

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**по выполнению лабораторных работ**  
**дисциплины «Программирование встраиваемых микропроцессорных**  
**систем»**  
для студентов направления подготовки  
09.03.01 Информатика и вычислительная техника,  
Направленность (профиль) «Программное обеспечение средств  
вычислительной техники и автоматизированных систем»

Ставрополь, 2026

## ОГЛАВЛЕНИЕ

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| ВВЕДЕНИЕ.....                                                                         | 4  |
| 1. ЛАБОРАТОРНАЯ РАБОТА №1. СТРУКТУРА<br>МИКРОКОНТРОЛЛЕРОВ.....                        | 5  |
| 2. ЛАБОРАТОРНАЯ РАБОТА №2. СТРУКТУРА ПРОЦЕССОРНОГО ЯДРА<br>МИКРОКОНТРОЛЛЕРА.....      | 10 |
| 3. ЛАБОРАТОРНАЯ РАБОТА №3. ПАМЯТЬ ПРОГРАММ И ДАННЫХ<br>МИКРОКОНТРОЛЛЕРА.....          | 15 |
| 4. ЛАБОРАТОРНАЯ РАБОТА №4. ПОРТЫ<br>ВВОДА/ВЫВОДА.....                                 | 19 |
| 5. ЛАБОРАТОРНАЯ РАБОТА №5. АППАРАТНЫЕ СРЕДСТВА<br>ОБЕСПЕЧЕНИЯ НАДЕЖНОЙ РАБОТЫ МК..... | 24 |
| СПИСОК ЛИТЕРАТУРЫ.....                                                                | 29 |

## **Введение**

Микропроцессорная система может рассматриваться как частный случай электронной системы, предназначенной для обработки входных сигналов и выдачи выходных сигналов. В качестве входных и выходных сигналов при этом могут использоваться аналоговые сигналы, одиночные цифровые сигналы, цифровые коды, последовательности цифровых кодов. Внутри системы может производиться хранение, накопление сигналов (или информации), но суть от этого не меняется. Если система цифровая (а микропроцессорные системы относятся к разряду цифровых), то входные аналоговые сигналы преобразуются в последовательности кодов выборок с помощью АЦП, а выходные аналоговые сигналы формируются из последовательности кодов выборок с помощью ЦАП. Обработка и хранение информации производятся в цифровом виде.

В данном курсе работ предлагается ознакомиться с внутренним устройством микропроцессорных систем. Основные затрагиваемые области: структура микроконтроллеров, структура процессорного ядра микроконтроллера, память программ и данных микроконтроллера, порты ввода/вывода, аппаратные средства обеспечения надежной работы МК.

## Лабораторная работа №1

### Структура микроконтроллеров

Цель работы – ознакомление с работой основных аппаратных средств современных микроконтроллеров.

#### Общие сведения

Все МК обладают процессорным ядром. Семейство объединяет ряд моделей. Отдельные модели в составе семейства различаются набором периферийных модулей, которые подключаются к внутренней межмодульной магистрали. Основные отличия между отдельными представителями семейства состоят в типе и объеме размещенной на кристалле резидентной памяти, количестве параллельных портов и контроллеров последовательных интерфейсов. Структура современных микроконтроллеров на примере МК MC68HC912B32 приведена на рис. 1

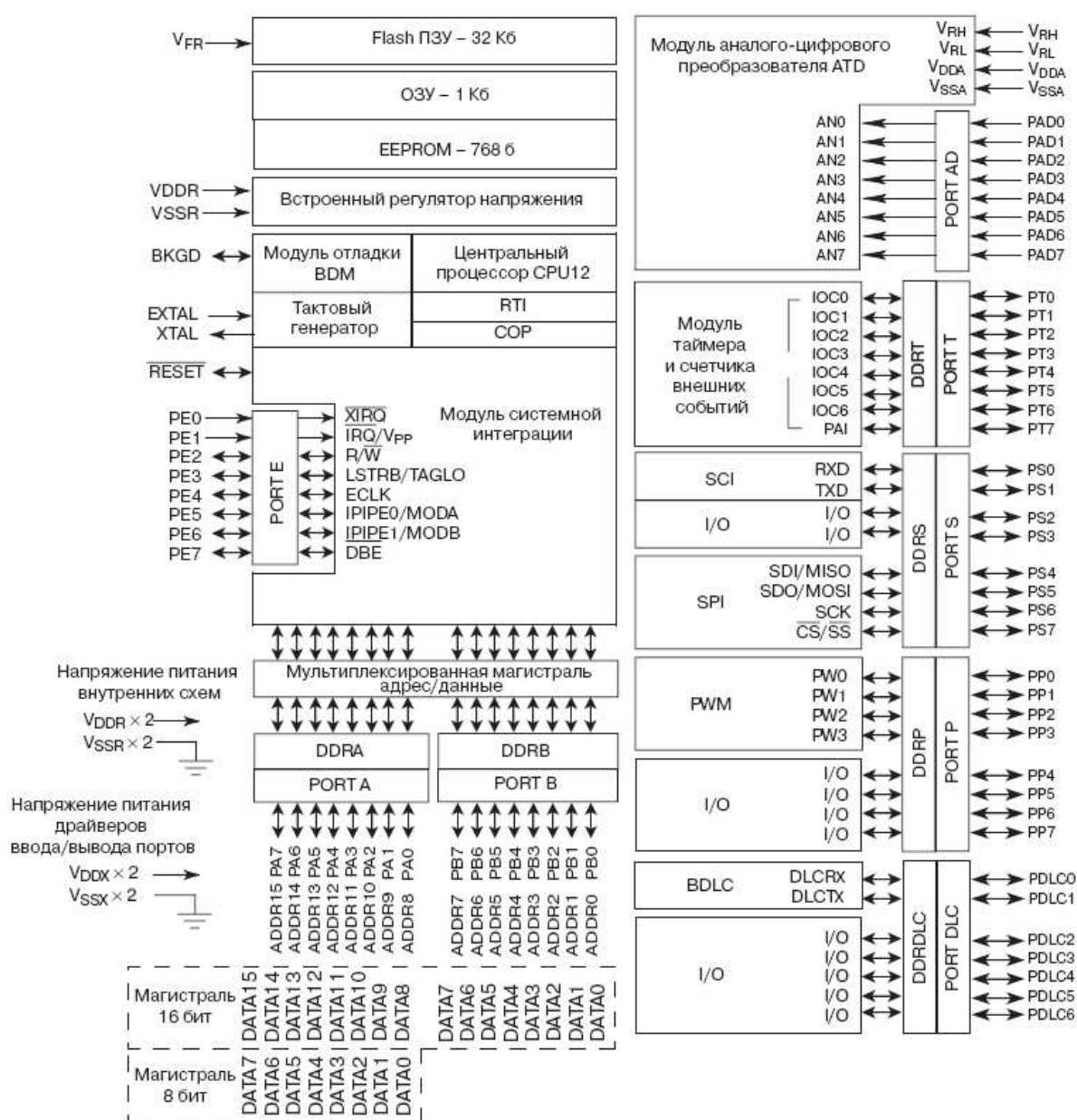


Рис.1. Структура современных микроконтроллеров

### Особенности современных микроконтроллеров на примере МК Motorola

**Резидентное ОЗУ объемом 1024 байта (1Кб).** Встроенное в кристалл микроконтроллера ОЗУ используется для хранения промежуточных результатов вычислений. Число ячеек, равное 1К, достаточно для большинства прогнозируемых для семейства 68HC12 применений.

**Резидентная энергонезависимая память данных типа EEPROM объемом 768 б.** Этот тип энергонезависимой памяти (энергонезависимая память сохраняет содержимое после отключения питания) обычно используют для сохранения изменяемых констант прикладной программы. Например, в области EEPROM могут храниться коды доступа к данной модели устройства, или на основе ячеек EEPROM могут быть организованы счетчики аварий исполнительного механизма, которым управляет микропроцессорный контроллер. Энергонезависимая память типа EEPROM позволяет выполнять операции записи и перезаписи содержимого ячеек памяти в течение сеанса работы микропроцессорного устройства под управлением прикладной программы, а также чтение ячеек памяти в произвольном порядке.

**Резидентная память программ типа Flash объемом 32 Кб.** Этот тип памяти предназначен для хранения прикладной программы, которая функционально завершена, прошла отладку и тестирование на реальном объекте. Объем памяти программ МК В32 составляет 32 Кб, что позволяет разместить в ней достаточно большие программы. В процессе работы над примерами из нашей книги Вы почувствуете, какой алгоритм может быть реализован программным кодом объемом 32 Кб. Использование Flash памяти в качестве памяти программ позволяет реализовать технологию программирования в системе ISP (In System Programming). Эта технология обеспечивает выполнение операций стирания и записи новых кодов в резидентное ПЗУ программ микроконтроллера без демонтажа МК с платы конечного изделия.

**Мультиплексированная шина адрес/данные для адресации внешней памяти и периферийных устройств.** Число выводов корпуса, в котором размещается полупроводниковый кристалл микроконтроллера, ограничено. Причиной тому – стремление разработчика выпускать миниатюрные и относительно недорогие МК, в то время, как увеличение числа выводов корпуса увеличивает его размеры и стоимость. Одним из способов сокращения числа выводов корпуса МК при сохранении функций этих выводов является мультиплексирование линий магистралей адреса и данных для сопряжения МК с внешней памятью. При мультиплексировании одни и те же выводы МК на протяжении одного временного интервала используются для передачи информации об адресе внешней ячейки памяти, а в течение другого временного интервала – для обмена данными с этой ячейкой. В микроконтроллере В32 функции мультиплексированных во времени магистралей адрес/данные выполняют линии портов Port A и Port B (см. рис.).

### Аппаратные средства современных микроконтроллеров на примере МК Motorola

**Многофункциональный таймер.** Подсистема реального времени МК семейства 68HC12 включает несколько модулей, но основным является таймер с 16-разрядным счетчиком временной базы, программируемым делителем частоты тактирования и 8 каналами входного захвата IC (Input Capture) или выходного сравнения ОС (Output Compare). Эти каналы могут быть сконфигурированы произвольно: любое число каналов из 8 настраивается на реализацию функции входного захвата IC, оставшиеся каналы – на функцию выходного сравнения ОС. При этом возможны конфигурации, когда все каналы находятся в режиме IC или в режиме ОС. Такая организация модуля таймера позволяет производить точные измерения временных характеристик входных сигналов МК, и генерировать многоканальные импульсные последовательности на его выходах. •

**Независимый 16 разрядный счетчик внешних событий.** Этот модуль также

принадлежит к подсистеме реального времени. Он предназначен для подсчета так называемых внешних событий, каждое из которых представляется импульсом на одном из входов МК.

