактически микроконтроллер - это, маленький компьютер. Который имеет свой центральный процессор (регистры, блок управление и арифметическо-логическое устройство), память, а также разную периферию, вроде портов ввода вывода, таймеров, контроллеров прерываний, генераторов разных импульсов и даже аналоговых преобразователей.

Основной функцией микроконтроллера является «переключение между каналами (ножками)». Т.е. у него есть несколько выводов (от 6 до нескольких десятков в зависимости от модели) и на этих выводах он может выставить либо 1 (высокий уровень напряжения, например +5вольт), либо 0 (низкий уровень напряжения, около 0.1 вольта) в зависимости от программного алгоритма зашитого в его память. Также микроконтроллер может определять состояние сигнала на своих ножках (для этого они должны быть настроены на вход) — высокое там напряжение или низкое (ноль или единица). Современные микроконтроллеры также имеют на борту Аналогово Цифровой Преобразователь — это своеобразный вольтметр, позволяет не просто отследить 0 или 1



на входе, а полноценно замерить напряжение от 0 до опорного (обычно опорное равно напряжению питания) и представить его в виде числа от 0 до 1024 (или 255, в зависимости от разрядности АЦП).

Среди электронных компонентов микро контроллеров это один из самых универсальных устройств. Микроконтроллеров существует необычайно большое количество. Практически каждая уважающая себя фирма по производству радиокомпонентов выпускает свой собственный контроллер. Однако и в этом многообразии есть порядок.

На многочисленных форумах и конференциях по сей день идут ожесточенные бои на тему какое семейство лучше. Все дело в применении, масштабах задач и массе других параметров. У каждого семейства есть свои достоинства и недостатки. Мой выбор остановился на двух семействах микроконтроллеров AVR и ARM. По моему мнению эти два семейства на сегодняшний момент являются лидерами в мировой индустрии микропроцессоров. А в частности я бы хотел акцентировать свое внимание на STM32 от ARM и AT32 от AVR.

#### ARM

Компания занимается исключительно разработкой ядер и инструментов для них (компиляторы, средства отладки и т. п.), зарабатывая на лицензировании архитектуры сторонним производителям.

В 2007 году около 98 % из более чем миллиарда мобильных телефонов, продаваемых ежегодно, были оснащены по крайней мере одним процессором ARM. По состоянию на 2009 на процессоры ARM приходилось до 90 % всех встроенных 32-разрядных процессоров. Процессоры ARM широко используются в потребительской электронике — в том числе КПК, мобильных телефонах, цифровых носителях и плеерах, портативных игровых консолях, калькуляторах и компьютерных периферийных устройствах, таких как жесткие диски или маршрутизаторы.

Эти процессоры имеют низкое энергопотребление, поэтому находят широкое применение во встраиваемых системах и доминируют на рынке мобильных устройств, для которых немаловажен данный фактор.

Данный производитель микроконтроллеров заинтересовал своим ядром ARM Cortex-M3, а именно микроконтроллером STM32 о котором и пойдет речь ниже.

#### STM32

STM32 - это микроконтроллер, построенный на ядре ARM

Cortex-M3 [9]. Данное ядро имеет много преимуществ, которые будут перечислены ниже, но его основное преимущество на сегодняшний день - универсальность. За два года Cortex-M3 стал индустриальным стандартом. Об этом говорит количество производителей, присоединившихся к данной архитектуре. Все основные производители микроконтроллеров, кроме Microchip, имеют или развивают решения на основе этой архитектуры: STMicroelectronics, Texas Instrument, NXP, ATMEL, Analog Devices, Renesas и т.д. Компания ST одна из первых выпустила свои микроконтроллеры Cortex-M3 (2007 г.) и быстро стала доминирующим игроком на этом рынке. На рисунке 2.1 показано общее количество проданных в мире ядер Cortex-M3 и львиная доля ST: около 80% в 2009 году, и около 70% - в 2010. Это говорит о высоком качестве и привлекательности решений STMicroelectronics. Заметен и значительный рост продаж самих ядер Cortex-M3. В 2008 году их было продано порядка четырех миллионов, а затем три года подряд ежегодный рост составлял 400...500%.