Модули контроллеров последовательных интерфейсов SPI (Serial Peripheral Interface) и SCI (Serial Communication Interface). Микроконтроллеры семейства 68HC12 обладают достаточно мощными аппаратными средствами для обмена с другими устройствами в последовательном коде. Модуль контроллера SPI реализует обмен в синхронном режиме, в то время как модуль SCI предназначен для асинхронного последовательного обмена. Синхронный режим обмена в стандарте SPI характеризуется более высокими скоростями обмена по сравнению с стандартным асинхронным протоколом. Однако расстояние между взаимодействующими устройствами ограничено 20...30 см. Интерфейс в стандарте SPI часто используется для подключения к МК дополнительных интерфейсных компонентов, установленных на плате с МК. Например, МК семейства 68HC12 не имеют в своем составе цифроаналоговых преобразователей (ЦАП). Поэтому система с МК может быть дополнена внешней ИС ЦАП, подключенной к микроконтроллеру с использованием встроенного модуля контроллера SPI. Интерфейс асинхронного обмена SCI часто используется для обмена данным между двумя и более контроллерами, т.к. на его основе созданы интерфейсы, сигналы которых могут передаваться на значительные расстояния.

Модуль аналого-цифрового преобразователя ATD (Analog To Digital conversion system). Большинство сигналов в системах, которыми управляют микроконтроллеры, имеют аналоговую природу. Например, температура и давление воздуха окружающей среды изменяются плавно, а не скачкообразно. Для работы с аналоговыми сигналами в микропроцессорной системе, эти сигналы должны быть предварительно преобразованы в цифровую форму. В русскоязычной литературе говорят, что эти сигналы должны быть оцифрованы. Для этой цели в составе МК В32 имеется модуль аналого-цифрового преобразователя. Модуль имеет 8 входов для одновременного подключения восьми измеряемых аналоговых сигналов. Однако оцифровка этих сигналов будет производиться последовательно. Измеряемые аналоговые сигналы будут подключаться ко входу одного аналого-цифрового преобразователя (АЦП) посредством встроенного в модуль ATD мультиплексора. Оцифрованный сигнал представляется в 8 и или 10 разрядном прямом коде без знака. Два дополнительных бита кода представления результата позволяют увеличить чувствительность АЦП с 19.53 до 4.88 мВ.

Модуль широтно-импульсного модулятора PWM (Pulse Width Modulation). Широтно-импульсная модуляция (ШИМ) – один из способов формирования импульсного сигнала с регулируемыми временными характеристиками. Способ широтно-импульсной модуляции часто используется для регулирования скорости вращения двигателей постоянного тока, а также для управления электрическими двигателями других типов. Для генерации ШИМ сигнала в МК В32 могут быть использованы аппаратные средства модуля многофункционального таймера TIM. Однако МК В32 оснащен специальным модулем ШИМ. Этот модуль позволяет генерировать четыре независимых импульсных последовательности с 8 разрядным разрешением для задания коэффициента заполнения, или две импульсные последовательности с 16 разрядным заданием коэффициента заполнения. Допускается комбинация этих режимов. Например, ШИМ сигнал используется для управления двигателем рулевого управления в игрушечных радиоуправляемых машинках. Для того, чтобы машинка повернула направо или налево, она должна получить импульсный сигнал, у которого частота следования импульсов постоянная, а длительность импульсов изменяется, как показано на рис. 4.2. Отношение длительности импульса к длительности периода сигнала называется коэффициентом заполнения. Машинка повернет налево, если коэффициент заполнения менее 50% , или направо, если коэффициент заполнения превышает 50%. Модуль PWM микроконтроллера В32 позволяет организовать импульсную последовательность с требуемыми значениями

периода следования и коэффициента заполнения при использовании минимального числа команд прикладной программы.

Модуль контроллера CAN интерфейса msCAN12 (Motorola Scalable Controller Area Network) содержит в себе набор аппаратных средств для поддержки коммуникационного протокола промышленных сетей в стандарте CAN 2.0 A/B.

### Задание

Программирование модуля аналого-цифрового преобразователя ATD (Analog To Digital conversion system). Перед началом выполнения практической части необходимо ознакомиться с лабораторной установкой на базе платы Freescale и программным обеспечением CodeWarrior. Для того чтобы воспользоваться модулем аналого-цифрового преобразования ATD для измерения уровня напряжения на нескольких аналоговых входах МК, необходимо выполнить следующие действия:

1) Подключить источники стабилизированного напряжения ко входам опорного напряжения VRF и VRL . Необходимо помнить, что напряжение на входе высокого уровня опорного напряжения VRF не должно превышать 5,0 В, а на входе низкого уровня VRL напряжение должно быть не менее 0 В. Кроме того, разность напряжений на входах VRF и VRL, равная напряжению полной шкалы АЦП  $UREF = URH - URL$  , не должна быть менее 2,5 В;

2) Подключить источники измеряемых аналоговых сигналов ко входам AN0...AN7. Напряжение измеряемых сигналов должно находиться в диапазоне от 0 В до 5,0 В;

3) Осуществить внутреннюю коммутацию напряжения питания к модулю ATD. Для этого записать 1 в бит ADPU регистра управления ATDCTL2. Адрес регистра – \$0062;

4) Выдержать паузу в 100 мкс для завершения переходных процессов в модуле ATD. В рассматриваемом ниже программном фрагменте мы покажем, как организовать такую задержку;

5) Назначить режим работы модуля ATD посредством записи необходимых слов инициализации в управляющие регистры модуля;

6) Запустить измерительную последовательность посредством записи в регистр управления ATDCTL5;

7) Контролировать ход преобразования, используя флаги регистра состояния модуля ATDSTAT;

8) Когда измерительная последовательность будет завершена, считать данные из регистров результата ADR0H...ADR7H в память МК.

Программный фрагмент voltmeter.c производит измерение аналогового сигнала на входе AN6. Измерительная последовательность состоит из четырех преобразований. Режим измерения однократный. Результаты четырех последовательных преобразований одного и того же сигнала располагаются в четырех регистрах результата ADR0H...ADR3H. Эти измерения усредняются, что позволяет снизить влияние шумов. Полученный 8 разрядный двоичный код преобразуется к истинному значению измеряемого напряжения, умноженному на 100. Умножение на нормирующий коэффициент (100 в десятичной системе счисления) необходимо, чтобы использовать в программе целочисленные форматы представления данных. Полученный результат содержит одну десятичную цифру целой части измеренного напряжения, это единицы Вольт. А также две цифры десятичной дробной части. Это десятые и сотые доли Вольт в представлении результата. Промежуточные результаты исполнения программы, а также измеренное напряжение выводятся на экран персонального компьютера.

/\* \_\_\_\_\_ \*/  
/\* filename: voltmeter.c \*/

```

/* В этом примере реализован простейший цифровой вольтметр. */
/* Программа выполняет одно измерение и выводит данные на экран */
/* персонального компьютера. Для того, чтобы выполнить следующее измерение*/
/* следует перезапустить программу */
/* _____ */
/*подключаемые файлы*/
#include<912b32.h>
#include<stdio.h>
#define DECIMAL 0x2E /*определить код точки на экране*/
#define V 0x56 /*определить код символа "V" для экрана*/
/*используемые функции*/
void delay_100us(void);
void ADC_convert(void);
void delay_5ms (void);
void main(void)
{
printf("HELLO\n"); /*вывести приветствие на экран*/
ATDCTL2 = 0x80; /*включить питание модуля, запретить */
/*прерывания от модуля*/
printf ("ADC \n" );
delay_100us(); /*задержка 100мкс*/
printf ("warmed up\n");
ATDCTL3 = 0x00; /*обеспечить доступ к модулю в отладочном режиме*/
ATDCTL4 = 0x01; /*установить 2 такта для времени выборки*/
/*и коэффициент деления, равный 4*/
printf ( "ready\n" );
ADC_convert(); /*реализовать цифровой вольтметр*/
}
/* _____ */

```

### Контрольные вопросы

- 1) Назначение процессорного ядра микроконтроллеров.
- 2) Основные аппаратные средства современных МК.
- 3) Дополнительные аппаратные средства современных МК.
- 4) Особенности микроконтроллеров на примере МК семейства Motorola.
- 5) Многофункциональный таймер.
- 6) Модуль контроллера CAN интерфейса msCAN12 (Motorola Scalable Controller Area Network).
- 7) Модули контроллеров последовательных интерфейсов SPI (Serial Peripheral Interface) и SCI (Serial Communication Interface).
- 8) Модуль аналого-цифрового преобразователя ATD (Analog To Digital conversion system).
- 9) Модуль широтно-импульсного модулятора PWM (Pulse Width Modulation).

## Лабораторная работа №2

### Структура процессорного ядра микроконтроллера

Цель работы – ознакомление с внутренней структурой современных микроконтроллеров. Изучение основных функциональных блоков на примере МК семейства Motorola.

#### Общие сведения

Все МК обладают процессорным ядром. Семейство объединяет ряд моделей. Отдельные модели в составе семейства различаются набором периферийных модулей, которые подключаются к внутренней межмодульной магистрали. Основные отличия между отдельными представителями семейства состоят в типе и объеме размещенной на кристалле резидентной памяти, количестве параллельных портов и контроллеров последовательных интерфейсов. Структура современных микроконтроллеров на примере МК MC68HC912B32 приведена на рис. 2.