#### ATMEL

Микроконтроллеры Atmel® объединяют в себе высокоэффективные встроенные решения, проверенные технологии и революционные идеи, являясь идеальным выбором для современных интеллектуальных сетевых продуктов. В эпоху Интернета вещей микроконтроллерные технологии – ключевой аспект развития межмашинных (M2M) коммуникаций.

Компания Atmel уже десятки лет занимает ведущие позиции в своей отрасли, и опыт, полученный за это время, позволил создать оптимизированные архитектуры, обеспечивающие низкое потребление энергии, высокоскоростное взаимодействие, отличную пропускную способность и поддержку множества интерфейсов. Широкий выбор вариантов конфигураций дает разработчикам возможность проектировать готовые системные решения для любых приложений.

Микроконтроллеры Atmel также поддерживают простую и удобную интеграцию технологии емкостного сенсорного ввода для реализации кнопок, ползунков и колес прокрутки. Кроме того, микроконтроллеры Atmel позволяют внедрять беспроводные функции и функции защиты.

32-разрядные микроконтроллеры Atmel семейства AVR® уникальным образом сочетают в себе производительность, эффективность энергопотребления и гибкость проектирования. Эти устройства оптимизированы для сокращения цикла разработки, обладают самой эффективной в промышленности архитектурой

для программирования на языке С и ассемблере и способны быстро подстраиваться под любые новые требования рынка. Широкий спектр устройств AVR в сочетании с легко интегрируемой платформой Atmel Studio значительно упрощает повторное использование наработанных материалов при развитии продуктов и выходе на новые рынки.

#### ATMEL AVR

Линейка представлена тремя семействами.

Classic AVR - это классика AVR, что следует и из их названия. Постепенно заменяются микроконтроллерами семейств Tiny и Mega.

Tiny AVR - недорогие микроконтроллеры, чаще в восьмивыводном исполнении. Используются в системах, где не требуются возможности Mega AVR.

Mega AVR - мощные микроконтроллеры, включающие хороший набор периферии. Это семейство имеет самое большое разнообразие моделей для выбора.

### 3.3. Литература

1. Микроконтроллер  
<http://ru.wikipedia.org/wiki/Микроконтроллер>
2. STM32 – универсальное решение на ARM-ядре  
<http://www.kosmodrom.com.ua/data/stm32.php>
3. Модели микроконтроллеров AVR  
[http://www.myrobot.ru/stepbystep/mc\\_table.php](http://www.myrobot.ru/stepbystep/mc_table.php)

### 3.4. Контрольные вопросы

1. Что находится «на борту» современного микроконтроллера?
2. Какие интерфейсы бывают у микроконтроллеров?
3. Чем отличаются микроконтроллеры от ЭВМ?
4. Какого размера бывают современные микроконтроллеры?

## 4. ЛАБОРАТОРНАЯ РАБОТА №3: ПРОГРАММА С ОСНОВНЫМ ЦИКЛОМ

### 4.1. Цель работы

Инициализация микроконтроллера. Опрос кнопок микроконтроллера, управление светодиодной индикацией (по вариантам).

**Цель работы:** знакомство с программированием микроконтроллеров.

#### 4.2. Задание

Написать минимальную программу с конфигурацией согласно варианту. Основная программа должна зажигать или переключать светодиод по нажатию кнопок. Отслеживание нажатий кнопок можно производить в основном цикле программы.

При наличии нескольких входов считать, что на входах установлены отдельные кнопки. При наличии нескольких выходов считать, что к ним подключены отдельные светодиоды и их включение/выключение должно проводиться одновременно.

Считать, что кнопки подключаются от входа на землю, без «подтягивающего» резистора.

#### 4.3. Варианты заданий

Таблица 4.1 – Варианты заданий

No	Частота, МГц	Входы	Выходы	Что делать
1	0,6	1,2,3	4	включать и выключать по разным кнопкам
2	1,2	1,2	3	переключать по одной из кнопок
3	2,4	3,4	1,2	включать и выключать по разным кнопкам
4	4,8	1,2	3,4,5	переключать по одной из кнопок
5	9,6	1,4	2,3	включать и выключать по разным кнопкам
6	0,6	2,4	1,3	переключать по одной из кнопок
7	1,2	1,3	2,4	включать и выключать по разным кнопкам
8	2,4	1	2	переключать по одной из кнопок
9	4,8	3,4	1	включать и выключать по разным кнопкам
10	9,6	1,3	2	переключать по одной из кнопок
11	0,6	1,2	3	включать и выключать по разным кнопкам
12	1,2	3,4	1,2	переключать по одной из кнопок
13	2,4	1,2	3,4,5	включать и выключать по разным кнопкам
14	4,8	1,4	2,3	переключать по одной из кнопок
15	9,6	2,4	1,3	включать и выключать по разным кнопкам
16	0,6	1,3	2,4	переключать по одной из кнопок
17	1,2	1	2	включать и выключать по разным кнопкам
18	2,4	3,4	1	переключать по одной из кнопок
19	4,8	1,3	2	включать и выключать по разным кнопкам
20	9,6	1,2,3	4	переключать по одной из кнопок

#### 4.4. Литература

1. AVR. Учебный курс. Архитектура программ <http://easyelectronics.ru/avr-uchebnyj-kurs-arxitektura-programm.html>
2. AVR. Учебный курс. Программирование на Си. Часть 2 <http://easyelectronics.ru/avr-uchebnyj-kurs-programmirovanie-na-si-chast-2.html>
3. AVR. Учебный курс. Работа с портами ввода-вывода. Практика. <http://easyelectronics.ru/avr-uchebnyj-kurs-rabota-s-portami-vvoda-vyvoda-praktika.html>

#### 4.5. Контрольные вопросы

1. В каких режимах может работать порт микроконтроллера?
2. Как производится задание частоты микроконтроллера?
3. Как опросить вывод и установить новое состояние?
4. Какие функции могут быть у выводов микроконтроллера?

### 5. ЛАБОРАТОРНАЯ РАБОТА №4: ПРОГРАММА, УПРАВЛЯЕМАЯ ПРЕРЫВАНИЯМИ

#### 5.1. Цель работы

Опрос кнопок микроконтроллера, управление светодиодной индикацией (по вариантам).

**Цель работы:** освоить обработку прерываний микроконтроллерами; углубить знания об архитектуре микроконтроллеров.

#### 5.2. Задание

Описать собственный обработчик прерывания таймера, производящий периодические действия, заключающиеся во включении и выключении светодиода на одном из выводов. Интервал срабатывания таймера подобрать самостоятельно. Обязательно привести расчёты в комментариях в программе. Приветствуется использование прерывания по изменению состояния порта для запуска и останова действий по таймеру.

### 5.3. Варианты

Таблица 5.1 – Варианты заданий

No	Частота МК	Период таймера, мкс	Период действий	Длительность действия
1	0,6	2460	31980	17220
2	1,2	1500	28500	3000
3	2,4	2440	29280	12200
4	4,8	2740	63020	32880
5	9,6	1090	26160	4360
6	0,6	1100	8800	3300
7	1,2	3700	96200	44400
8	2,4	3690	99630	11070
9	4,8	3000	78000	24000
10	9,6	2250	49500	13500
11	0,6	2510	17570	5020
12	1,2	1240	37200	4960
13	2,4	1230	20910	3690
14	4,8	2470	7410	2470
15	9,6	980	10780	4900
16	0,6	860	9460	5160
17	1,2	3620	90500	43440
18	2,4	2180	34880	19620
19	4,8	3080	80080	6160
20	9,6	950	4750	2850

### 5.4. Литература

1. AVR. Учебный курс. Архитектура программ  
<http://easyelectronics.ru/avr-uchebnyj-kurs-arxitektura-programm.html>
2. AVR. Учебный курс. Программирование на Си. Часть 3  
<http://easyelectronics.ru/avr-uchebnyj-kurs-programmirovanie-na-si-chast-3.html>
3. AVR. Учебный курс. Работа с портами ввода-вывода. Практика.  
<http://easyelectronics.ru/avr-uchebnyj-kurs-rabota-s-portami-vvoda-vyvoda-praktika.html>
4. AVR. Учебный курс. Таймеры.  
<http://easyelectronics.ru/avr-uchebnyj-kurs-tajmery.html>

### 5.5. Контрольные вопросы

1. Как работает прерывание?
2. Какие прерывания есть в используемом микроконтроллере?
3. Как устанавливается обработчик прерывания?
4. Как работает таймер? С какими регистрами?
5. Какие прерывания таймера доступны программисту?
6. Как рассчитать или подобрать частоту срабатывания таймера?