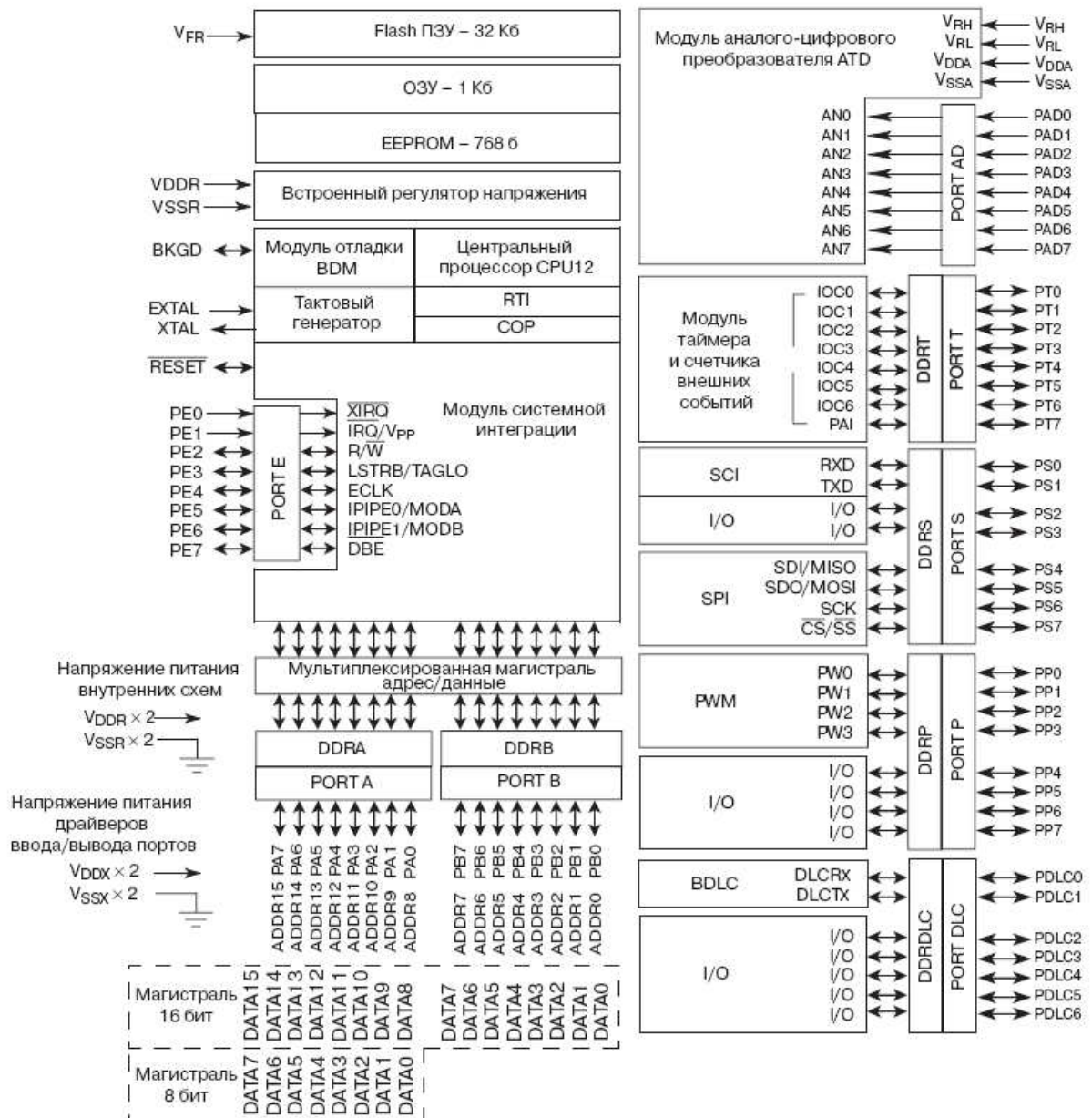


Рис.2. Структура современных микроконтроллеров

### Отличия современных микроконтроллеров на примере МК Motorola

**Низкое энергопотребление.** Микроконтроллеры семейства 68HC12 производятся на основе CMOS технологии (CMOS – Complementary Metaloxide semiconductor). Эта технология позволяет создать транзисторные структуры с относительно низкими потерями энергии при работе в ключевых режимах. Поэтому МК семейства 68HC12 характеризуются малым потреблением энергии, что позволяет рекомендовать их для использования в изделиях с автономным питанием (от аккумуляторов или батареек). Однако не следует забывать, что потребляемая энергия для полупроводниковых CMOS структур прямопропорциональна частоте переключения. Поэтому для достижения оптимальных энергетических характеристик следует выбирать частоту тактирования центрального процессора микроконтроллеров 68HC12 минимально возможной для конкретного применения. Микроконтроллеры 68HC12 имеют специальные режимы пониженного энергопотребления, которые также позволяют оптимизировать энергетические характеристики проектируемого изделия.

**Высокопроизводительное 16 разрядное процессорное ядро.** Центральный процессор семейства 68HC12 выполняет действия над 16 разрядными операндами, поэтому время выполнения алгоритмов над переменными, разрядность которых превышает байт, существенно сокращается по сравнению с 8 разрядными МК. Максимальная частота тактирования процессорного ядра fBUS для микроконтроллеров семейства 68HC12 составляет 8 МГц. Однако частота внешнего генератора или кварцевого резонатора должна составлять 16 МГц, поскольку внутренние цепи микроконтроллера делят эту частоту на два для получения fBUS.

Новое семейство МК HCS12 унаследовало архитектуру процессорного ядра и большинства периферийных модулей от своего предшественника, семейства 68HC12. Отметим основные отличия:

- 1) Напряжение питания большинства моделей МК семейства HCS12 равно 5,0 В, что позволяет обеспечить электромагнитную совместимость в авто мобильных и общепромышленных применениях;
- 2) Повышенна производительность процессорного ядра. Частота тактирования центрального процессора и межмодульных магистралей МК семейства HCS12 составляет 25 МГц против 8 МГц у HC12;
- 3) Увеличен объем резидентной памяти. Объем встроенного в МК семейства HCS12 ОЗУ достигает 12 Кб, объем Flash ПЗУ – 512 Кб. Кроме того, в составе большинства моделей МК имеется значительная область EEPROM (до 4 Кб) для хранения перепрограммируемых констант пользователя;
- 4) Большое число интегрированных на кристалл разнообразных контроллеров последовательных интерфейсов, т.к. МК семейства предназначены для работы в качестве интеллектуальных узлов распределенных систем управления.

### Функциональные блоки микроконтроллеров

Семейство HCS12 объединяет ряд моделей МК с одинаковым процессорным ядром CPU HCS12. Отдельные представители семейства различаются объемом встроенной памяти и количеством и типом интегрированных на кристалл МК периферийных модулей. Однако каждый МК из семейства HCS12 имеет в своем составе следующие функциональные модули:

- 1) Память трех типов: FLASH память программ, энергонезависимая память EEPROM для хранения изменяемых констант пользователя и статическое ОЗУ для размещения промежуточных переменных прикладной программы управления;
- 2) Многофункциональный 16 разрядный таймер с 8 каналами IC/OC;
- 3) Многоканальный аналого-цифровой преобразователь;

- 4) Контроллеры последовательного обмена нескольких стандартов;
- 5) Модуль ШИМ общего назначения, ряд моделей оснащен специализированным модулем ШИМ для управления автономными вентильными преобразователями.

### Режимы работы микроконтроллеров

Микроконтроллеры семейства 68HC12/HCS12 функционируют в одном из восьми режимов, которые делят на две группы: рабочие режимы и специальные режимы. Рабочие режимы позволяют создать различную аппаратную реализацию встраиваемого контроллера, в то время как специальные режимы работы предназначены для проведения тестовых испытаний и диагностики МК в процессе производства. Поэтому инженерам по применению микроконтроллеров важно изучить лишь группу рабочих режимов. Каждый режим из группы рабочих задает собственное распределение адресного пространства МК и конфигурацию магистралей для подключения внешней памяти. Режим работы МК назначается посредством комбинации логических сигналов на входах BKGD, MODB, MODA микроконтроллера в состоянии начального запуска МК. Состояние начального запуска именуют также состоянием сброса (Reset). Сразу после выхода из состояния сброса МК запоминает кодовую комбинацию на перечисленных входах и переходит в соответствующий режим работы. Там же указаны альтернативные функции линий портов PORT A и PORT B, которые они приобретают в каждом из режимов работы.

В большинстве проектируемых устройств Вы будете использовать МК 68HC12/HCS12 в одном из трех рабочих режимов:

- Однокристалльном или автономном режиме;
- Расширенном режиме с 16 разрядной внешней шиной;
- Расширенном режиме с 8 разрядной внешней шиной.

Расширенные режимы работы предоставляют возможность подключения к МК внешней памяти и внешних периферийных ИС с использованием параллельных магистралей адреса и данных. Сигналы магистралей формируются на линиях портов PORTA и PORTB, поэтому использование соответствующих выводов МК в качестве линий ввода/вывода общего назначения в расширенных режимах работы становится невозможным.

### Краткое описание рабочих режимов

Однокристалльный режим работы (BKGD: 1, MODB: 0, MODA: 0) обеспечивает функционирование МК с использованием только внутренней памяти. Поэтому коды прикладной программы управления и ее переменные должны размещаться только во внутреннем ПЗУ и ОЗУ МК.

Порты PORTA и PORTB используются в качестве обычных двунаправленных портов ввода/вывода. Подключение внешних периферийных ИС должно производиться с использованием последовательных интерфейсов или с программной поддержкой временной диаграммы обмена на линиях портов ввода/вывода.

Расширенный режим с 16 разрядной системной шиной (BKGD: 1, MODB: 1, MODA: 1) обеспечивает функционирование МК с использованием как внутренней, так и внешней памяти. Для подключения внешней памяти предназначена 16-разрядная мультиплексированная магистраль адрес/данные ADDR15\_0/DATA15\_0. При этом старший байт мультиплексированной во времени магистрали ADDR15\_8/DATA15\_8 формируется на линиях PORTA, младший байт ADDR7\_0/DATA7\_0 – на линиях PORTB.

Расширенный режим с 8 разрядной системной шиной (BKGD: 1, MODB: 0, MODA: 1) также реализует работу МК с использованием внутренней и внешней памяти. Но для

подключения внешней памяти предназначены 16 разрядная магистраль адреса ADDR15\_0 и 8 разрядная магистраль данных DATA7\_0. Старший байт магистрали адреса ADDR15\_8 выводится на PORTA, младший байт ADDR7\_0 – на PORTB. Двухнаправленная 8 разрядная магистраль данных DATA7\_0 использует линии порта PORTA в мультиплексированном со старшими разрядами магистрали адреса режиме. В обоих расширенных режимах некоторые линии порта PORTE используются для передачи сигналов управления обменом по шине.

### Задание

Программирование модуля широтно-импульсного модулятора PWM (Pulse Width Modulation). Перед началом выполнения практической части необходимо ознакомиться с лабораторной установкой на базе платы Freescale и программным обеспечением CodeWarrior. Рассмотрим последовательность действий, которую необходимо проделать для инициализации модуля PWM:

1) Для конкретного приложения следует определить разрешающую способность генерируемого ШИМ сигнала, т.е. число дискретных отсчетов частоты тактирования канала в периоде и длительности импульса выходного сигнала канала. На основании полученных данных следует определить, в каком режиме, 8\_разрядном или 16\_разрядном, Вы будете использовать каналы модуля PWM;

2) Для конкретного приложения следует определить требуемую частоту генерируемого ШИМ сигнала. На основе полученных данных определить структуру подсистемы тактирования каналов модуля PWM;

3) Установить биты CON23 и CON01 в регистре PWCLK для выбора 8 разрядного или 16 разрядного режима работы;

4) Определить, будете ли Вы использовать режим центрированной или режим фронтной ШИМ. В соответствии с выбранным режимом установить бит CENTR в регистре PWCTL;

5) Определить активный уровень ШИМ сигнала, в соответствии с выбором установить биты PPOL0...PPOL3 в регистре PWPOL;

6) Назначить источники тактирования для каналов, для чего установить биты PCLK0...PCLK3 в регистре PWPOL;

7) Установить коэффициенты деления для импульсных последовательностей CLOCK\_A и CLOCK\_B, используя для этого биты PSCA2...PSCA0 и PCKB2...PCKB0 регистра PWCLK;

8) Установить для используемых каналов значения регистров периода и регистров коэффициента заполнения;

9) Разрешить работу выбранных каналов модуля PWM, используя для этого биты PWEN0...PWEN3 регистра EPWM.