## 6. ЛАБОРАТОРНАЯ РАБОТА №5: РАБОТА С ТАЙМЕРОМ

### 6.1. Цель работы

**Цель работы:** углубить знания о внутреннем устройстве и деталях работы таймеров микроконтроллера; освоить квазипараллельное выполнение действий в однопроцессорных системах.

### 6.2. Задание

Реализовать с помощью диспетчера параллельные процессы. Диспетчер задач - без приоритетов. Включение и выключение задач — по результатам мониторинга состояния кнопок.

### 6.3. Варианты

1. Моргалка светодиодами (20 раз/сек переключаются с одного светодиода на другой и обратно) с возможностью изменения яркости светодиодов кнопками (2 кнопки — увеличение и уменьшение яркости). Яркость формировать через ШИМ.
2. Светофор на перекрестке (крайне желательно ознакомиться с тем, как работает реальный светофор). Автоматическое переключение с периодом 3 минуты.
3. Светофор на пешеходном переходе (крайне желательно ознакомиться с тем, как работает реальный светофор). Переключается как автоматически (период 2 минуты), так и по нажатию кнопки (через 15 секунд после нажатия)
4. Семафор на переезде. При срабатывании кнопки 1 начинает перемигиваться 2 светодиодами, мигание прекращается через 10 секунд после срабатывания кнопки 2.

5. Модель танка. 4 кнопки и 4 соответствующих выхода управления движением (вперед и назад для каждого из 2 моторов привода гусениц) плюс 2 кнопки — выстрел и аварийный останов. Аварийный останов выключает моторы на срок не менее 2 секунд. Выстрел требует не более чем 1 секунду подачи управляющего сигнала.

#### 6.4. Литература

1. AVR. Учебный Курс. Архитектура Программ Часть 2  
<http://easyelectronics.ru/avr-uchebnyj-kurs-arxitektura-programm-chast-2.html>
2. AVR. Учебный курс. Программирование на Си. Часть 3  
<http://easyelectronics.ru/avr-uchebnyj-kurs-programmirovanie-na-si-chast-3.html>
3. AVR. Учебный курс. Таймеры.  
<http://easyelectronics.ru/avr-uchebnyj-kurs-tajmery.html>
4. AVR. Учебный курс. Конечный автомат  
<http://easyelectronics.ru/avr-uchebnyj-kurs-konechnyj-avtomat.html>

#### 6.5. Контрольные вопросы

1. Какие способы квазипараллельного выполнения задач вы знаете?
2. Как можно задействовать таймер для выполнения более чем одно действия (генерация сигналов и счёт времени)?
3. Как оценить точность счёта времени таймером для заданной частоты МК и режима работы таймера?
4. Какие режимы таймера вы знаете?

## 7. ЛАБОРАТОРНАЯ РАБОТА №6: КОРПОРАТИВНАЯ МНОГОЗАДАЧНОСТЬ

### 7.1. Цель работы

**Цель работы:** углубить знания о системах реального времени и квазипараллельном выполнении заданий.

### 7.2. Задание

Написать программу с корпоративной многозадачностью, то есть процессы должны выполняться квазипараллельно без использования очередей ожидания и других механизмов.



Каждый процесс выполняет часть своей работы и передает управление диспетчеру процессов. Диспетчер выбирает следующий процесс и передает ему управление.

Переключение процессов необходимо организовать подменной точек возврата из подпрограммы диспетчера в стеке.