Определим параметры настройки модуля PWM для генерации ШИМ сигнала с частотой 976 Гц и коэффициентом заполнения 66,7%. Частоту тактирования канала ШИМ выберем равной  $8\text{ МГц}/32 = 250\text{ кГц}$ . Этой частотой будем тактировать 8\_разрядный счетчик канала. Для формирования частоты 976 Гц потребуется 256 отсчетов частоты 250 кГц, что соответствует максимально возможному коэффициенту счета 8\_разрядного счетчика периода канала. Для формирования сигнала с коэффициентом заполнения 66,7% следует установить код периода, равный 256 отсчетам, а код коэффициента заполнения – 171 отсчету. Для формирования ШИМ сигнала будем использовать канал 0 модуля PWM. Программный фрагмент `init_pwm.c` производит начальную установку регистров специальных функций модуля PWM для генерации на выходе PPO ШИМ сигнала с частотой 976 Гц и коэффициентом заполнения 66,7%:

```

/* _____ */
/* Функция init_pwm задает начальные установки модуля PWM */
/* _____ */
void ini_pwm(void)
{
    PWTST = 0x00; /*выбрать нормальный режим работы модуля PWM*/
    PWCTL = 0x00; /*выбрать режим фронтовой ШИМ*/
    PWCLK = 0x28 /*канал 0 в 8_разрядном режиме, коэфф. деления*/
    /* частоты E_CLOCK равен 32*/
    PWPOL = 0x01; /*установить высокий активный уровень сигнала*/
    DDRP = 0xFF; /*настроить порт P на вывод*/
    PWEN = 0x01; /*разрешить работу канала 0 модуля PWM */
    PWPER0 = 255; /*установить код периода*/
    PWDY0 = 171 /*установить код коэффициента заполнения*/
}
/* _____ */

```

### Контрольные вопросы

- 1) Внутренняя структура микроконтроллеров.
- 2) Отличия современных МК.
- 3) Функциональные блоки МК.
- 4) Режимы работы МК.
- 5) Однокристалльный режим работы.
- 6) Расширенный режим с 16 разрядной системной шиной.
- 7) Расширенный режим с 8 разрядной системной шиной.
- 8) Назначение модуля широтно-импульсного модулятора PWM (Pulse Width Modulation).

## Лабораторная работа №3

### *Память программ и данных микроконтроллера*

Цель работы – ознакомление с видами памяти микроконтроллеров: память данных, память программ. Изучение регистров и стековой памяти МК.

#### Общие сведения

Подсистема памяти МК семейства 68HC12/HCS12 включает четыре различных модуля памяти: энергонезависимая Flash память программ, энергонезависимая EEPROM память данных, статическое ОЗУ и блок регистров специальных функций для управления режимами работы периферийных модулей. Расположение различных модулей памяти в адресном пространстве МК принято отражать на так называемой карте памяти. Карта памяти МК В32 представлена на рис. ниже. Указанные на ней адреса будут действительны при выходе МК из состояния сброса. В ходе исполнения прикладной программы адресное пространство для каждого модуля памяти может быть изменено. Тогда для обращения к ячейкам памяти будут использоваться не указанные на рис. физические адреса, а измененные виртуальные адреса.

#### Память программ и память данных микроконтроллера на примере МК семейства Motorola

Микроконтроллер В32 предназначен для использования преимущественно в однокристальном режиме работы. Он содержит в себе 32Кб ПЗУ программ, 768 байт памяти типа EEPROM, 1Кб статического ОЗУ и 512 регистров управления. Строго говоря, аббревиатура EEPROM (Electrically Erasable Programmable ROM) обозначает энергонезависимую память с электрическим программированием и электрическим стиранием. Поэтому и резидентная память программ, выполненная на основе технологии FLASH, и энергонезависимая память данных должны быть характеризованы как EEPROM. Однако энергонезависимая память программ и энергонезависимая память данных отличаются по своим свойствам не только на уровне технологии изготовления, но и на уровне разработчика встраиваемых систем. Память типа Flash допускает выполнение операции стирания только над некоторым множеством ячеек, что неудобно при необходимости замены только одного байта информации. Энергонезависимая память данных позволяет стереть и потом запрограммировать один байт информации. Однако ячейки памяти с подобным свойством занимают значительную площадь полупроводникового кристалла МК, поэтому на их основе не может быть выполнена память программ большого объема. Во избежание путаницы у российских разработчиков принято использовать аббревиатуру EEPROM только для памяти с побайтным стиранием и побайтным программированием. А память со стиранием блоками обозначают как Flash, хотя и эта память по своим свойствам относится к EEPROM. Используя МК от Freescale Semiconductor, в частности семейство 68HC12/HCS12 следует знать, что гарантированное число циклов стирания/программирования для МК HCS12 составляет 10000, а для МК 68HC12 – всего 100. Именно поэтому во многих отладочных средствах на основе МК 68HC12 рекомендуется промежуточные версии программы записывать и исполнять из ОЗУ. Поскольку резидентное ОЗУ у микроконтроллеров обладает недостаточным объемом для размещения программы, то многие отладочные платформы используют расширенный режим работы МК с подключением внешнего ОЗУ. В МК семейства HCS12 для целей отладки обычно используется внутреннее Flash ПЗУ программ, т.к. 10000 циклов перезаписи обычно достаточно для внесения всех исправлений в процессе отладки.

*Пример применения*

В процессе эксплуатации память типа EEPROM часто используют для создания счетчиков аварийных ситуаций на объекте. Возможные аварии предварительно классифицируются, в системе устанавливаются датчики, которые позволяют микроконтроллеру отнести возникшую аварийную ситуацию к тому или иному типу. Если тип аварии диагностирован, то МК увеличивает соответствующий счетчик и запоминает его новое состояние в энергонезависимой памяти данных типа EEPROM. Такое решение позволяет сохранить информацию об авариях даже при отключении системы питания.

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \$0000 | Регистры специальных функций                                                                                                                                                                                                                                                                                                                                                                                                                     |
| \$01FF |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \$0800 | ОЗУ 1 Кб                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \$0BFF | <ul style="list-style-type: none"> <li>● Код/данные пользователя (\$0800-\$09FF)</li> <li>● Резервная область для D-Bug12 (\$0A00-\$0BFF)</li> </ul>                                                                                                                                                                                                                                                                                             |
|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \$0D00 | EEPROM 768 байт                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \$0FFF | <ul style="list-style-type: none"> <li>● Код/данные пользователя</li> </ul>                                                                                                                                                                                                                                                                                                                                                                      |
|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| \$8000 | FLASH ПЗУ программ 32 Кб                                                                                                                                                                                                                                                                                                                                                                                                                         |
| \$FFFF | <ul style="list-style-type: none"> <li>● код D-Bug12 (\$8000-\$F67F)</li> <li>● Область, доступная пользователю (\$F6C0-\$F6FF)</li> <li>● Настройка функций D-Bug12 (\$F680-\$F6BF)</li> <li>● Код запуска D-Bug12 (\$F700-\$F77F)</li> <li>● Вектора прерывания (\$F780-\$F7FF)</li> <li>● Расширение загрузчика (\$F800-\$FBFF)</li> <li>● Загрузчик EEPROM (\$FC00-\$FFBF)</li> <li>● Вектора сброса и прерывания (\$FFC0-\$FFFF)</li> </ul> |

Карта памяти микроконтроллера

#### Регистры специальных функций

Блок регистров специальных функций микроконтроллера по реализуемым функциям очень похож на пульт управления. Каждый регистр осуществляет управление или отображает состояние того периферийного модуля МК, к которому он отнесен в техническом описании. Часть битов регистров специальных функций может быть сопоставлена с переключателями: установка бита в 1 разрешает реализацию, какой либо функции периферийного модуля, сброс в 0 – прекращает исполнение этой функции. Другая часть битов регистров аналогична индикаторам состояния: установка 1 свидетельствует о завершении выполнения назначенной функции, пребывание в 0 свидетельствует о том, что процесс еще не завершился. Регистры специальных функций МК семейства 68HC12/HCS12 объединены в блок, который размещается в адресном пространстве памяти МК. Доступ к каждому из 512 регистров блока осуществляется теми же командами центрального процессора, что и к ячейкам памяти. Никакого различия с точки зрения программного обслуживания между регистрами специальных функций и ячейками ОЗУ не существует. Исключение составляют лишь регистры конфигурации и другие, так называемые защищенные регистры, состояние которых невозможно изменить

в произвольный момент исполнения прикладной программы. В момент начального запуска, когда МК находится в состоянии сброса, в каждый регистр специальных функций записывается определенное в техническом описании значение. Начальные состояния регистров специальных функций обычно таковы, что функции соответствующего периферийного модуля отключены. Это очень удобно как с точки зрения потребления энергии микроконтроллером, так и с точки зрения программиста, составляющего прикладную программу управления. Если при запуске большинство периферийных модулей не работает, то МК потребляет минимальную энергию. Далее будут активизированы только те модули, которые потребуются для выполнения управляющего алгоритма. Остальные модули останутся не задействованными, что обеспечит минимизацию энергии потребления в процессе функционирования конечного устройства. При составлении программы управления МК сначала предстает перед разработчиком в минимальной конфигурации, когда большинство периферийных модулей как бы не существует. Таким МК легко управлять, поэтому ошибок в прикладной программе будет меньше. По мере реализации программы управления разработчик активизирует работу только тех модулей, которые ему потребуются. Поэтому ему не придется думать о программном обслуживании не задействованных при реализации алгоритма периферийных модулей. Каждый регистр специальных функций имеет свой собственный адрес в карте памяти МК. При программировании на ассемблере изменение значений регистров специальных функций и считывание их состояния может быть произведено командами загрузки и пересылки данных, такими как LDAA, STAA, MOV, а также командами битовых операций BSET, BCLR. Для того чтобы избавить программиста от запоминания физических адресов регистров специальных функций, следует использовать псевдокоманду операции присваивания EQU языка Ассемблер. Эта псевдокоманда назначит символьному имени регистра, которое одинаково для всех моделей МК семейства 68HC12/HCS12, абсолютный адрес. И при записи исходного текста программы можно будет использовать только символьные имена регистров специальных функций. Использование символьных имен также полезно при переносе прикладной программы с одной модели МК на другую. В этом случае следует заменить только абсолютные адреса в псевдокомандах, в то время, как исходный текст программы останется неизменным.

### Задание

Перед началом выполнения практической части необходимо ознакомиться с лабораторной установкой на базе платы Freescale и программным обеспечением CodeWarrior. В практической части данной работы рассматривается программирование регистров модуля АЦП.

Регистр ATDCTL2, расположенный в памяти МК по адресу 0x0062, управляет работой модуля аналого\_цифрового преобразователя АТД. Бит 7 этого регистра разрешает (1) или запрещает (0) работу модуля. Этот бит, обозначенный в техническом описании как ATPU (ATD Powerup Bit), устанавливается в 0 в состоянии сброса МК. Для включения модуля АТД в работу необходимо программно установить этот бит в 1. На языке Ассемблера это действие может быть произведено следующим фрагментом программы:

```
;
;_____
;MAIN PROGRAMM
;
;_____
ATDCTL2 EQU $0062 ;определить адрес регистра в МК
ATD_INI EQU $80 ;определить значение слова
;инициализации для включения АЦП
LDAA #ATD_INI ;загрузить слово инициализации в ACC
STAA ATDCTL2 ;записать значение инициализации в
```

;регистр управления АЦП

;

---

Первые две строки программы содержат директивы присвоения EQU языка Ассемблер, которые информируют программу о том, что вместо имени ATDCTL2 в кодах программы следует использовать значение \$0062, а вместо ATD\_INI – \$80 или 10000000 в двоичном коде. Команда LDAA использует непосредственную адресацию, в результате ее выполнения число \$80 загрузится в аккумулятор АСС. Команда STAA с прямой адресацией пересылает содержимое АСС в ячейку с адресом ATDCTL2. В результате, код в регистре управления ATDCTL2 будет равен 10000000, т.е. установленный в 1 бит 7 вызовет включение модуля АЦП. На языке Си аналогичное действие выполняет следующий программный фрагмент:

```
/* _____ */
/*MAIN PROGRAMM
/* _____ */
#include<912b32.h>
void main(void)
{
  unsigned char ATD_INI = 0x80;
  ATDCTL2 = ATD_INI;
}
/* _____ */
```

Сравнить преимущества написания программ на С и на Ассемблер.

#### Контрольные вопросы

- 1) Структура памяти микроконтроллеров.
- 2) Память программ.
- 3) Память данных.
- 4) Энергозависимая память.
- 5) Энергонезависимая память.
- 6) Регистры специальных функций МК.
- 7) Регистры управления модулем АЦП.
- 8) Стековая память.
- 9) Преимущества программирования на языке Си и Ассемблер.

## Лабораторная работа №4

### *Порты ввода/вывода*

Цель работы – ознакомление с устройством ввода/вывода микроконтроллеров. Изучение спецификации портов на примере МК Motorola.

#### Общие сведения

Все МК семейства 68HC12 имеют некоторое количество линий ввода/вывода данных. Линии объединены в 8 разрядные параллельные порты данных: Port A, Port B, Port E, PORT AD и т.д.

За редким исключением, все линии ввода/вывода двунаправленные. Направление передачи линий ввода/вывода настраивается программно путем записи управляющего слова в регистр направления передачи соответствующего порта. Возможно изменение направления передачи в ходе выполнения программы посредством перепрограммирования этих регистров. Сигнал сброса устанавливает все двунаправленные линии в режим ввода. Следует особо подчеркнуть, что направление передачи каждой линии может быть выбрано разработчиком произвольно, независимо от других линий, принадлежащих к одному и тому же порту ввода/вывода. Исключения составляют лишь линии однонаправленной передачи, которые изначально специализированы на ввод или на вывод. Часть линий ввода/вывода имеют так называемую альтернативную функцию, т.е. обеспечивают связь встроенных периферийных модулей МК с «внешним миром». Так линии порта PORT AD используются для подключения к встроенному АЦП измеряемых напряжений, линии порта PORT S служат входами и выходами контроллеров последовательного обмена. Если соответствующий периферийный модуль МК не используется, то его выводы можно задействовать как обычные линии ввода/вывода. Если линии порта двунаправленные, то для его обслуживания такого порта предусмотрены два типа регистров:

PORTx – регистр данных порта x, где x – имя порта ввода/вывода;

DDRx – регистр направления передачи порта x.

Например, порт PORT A обслуживается регистрами PORTA и DDRA, а порт PORT B – регистрами PORTB и DDRB. Если порт имеет схмотехнику с программно подключаемым «подтягивающим» резистором (R pull-up), то для обслуживания такого порта предусмотрен дополнительный регистр входного сопротивления порта. Ниже приведен фрагмент текста программы, которая конфигурирует PORT T для вывода данных, а затем записывает в порт число S62. Для того, чтобы все линии порта PORT T стали линиями вывода, необходимо записать в регистр направления передачи DDRT код \$FF.

#### Спецификация портов ввода/вывода

Подсистема параллельного ввода/вывода МК B32 состоит из 8 портов, причем линии многих портов обладают альтернативной функцией. Мы рассмотрим эти порты в порядке их расположения на рис. 4.1. по часовой стрелке.

1)PORT A. В однокристальном режиме работы – 8 разрядный порт ввода/вывода общего назначения. Направление передачи каждой линии порта определяется соответствующим битом регистра DDRA. В расширенном режиме работы на линиях порта формируются сигналы старшего байта мультиплексированной магистрали адрес/данные ADDR15\_8/DATA15\_8. В расширенном режиме с 8 разрядной шиной линии порта представляют собой мультиплексированную магистраль ADDR15\_8/DATA7\_0.

2) PORT B. В однокристальном режиме работы – 8 разрядный порт ввода/вывода общего назначения. Направление передачи каждой линии порта определяется соответствующим битом регистра DDRB. В расширенном режиме работы на линиях порта формируются сигналы младшего байта мультиплексированной магистрали адрес/данные ADDR7\_0/DATA7\_0. В расширенном режиме с 8 разрядной шиной линии порта представляют собой немultipлексированную магистраль ADDR7\_0.

3) PORT E. 8\_разрядный порт ввода/вывода общего назначения. Две линии порта PE0 и PE1 – однонаправленные и работают только на ввод. Остальные линии порта – двунаправленные, направление передачи линий PE2 и PE7 определяется соответствующими битами регистра DDRE. Все линии порта имеют альтернативную функцию. Линии PE1 и PE0 могут использоваться как входы внешних прерываний IRQ и XIRQ. Остальные линии служат для задания режимов работы МК (MOD A и MOD B) и в качестве сигналов управления внешней шиной при работе МК в расширенном режиме.

4) PORT AD. Однонаправленный 8\_разрядный порт ввода. Все линии имеют альтернативную функцию. Если работа модуля аналого-цифрового преобразователя ATD разрешена, то линии порта используются для подключения измеряемых аналоговых сигналов.

5) PORT T. Двунаправленный 8 разрядный порт ввода/вывода общего назначения. Направление передачи каждой линии порта определяется соответствующим битом регистра DDRT. Альтернативная функция линий порта PORT T – обслуживание модуля таймера. Если работа таймера разрешена, то линии используются в качестве входов входного захвата IC или выходов выходного сравнения OC.

6) PORT S. Двунаправленный 8 разрядный порт ввода/вывода общего назначения. Направление передачи каждой линии порта определяется соответствующим битом регистра DDRS. Альтернативная функция линий порта PORT S – обслуживание модулей последовательного обмена SCI и SPI.

7) PORT P. Двунаправленный 8 разрядный порт ввода/вывода общего назначения. Направление передачи каждой линии порта определяется соответствующим битом регистра DDRP. Четыре линии порта PORT P могут использоваться в качестве выходов модуля генератора ШИМ сигнала (модуль PWM), если работа последнего программно разрешена.

8) PORT DLC. Двунаправленный 8\_разрядный порт ввода/вывода общего назначения. Направление передачи каждой линии порта определяется соответствующим битом регистра DDRDLC. Альтернативная функция двух линий порта PORT DLC – обслуживание модуля последовательного обмена в стандарте BDLC.

### Регистры управления портами

В МК семейства 68HC12/HCS12 каждый двунаправленный порт ввода/вывода общего назначения обслуживается двумя регистрами специальных функций. Это регистр данных порта и регистр направления передачи DDRx (вместо буквы «x» следует подставить буквенное обозначение порта). Если линии порта общего назначения настроены на ввод, то операция чтения регистра данных возвращает состояние выводов корпуса МК, с которыми связан порт. Если порт настроен на вывод, то операция записи в регистр данных устанавливает на выводах корпуса МК, связанных с портом, соответствующие логические уровни.

Регистр DDRx определяет направления передачи каждой линии порта независимо от других линий этого же порта. Если какой либо бит регистра DDRx равен 0, то соответствующая линия настраивается на ввод, если 1 – то на вывод. Возможны решения, при которых часть линий одного и того же порта настроена на ввод, а часть на вывод. Например, при значении DDRx=10110010 линии D6, D3, D2 и D0 развернуты на ввод, а

линии D7, D5, D4 и D1 – на вывод. В состоянии сброса МК все биты регистров направления передачи DDRx сбрасываются, поэтому сразу после включения питания все линии портов МК конфигурированы как входы с высоким входным сопротивлением.

### Задание

Перед началом выполнения практической части необходимо ознакомиться с лабораторной установкой на базе платы Freescale и программным обеспечением CodeWarrior.

В практической части мы рассмотрим работу порта PORTA. Для этого подключим группу зеленых и красных светодиодов к выводам этого порта. Схема подключения показана на рис. 3 ниже. Договоримся, что если на линии порта установлена 1, то будет гореть зеленый светодиод, если логический 0 – то красный светодиод. А если линия порта переведена в состояние ввода, т.е. она представляется для цепи светодиодов нагрузкой с высоким входным сопротивлением, то оба светодиода окажутся погашенными. На рис. 4 приведена блок-схема алгоритма программы, которая зажигает на 30 мс зеленым цветом светодиоды с четными номерами и одновременно красным цветом светодиоды с нечетными номерами. Следующие 30 мс светодиоды «меняются цветами», далее этот процесс продолжается до бесконечности. Ниже приведен текст программы на языке Си, который реализует этот алгоритм.