### 7.3. Варианты

1. Тестовый прибор для проверки реакции. С интервалом от 1 до 5 секунд на 250 миллисекунд, по одному в случайном порядке включаются в режиме моргания 4 имеющихся светодиода. Если пользователь успевает нажать соответствующую горящему светодиоду кнопку, то данный светодиод перестает моргать и загорается в полную силу. Задача испытуемого — успеть зажечь все светодиоды за два периода включения светодиодов. Задачи переключения, опроса кнопок и слежения выиграл/не выиграл выполняются параллельно.

2. Таймер с двоичной индикацией - . Счет идет от 7 до 0 с интервалом в секунду. Отдельная задача — моргалка светодиодом — индикация работы (как на часах моргающие точки). На устройстве кнопка — одно нажатие - запуск/останов таймера, два нажатия с интервалом менее 1 секунды - сброс таймера в состояние 7. Таймер, моргалка и опрос кнопок выполняются параллельно.

3. Набор таймеров. Есть кнопка, запускающая один из свободных минутных таймеров (не более 5 таймеров одновременно). Есть индикаторы работы (5 шт), мигающие с частотой 2 Гц, если соответствующий таймер включен. Опрос кнопки и процедуры работы таймеров выполняются параллельно.

4. Есть 3 кнопки и 3 соответствующих им светодиода. В «случайный» момент времени запускается светодиод (интервал не менее 750мс). Задача пользователя — нажать на соответствующую ему кнопку в течение 500мс. Если загорятся все три светодиода — пользователь проиграл. Опрос кнопок, системный таймер (получает миллисекунды из тиков таймера) и ожидание проигрыша пользователя — параллельные процессы.

5. Замечание: случайные величины можно получить забыв время нажатия кнопки пользователем. Для этого можно пользоваться, к примеру, двумя величинами - значением программного таймера и текущим значением счетного регистра таймера (произведя над ними какое-либо математическое надруга-

тельство), либо использовать эти величины как начальное значение псевдослучайной последовательности (см. в «Псевдослучайные последовательности» в любой поисковой системе).

#### 7.4. Литература

1. AVR. Учебный Курс. Архитектура Программ Часть 2  
<http://easyelectronics.ru/avr-uchebnyj-kurs-arxitektura-programm-chast-2.html>
2. AVR. Учебный Курс. Архитектура Программ Часть 3  
<http://easyelectronics.ru/avr-uchebnyj-kurs-arxitektura-programm-chast-3.html>
3. AVR. Учебный Курс. Архитектура Программ Часть 4  
<http://easyelectronics.ru/avr-uchebnyj-kurs-vytesnyayushhij-dispetcher-zadach.html>
4. AVR. Учебный Курс. Работа на прерываниях  
<http://easyelectronics.ru/avr-uchebnyj-kurs-rabota-na-preryvaniyah.html>
5. AVR. Учебный курс. Таймеры.  
<http://easyelectronics.ru/avr-uchebnyj-kurs-tajmery.html>
6. AVR. Учебный курс. Конечный автомат  
<http://easyelectronics.ru/avr-uchebnyj-kurs-konechnyj-avtomat.html>

#### 7.5. Контрольные вопросы

1. Чем отличается корпоративная и вытесняющая многозадачность?
2. Как оценить загрузку микроконтроллера?
3. Как рассчитать необходимое количество тактов для выполнения подпрограммы?
4. Сколько тактов занимает команда микроконтроллера?
5. Как вставить ассемблерный код в функцию на Си?

## СПИСОК ЛИТЕРАТУРЫ

1. Установка QNX в картинках.  
<http://www.cyberforum.ru/unix/thread173415.html>
2. Операционная система реального времени QNX: Знакомство, <http://habrahabr.ru/post/124656/>
3. Микроконтроллер,  
<http://ru.wikipedia.org/wiki/Микроконтроллер>
4. STM32 – универсальное решение на ARM-ядре,  
<http://www.kosmodrom.com.ua/data/stm32.php>
5. Модели микроконтроллеров AVR,  
[http://www.myrobot.ru/stepbystep/mc\\_table.php](http://www.myrobot.ru/stepbystep/mc_table.php)
6. AVR. Учебный курс. Электроника для всех,  
<http://easyelectronics.ru/category/arm-uchebnyj-kurs>