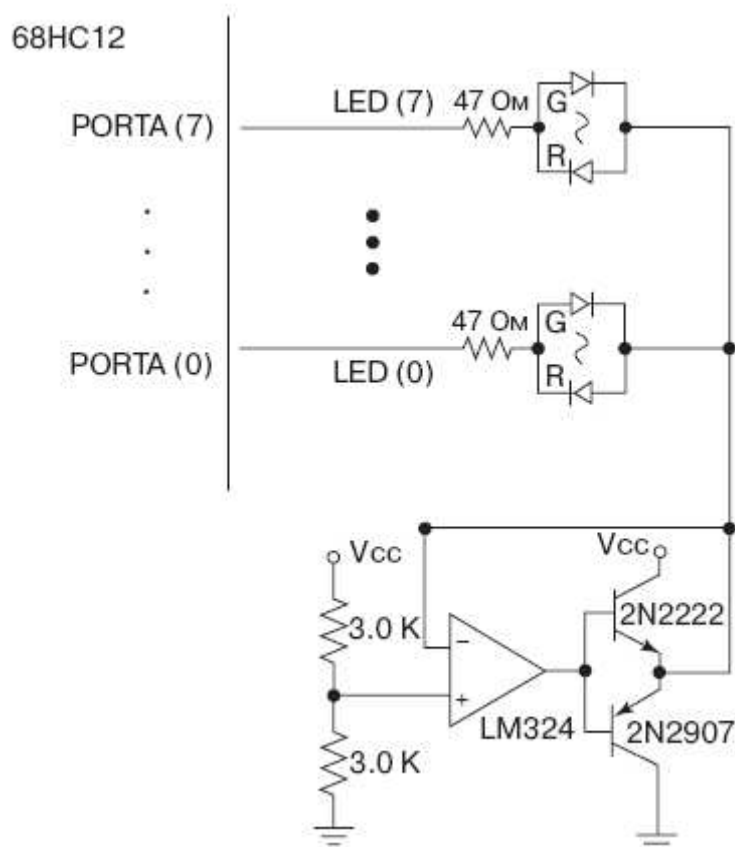


Рис.3. Схема отображения состояния порта PORTA



Рис.4. Блок-схема алгоритма управления светодиодами

```

/* _____ */
/* MAIN PROGRAM: Эта программа «зажигает» на выходах порта PORTA */
/*30 мс зеленым цветом горят светодиоды на выходах порта с четными */
/*номерами, красным цветом – светодиоды на выходах порта с нечетными */
/* номерами. Следующие 30 мс на месте зеленых горят красные, и наоборот */
/* _____ */
/*подключаемые файлы*/
#include<912b32.h>
/*используемые функции*/
void delay_100us(void)
void delay_30ms(void)
void main{void)
{
DDRA = 0xFF //установить порт PORTA на вывод
while(1)
{
PORTA = 0x55;
delay_30ms();
PORTA = 0xAA;

```

```

delay_30ms();
}
}
/* _____ */
/* Функция delay_30ms формирует задержку в 30мс мкс, частота тактирования*/
/* межмодульных магистралей МК составляет 8 МГц */
/* _____ */
void delay_30ms(void)
{
int i;
for (i=0; i<=299; i++)
delay_100us();
}
/* _____ */
/* Функция delay_100us формирует задержку в 100 мкс, частота тактирования */
/* межмодульных магистралей МК составляет 8 МГц */
/* _____ */
void delay_100us(void)
{
int j;
for (j=0; j<50; j++)
{
asm(«nop»\n) ;
}
}
/* _____ */

```

Обратите внимание, что функция задержки на 30 мс использует вложенную функцию задержки на 100 мкс. В приведенном тексте программы для формирования задержки на 100 мкс используются 50 циклов повторения операторов функции delay\_100us. Выбор числа повторений производился из предположения, что данная программа будет исполняться микроконтроллером, частота внутренней шины которого составляет 8 МГц. По результатам рассмотрения файла в формате \*.lst было установлено, что команды ассемблера, соответствующие одному повторению цикла функции delay\_100us, реализуются за 16 машинных циклов. При частоте шины в 8 МГц для формирования временного интервала в 100 мкс потребуется 800 машинных циклов. Поэтому число повторений цикла функции delay\_100us должно составлять  $800/16 = 50$ . Если бы эта программа исполнялась бы микроконтроллером DP256, частота внутренней шины которого составляет 25 МГц, то число циклов функции delay\_100us должно было бы быть увеличено до 156.

#### Контрольные вопросы

- 1) Назначение устройств ввода/вывода информации.
- 2) Типы портов и алгоритмы обмена информацией.
- 3) Направление работы портов.
- 4) Спецификация портов.
- 5) Специальные регистры управления портами.
- 6) Дополнительные регистры управления портами.
- 7) Схема для реализации проверки состояния линий портов.
- 8) Программирование портов ввода/вывода.
- 9) Программная реализация временной задержки на языке Си и Ассемблер.

## Лабораторная работа №5

### *Аппаратные средства обеспечения надежной работы МК*

Цель работы – ознакомление с состояниями сброса и прерывания микроконтроллеров в процессе выполнения программы управления.

#### Общие сведения

В процессе исполнения прикладной программы МК реализует монотонную многократно повторяющуюся последовательность действий:

- Выборку кода команды из памяти программ в регистр команды центрального процессора;
- Дешифрацию кода команды;
- Выборку из памяти следующих байтов команды;
- Исполнение команды;
- Сохранение в памяти результатов исполнения команды.

Если исполняется линейная последовательность команд, то содержимое счетчика РС центрального процессора постоянно увеличивается на 1, обеспечивая выборку из памяти следующих команд прикладной программы. Линейная последовательность исполняемых команд может быть изменена под управлением самой программы, например инструкциями «jmp» или «branch». При этом в счетчик команд под управлением программы будет записано новое число, и начнется исполнение следующего линейного фрагмента программы из другого сегмента памяти программ. Несмотря на явные различия механизмов формирования следующего за исполнением текущей операции значения счетчика команд РС, в обоих рассмотренных случаях это следующее значение РС определяется ходом вычислительного процесса и предсказывается программистом в ходе написания прикладной программы.

В противоположность только что рассмотренному полностью предсказуемому потоку событий, который формируется самой микропроцессорной системой, существует еще поток внешних событий, очередность и моменты возникновения которых не синхронизированы с исполнением центральным процессором тех или иных команд прикладной программы. Однако МК должен реагировать на эти события, для чего необходимо изменить последовательность исполнения операторов программы в произвольный, непредсказуемый с точки зрения устройства управления центральным процессором момент времени. Такое изменение реализуется принудительной записью нового значения в счетчик команд РС под управлением специальных аппаратных средств микроконтроллера, которые реагируют на внешние события. Включение в работу механизма принудительного изменения текущего значения счетчика команд нельзя считать аварийным состоянием микропроцессорной системы. Это лишь специальное состояние, которое позволяет организовать эффективное распределение ресурса одного центрального процессора для обслуживания нескольких устройств, генерирующих в реальном времени несвязанные между собой внешние события. По способу обработки микроконтроллером исключения подразделяются на прерывания и сброс. В русскоязычной литературе термин «исключение» обычно не используется, и говорят просто о состоянии прерывания или о состоянии сброса микроконтроллера (примечание переводчика).

### Состояния сброса и прерывания в микроконтроллерах семейства Motorola

МК семейства 68HC12/HCS12 обладают мощной системой обработки исключений. По способу реакции микроконтроллера на возмущающие события исключения делятся на прерывание и на сброс.

- МК реализует прерывание или МК находится в состоянии прерывания;
- МК находится в состоянии сброса или в состоянии начального запуска. Прерывания в свою очередь делятся на маскируемые и немаскируемые.

### Состояние сброса микроконтроллера

Микроконтроллер семейства 68HC12/HCS12 переходит в состояние сброса по внешнему сигналу или при наступлении определенных внутренних событий. В состоянии сброса программный счетчик и часть битов регистра состояния центрального процессора, а также определенные в техническом описании регистры специальных функций периферийных модулей устанавливаются в начальное состояние. Это состояние однозначно определяет аппаратную конфигурацию микроконтроллера, с которого он начнет работу после включения питания. Поэтому второе название состояния сброса – состояние начального запуска. Состояние сброса МК использует также для восстановления работоспособного состояния после обнаружения внутренней аварийной ситуации.

Различают четыре источника событий, которые переводят МК в состояние сброса:

1) Внешний сброс (External reset). Все МК семейства 68HC12/HCS12 имеют специальный вывод корпуса RESET для подачи сигнала внешнего сброса. Активный уровень сигнала – логический 0. Пока на входе RESET удерживается низкий уровень сигнала, МК будет находиться в состоянии сброса. После перевода линии RESET в состояние логической 1 МК перейдет в активный режим работы по истечении задержки, которая составляет 4096 периодов системной магистрали МК.

2) Внутренний сброс по нарастанию напряжения питания (Power on reset – POR). Нарастание напряжения на входе VDD микроконтроллера вызывает состояние сброса. Таким образом реализуется начальный запуск МК с однозначно определенной аппаратной конфигурацией и с известным начальным адресом запускаемой на исполнение программы.

3) Внутренний сброс по сторожевому таймеру (Computer Operating Properly reset – COP). Логика работы сторожевого таймера позволяет микроконтроллеру выявлять перемежающиеся ошибки в исполнении прикладной программы, которые могут возникнуть в результате электромагнитных помех или при колебаниях напряжения питания микропроцессорной системы. В процессе отладки работа сторожевого таймера запрещена. Работа модуля сторожевого таймера разрешается в конечном варианте прикладной программы, который используется при работе МК в системе. Сторожевой таймер – это счетчик, коэффициент счета которого настраивается пользователем при инициализации системы. Счетчик начинает счет внутренних тактовых импульсов в момент начала исполнения программы. Если счетчик переполнится, то МК перейдет в состояние сброса. Правильно исполняемая прикладная программа, в которой очередность исполнения операторов совпадает с предусмотренной программистом очередностью, должна постоянно сбрасывать сторожевой таймер. Тогда внутреннего сброса от него случаться не будет. Для сброса сторожевого таймера в МК семейства 68HC12/HCS12 необходимо в регистр COPRST записать сначала код \$55, а затем код \$AA. При создании конечного кода прикладной программы разработчик должен разместить операции записи приведенной последовательности кодов так, чтобы исполнение программы по любому

возможному пути обеспечивало бы выполнение команд сброса через меньшие интервалы времени, чем период переполнения сторожевого таймера.

4) Внутренний сброс по отклонению частоты тактовых импульсов МК (Clock Monitor reset). МК переводится в состояние сброса, когда модуль встроенного генератора тактирования обнаруживает выход частоты тактирования за заданные пределы или просто останов системы тактирования.

### Состояние прерывания микроконтроллера

#### **Немаскируемые прерывания.**

В соответствие со своим названием немаскируемые прерывания не могут быть отключены пользователем. Однако в предыдущем абзаце было упомянуто, что установка бита X в 1 запрещает немаскируемые прерывания. Значение бита X действительно равно 1 в состоянии сброса МК. Однако далее он может быть установлен в 0 под управлением программы инициализации, разрешая тем самым немаскируемые прерывания. Далее этот бит не может быть изменен под управлением программы, и в этом его отличие от бита глобальной маски прерываний I.

Три типа немаскируемых прерываний реализуются в МК 68HC12/HCS12:

1) Прерывание по внешнему запросу XIRQ. Все МК 68HC12/HCS12 имеют вывод внешнего немаскируемого прерывания XIRQ. Активный уровень сигнала для генерации запроса на прерывание – логический 0.

2) Прерывание по несуществующему коду команды. Каждая инструкция языка ассемблер МК имеет собственный код. В МК 68HC12/HCS12 коды операций могут быть однобайтовыми и двухбайтовыми. Но не все теоретически возможные коды использованы для кодирования реальных команд процессорного ядра CPU12. Если на этапе выборки кода команды из памяти произошло считывание несуществующего кода команды, то генерируется запрос на немаскируемое прерывание.

3) Программное прерывание – инструкция SWI. Система команд МК 68HC12/HCS12 имеет инструкцию программного прерывания, которая позволяет перейти к исполнению подпрограммы прерывания из прикладной программы.

#### **Маскируемые прерывания.**

1) Прерывание по внешнему запросу IRQ. Все МК 68HC12/HCS12 имеют вывод внешнего маскируемого прерывания IRQ. Активный уровень сигнала для генерации запроса на прерывание – логический 0. В некоторых приложениях требуется принять запросы от нескольких внешних источников сигналов. Для таких случаев следует использовать дополнительный логический элемент, который объединяет запросы от всех источников по логике ИЛИ. Если запрос на вход IRQ поступил, и МК перешел к выполнению подпрограммы прерывания, то в этой подпрограмме следует опросить линии порта для того, чтобы установить, какой из источников вызвал прерывание. Обработка нескольких объединенных по ИЛИ запросов с программным поиском установившегося запрос источника называется поллингом.

2) Прерывание по таймеру меток реального времени RTI. Таймер меток реального времени генерирует последовательность равноотстоящих во времени запросов на прерывание. Период повторения запросов настраивается программистом. Эти прерывания могут быть использованы для регулярного выполнения микроконтроллером некоторой задачи. Например, для измерения напряжения аккумуляторной батареи каждые три мин, чтобы сигнализировать о необходимости ее замены. Мы рассмотрим особенности прерываний RTI в главе 7 на примере управления скоростью вращения электрическим двигателем.

3) Прерывание по событию канала захвата/сравнения (IC/OC) таймера. Восемь одинаковых блоков в составе модуля таймера, которые именуют «каналами»,

предназначены для контроля за уровнем сигнала на входе канала или для изменения в строго определенный момент времени логического уровня на выходе канала. Заданное программистом изменение входного или выходного сигнала канала рассматривается как событие, которое генерирует запрос на прерывание. Например, если канал настроен на слежение за перепадом входного сигнала из 1 в 0, то когда такое изменение произойдет, будет выставлен запрос на прерывание.

4) Прерывание по переполнению таймера. Основным блоком модуля таймера является 16\_разрядный счетчик временной базы. Этот счетчик невозможно остановить. Также невозможно изменить его коэффициент счета, который составляет  $2^{16} = 65536$ . Поэтому регулярно счетчик временной базы изменяет свой код с \$FFFF на \$0000. Такое изменение кода называют переполнением счетчика. В момент переполнения по желанию программиста может генерироваться запрос на прерывание, в то время как счетчик продолжает считать дальше. Такие прерывания особенно удобны при необходимости измерения очень больших временных интервалов. Для этого в микроконтроллере производят подсчет, сколько переполнений счетчика произошло за этот временной интервал, и, зная период счета счетчика, определяют длительность исследуемого временного интервала.

5) Прерывание по переполнению счетчика внешних событий. Когда счетчик внешних событий переполняется, то может генерироваться запрос на прерывание точно так же, как и для счетчика временной базы.

6) Прерывание по событию на входе счетчика внешних событий. Этот запрос на прерывание формируется, если сигнал на входе счетчика внешних событий изменил свое значение. Характер изменения, т.е. перепад из 0 в 1, или из 1 в 0, или любое изменение логического уровня, определяется программистом.

7) Прерывание от модулей контроллеров последовательного ввода/вывода SCI и SPI. Каждый модуль последовательного ввода/вывода формирует целую группу запросов на прерывание: при завершении передачи слова, при приеме слова, при обнаружении различного рода нарушений в протоколе передачи информации.

8) Прерывание от модуля АЦП. Модуль аналого-цифрового преобразователя формирует запрос на прерывание, когда процесс оцифровки очередного сигнала завершен, и двоичный код сигнала может быть считан в память МК.

9) Прерывание при выходе МК из энергосберегающих режимов. Это прерывание позволяет вывести МК из состояния STOP или WAIT, в котором он находился с целью снижения потребляемой энергии. Такие прерывания очень полезны при объединении нескольких МК в информационную сеть. Мы рассмотрим этот тип прерывания более подробно в следующем параграфе.

### Задание

Перед началом выполнения практической части необходимо ознакомиться с лабораторной установкой на базе платы Freescale и программным обеспечением CodeWarrior.

В данной практической работе мы рассмотрим основные особенности формирования исходного текста программы с прерываниями на Си.

При программировании на Си Вы должны обязательно реализовать следующие этапы записи исходного текста:

1. При написании программы обработки прерывания на Си, имя подпрограммы обработки прерывания должно быть объявлено с использованием специальной директивы препроцессора.

```
#pragma interrupt_handler <name>
```

В поле <name> следует записать имя подпрограммы прерывания, которое Вы будете далее использовать в тексте программы. Приведенная запись информирует компилятор о том, что функция с названным именем является подпрограммой прерывания.

2. Далее по тексту подпрограмма прерывания оформляется как обычная функция. Компилятор в процессе перевода исходного текста этой функции на Си в инструкции ассемблера автоматически подставит в конце подпрограммы команду возврата из прерывания RTI, потому что эта функция была объявлена подпрограммой прерывания (директива #pragma на этапе 1).

3. Для правильного функционирования МК в процессе прерывания необходимо инициализировать указатель стека. Его значение должно быть равно старшему адресу области оперативной памяти МК, увеличенному на единицу. Поскольку диапазоны памяти пользователя (как постоянной, так и оперативной) являются необходимыми установками в конфигурации компилятора, то функция инициализации указателя стека выполняется компилятором автоматически. Поэтому программист не должен записывать какой либо текст в программе для инициализации указателя стека. Зато следует проверить карту памяти в установках компилятора, которая обязательно должна совпадать с реальной проектируемой системой.

4. Подсистема прерывания будет функционировать корректно, если для нее сформирована таблица векторов прерывания. Мы уже обсуждали, что таблица векторов прерывания в МК B32 находится в области Flash памяти, которая защищена от перезаписи информации. Для того, чтобы пользователь имел возможность записать собственную таблицу векторов сброса и прерывания, в эту нестираемую область памяти записаны фиксированные вектора, которые передают управление по известным адресам в области перезаписываемой EEPROM памяти. По этим адресам программист должен вписать команду безусловного перехода JMP с адресом соответствующей подпрограммы обработки прерывания.

5. Каждый маскируемый источник запроса на прерывание должен быть разрешен установкой соответствующего бита в регистре управления периферийного модуля. Мы рассмотрим, как это записать на Си в последующих примерах.

6. После установки всех индивидуальных битов на разрешение прерывания, необходимо сбросить глобальную маску прерывания I. На ассемблере для этого используют команду CLI. При программировании на Си мы также воспользуемся этой командой, посредством следующих макросов:

```
#define CLI ( ) asm(«cli\n»); //разрешить маскируемые прерывания
#define SEI ( ) asm(«sei\n»); //запретить маскируемые прерывания
```

Далее по тексту программы, если необходимо разрешить прерывания, то следует ввести CLI ( ).

```
//объявление функции в модуле
void toggle_isr(void);
//директива #pragma для указания, что функция является подпрограммой
//обслуживания прерывания
#pragma interrupt_handler toggle_isr
//инициализация соответствующего вектора в таблице векторов прерываний
#pragma abs_address: 0xF7EA //B32 RAM_based vector address
void (*Timer_Channel_2_interrupt_vector[]) ()={toggle_isr};
#pragma end_abs_address
```

### Контрольные вопросы

- 1) Последовательность действий МК при выполнении исполняемой программы.
- 2) Состояния сброса МК.
- 3) Состояния прерывания МК.
- 4) Маскируемые и немаскируемые прерывания.
- 5) Вектора исключений.
- 6) Система приоритета для исключений.
- 7) Регистры подсистемы прерывания.
- 8) Процесс перехода к системе прерывания.
- 9) Программная реализация систем сброса.
- 10) Программная реализация систем прерывания.

### Список литературы

#### Основная литература:

1. Таненбаум Эндрю С. Архитектура компьютера. 5-е изд. (+CD) . – СПб.: Питер, 2007. – 848 с.
2. Бройдо В.Л., Ильина О.П. Архитектура ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Питер, 2009. – 720 с.
3. Новиков Ю.В., Скоробогатов П.К. Основы микропроцессорной техники – Интернет-университет информационных технологий, ИНТУИТ.ру, 2006 г. 440 стр. Свободный доступ в интернет: <http://www.intuit.ru/departement/hardware/mpbasics/>

#### Дополнительная литература:

4. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения: Справочник. изд. 2-е, перераб. и доп. — М.: Радио и связь, 1990 — 512 с.: ил.
5. Новиков Ю.В. Функциональные модули контрольно-измерительных систем на базе микроЭВМ Микропроцессорные средства и системы. — 1990.—N 3.— с. 75—77.
6. Новиков Ю.В., Калашников О.А., Гуляев С.Э., Под общ. ред. Ю.В. Новикова. Разработка устройств сопряжения для персональных компьютеров типа IBM PC. Практик. пособие. — М.: ЭКОМ, 1997. — 224 с.: ил.
7. Предко М. Руководство по микроконтроллерам. Том 1. М.: Постмаркет, 2001. — 416 с.: ил.
8. Баррет С.Ф., Пак Д.Дж. Встраиваемые системы. Проектирование приложений на МК семейства 68HC12/HCS12 с применением языка С. — М.: ДМК - пресс, 2007. — 640 с.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕ-  
ЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**для самостоятельной работы студентов по  
дисциплине «Программирование встраиваемых  
микропроцессорных систем»**

для студентов направления подготовки  
09.03.01 Информатика и вычислительная техника,  
Направленность (профиль) «Программное обеспечение  
средств вычислительной техники и автоматизированных систем»

Ставрополь, 2026

Методические указания по организации самостоятельной работы студентов составлено в соответствии с требованиями программы дисциплины «Программирование встраиваемых микропроцессорных систем» для бакалавров направления подготовки 09.03.01 Информатика и вычислительная техника. Учебно-методическое пособие посвящено планируемой учебной работе студентов, выполняемой во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия, при частичном, непосредственном участии преподавателя.

## СОДЕРЖАНИЕ

1. Самостоятельная работа как важнейшая форма учебного процесса
2. Цели и основные задачи СРС
3. Виды самостоятельной работы
4. Организация СРС
5. Общие рекомендации по организации самостоятельной работы
6. Самостоятельная работа студента - необходимое звено становления исследователя
7. Методические рекомендации для студентов по отдельным формам самостоятельной работы
8. Учебно-методическое и информационное обеспечение дисциплины

## 1. Самостоятельная работа как важнейшая форма учебного процесса

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Необходимо ознакомиться с технологической картой самостоятельной работы обучающегося.

| № | Раздел (тема)<br>дисциплины и краткое<br>содержание                                                                                                                                                                                                                 | Формируе-<br>мые<br>компетенции<br>, индикаторы | Самостояте-<br>льная<br>работа,<br>часов | Формы текущего<br>контроля<br>успеваемости        |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|------------------------------------------|---------------------------------------------------|
| 1 | Тема 1. Особенности программирования микропроцессорных систем.<br>Уровни представления микропроцессорной системы.<br>Программная модель микропроцессора.<br>Архитектура микроконтроллеров.<br>Типовой состав микроконтроллеров.<br>Низкоуровневое программирование. | ИД-1 ПК-4<br>ИД-2 ПК-4<br>ИД-3 ПК-4             | 4                                        | собеседование<br>защита<br>лабораторной<br>работы |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                    |   |                                                                      |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|---|----------------------------------------------------------------------|
| 2 | <p>Тема 2.</p> <p>Инструментальные средства разработки программного обеспечения встраиваемых микропроцессорных систем.</p> <p>Назначение и классификация инструментальных средств разработки программного обеспечения (IDE).</p> <p>Состав типовой IDE.</p> <p>Сравнительный анализ Arduino IDE, Microchip Studio, Keil uVision, STM32 Cube IDE.</p> <p>Отечественные инструментальные средства разработки программного обеспечения.</p> <p>Создание блок-схемы алгоритмов функционирования разрабатываемых программных продуктов и программной документации согласно ЕСПД с использованием IDE</p> | <p>ИД-1 ПК-4</p> <p>ИД-2 ПК-4</p> <p>ИД-3 ПК-4</p> | 6 | <p>собеседование</p> <p>защита</p> <p>лабораторной</p> <p>работы</p> |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|---|----------------------------------------------------------------------|

|   |                                                                                                                                                                                                                                                                                                                                                                                    |                                              |   |                                                             |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|---|-------------------------------------------------------------|
| 3 | <p>Тема 3. Организация вычислительного процесса в системах реального времени. Организация вычислительного процесса с использованием диспетчеризации. Организация обработки прерываний. Операционная система реального времени FreeRTOS. Организация вычислительного процесса с использованием OCPB</p>                                                                             | <p>ИД-1 ПК-4<br/>ИД-2 ПК-4<br/>ИД-3 ПК-4</p> | 6 | <p>собеседование<br/>защита<br/>лабораторной<br/>работы</p> |
| 4 | <p>Тема 4. Программирование драйверов микропроцессорных устройств. Универсальный синхронно-асинхронный приемопередатчик (USART). Реализация последовательного информационного обмена с использованием USART. Программирование драйверов интерфейсов RS485, RS232 на основе протокола ModBus. Разработка драйверов для обмена данными с использованием проприетарных протоколов</p> | <p>ИД-1 ПК-4<br/>ИД-2 ПК-4<br/>ИД-3 ПК-4</p> | 6 | <p>собеседование<br/>защита<br/>лабораторной<br/>работы</p> |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                    |   |                                                               |
|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|---|---------------------------------------------------------------|
| 5 | <p>Тема 5.</p> <p>Программирование задач, связанных с генерацией и обработкой временных параметров. Таймеры и процессоры событий и их реализация в структуре микроконтроллеров. Измерение временных интервалов. Часы реального времени. Встроенные функции измерения времени. Генерация последовательностей импульсов заданными длительностями и периодом следования. Подсчет событий, связанных с внешними сигналами и результатами работы программ.</p> | <p>ИД-1 ПК-4</p> <p>ИД-2 ПК-4</p> <p>ИД-3 ПК-4</p> | 4 | <p>собеседование</p> <p>защита</p> <p>лабораторной работы</p> |
| 6 | <p>Программирование задач, связанных с обработкой аналоговых сигналов. Устройства, алгоритмы, точность аналогоцифровых преобразований. Сравнение аналоговых величин. Цифро-аналоговые преобразователи. Формирование аналоговых сигналов на основе ШИМ</p>                                                                                                                                                                                                 | <p>ИД-1 ПК-4</p> <p>ИД-2 ПК-4</p> <p>ИД-3 ПК-4</p> | 6 | <p>собеседование</p> <p>защита</p> <p>лабораторной работы</p> |

|   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                    |   |                                                                      |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|---|----------------------------------------------------------------------|
| 7 | <p>Тема 7. Комплексная отладка программного обеспечения микропроцессорных систем.</p> <p>Отладочные платы и инструментальные средства отладки.</p> <p>Эмуляторы.</p> <p>Интерфейс JTAG и его использование для верификации программного обеспечения микропроцессорного устройства.</p> <p>Внутрисхемное программирование с использованием интерфейса JTAG.</p> <p>Программирование микроконтроллеров по интерфейсу SPI.</p> <p>Оценка вычислительной сложности разрабатываемых программных продуктов</p> | <p>ИД-1 ПК-4</p> <p>ИД-2 ПК-4</p> <p>ИД-3 ПК-4</p> | 6 | <p>собеседование</p> <p>защита</p> <p>лабораторной</p> <p>работы</p> |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|---|----------------------------------------------------------------------|

|   |                                                                                                                                                                                                                                                                                                                                                                                                             |                                        |    |                                                                   |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|----|-------------------------------------------------------------------|
| 8 | Тема<br>Программируемые<br>логические<br>контроллеры<br>Назначение<br>программируемых<br>логических<br>контроллеров.<br>Программируемые<br>логические<br>контроллеры<br>компания Овен.<br>Общая характеристика<br>языков<br>программирования<br>ПЛК. Графические<br>языки<br>программирования<br>ПЛК. Текстовые<br>языки<br>программирования<br>ПЛК.<br>Интегрированные<br>среды<br>программирования<br>ПЛК | 8. ИД-1 ПК-4<br>ИД-2 ПК-4<br>ИД-3 ПК-4 | 6  | защита<br>лабораторной<br>работы,<br>компьютерное<br>тестирование |
|   | ИТОГО                                                                                                                                                                                                                                                                                                                                                                                                       |                                        | 44 |                                                                   |

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.

Концепцией модернизации российского образования определены основные задачи профессионального образования - "подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности".

Решение этих задач невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за развитие навыков самостоятельной работы, за стимулирование

профессионального роста студентов, воспитание творческой активности и инициативы.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в лабораторных занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Формы самостоятельной работы студентов разнообразны. По данной дисциплине «Программирование микропроцессорных систем» предусмотрены следующие:

#### **Использование материала учебно-методического комплекса дисциплины**

На первом этапе необходимо ознакомиться с обязательным минимумом содержания основной образовательной программы по изучаемой дисциплине. Далее необходимо изучить рабочую программу дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций с лабораторными занятиями, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, представленные в учебной программе по дисциплине «Программирование микропроцессорных систем».

#### **Работа с литературой**

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации, которые представлены в учебной программе.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

## **2. Цели и основные задачи СРС**

Ведущая цель организации и осуществления СРС должна совпадать с целью обучения студента – подготовкой бакалавра с высшим образованием. При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

### **3. Виды самостоятельной работы**

В образовательном процессе высшего профессионального образовательного учреждения выделяется два вида самостоятельной работы – аудиторная, под руководством преподавателя, и внеаудиторная. Тесная взаимосвязь этих видов работ предусматривает дифференциацию и эффективность результатов ее выполнения и зависит от организации, содержания, логики учебного процесса (межпредметных связей, перспективных знаний и др.):

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Основными видами самостоятельной работы студентов без участия преподавателя по данной дисциплине являются:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- подготовка к лабораторным работам, их оформление.

Основными видами самостоятельной работы студентов с участием преподавателей являются:

- текущие консультации;
- прием и защита лабораторных работ (во время проведения л/р).
- выполнение курсовой работы.

#### **4. Организация СРС**

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Программирование цифровых сигнальных процессоров», объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

#### **Деятельность студентов по формированию и развитию навыков учебной самостоятельной работы**

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

*Выполняя самостоятельную работу под контролем преподавателя студент должен:*

- освоить минимум содержания, выносимый на самостоятельную работу студентов и предложенный преподавателем в соответствии с образовательными стандартами высшего образования;
- планировать самостоятельную работу в соответствии с графиком самостоятельной работы, предложенным преподавателем;
- самостоятельную работу студент должен осуществлять в организационных формах, предусмотренных учебным планом и рабочей программой преподавателя;

– выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

*студент может:*

сверх предложенного преподавателем (при обосновании и согласовании с ним) и минимума обязательного содержания, определяемого программой по данной дисциплине:

–самостоятельно определять уровень (глубину) проработки содержания материала;

–предлагать темы и вопросы для самостоятельной проработки;

– в рамках общего графика выполнения самостоятельной работы предлагать обоснованный индивидуальный график выполнения и отчетности по результатам самостоятельной работы;

– предлагать свои варианты организационных форм самостоятельной работы;

–использовать для самостоятельной работы методические пособия, учебные пособия, разработки сверх предложенного преподавателем перечня;

–использовать не только контроль, но и самоконтроль результатов самостоятельной работы в соответствии с методами самоконтроля, предложенными преподавателем или выбранными самостоятельно.

Самостоятельная работа студентов должна оказывать важное влияние на формирование личности будущего бакалавра, она планируется студентом самостоятельно. Каждый студент самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием по каждой дисциплине. Он выполняет внеаудиторную работу по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

## **5. Общие рекомендации по организации самостоятельной работы**

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на лабораторных занятиях. Но для успешной учебной деятельности, ее интенсификации, необходимо учитывать следующие субъективные факторы:

1. Знание школьного программного материала, наличие прочной системы знаний, необходимой для усвоения основных вузовских курсов. Это особенно важно для математических дисциплин. Необходимо отличать пробелы в знаниях, затрудняющие усвоение нового материала, от малых способностей.

2. Затратив силы на преодоление этих пробелов, студент обеспечит себе нормальную успеваемость и поверит в свои способности.

3. Наличие умений, навыков умственного труда:

а) умение конспектировать на лекции и при работе с книгой;

б) владение логическими операциями: сравнение, анализ, синтез, обобщение, определение понятий, правила систематизации и классификации.

4. Специфика познавательных психических процессов: внимание, память, речь, наблюдательность, интеллект и мышление. Слабое развитие каждого из них становится серьезным препятствием в учебе.

5. Хорошая работоспособность, которая обеспечивается нормальным физическим состоянием. Ведь серьезное учение - это большой многосторонний и разнообразный труд. Результат обучения оценивается не количеством сообщаемой информации, а качеством ее усвоения, умением ее использовать и развитием у себя способности к дальнейшему самостоятельному образованию.

6. Соответствие избранной деятельности, профессии индивидуальным способностям. Необходимо выработать у себя умение саморегулировать свое эмоциональное состояние и устранять обстоятельства, нарушающие деловой настрой, мешающие намеченной работе.

7. Овладение оптимальным стилем работы, обеспечивающим успех в деятельности. Чередование труда и пауз в работе, периоды отдыха, индивидуально обоснованная норма продолжительности сна, предпочтение вечерних или утренних занятий, стрессоустойчивость на экзаменах и особенности подготовки к ним,

8. Уровень требований к себе, определяемый сложившейся самооценкой.

Адекватная оценка знаний, достоинств, недостатков - важная составляющая самоорганизации человека, без нее невозможна успешная работа по управлению своим поведением, деятельностью.

Одна из основных особенностей обучения в высшей школе заключается в том, что постоянный внешний контроль заменяется самоконтролем, активная роль в обучении принадлежит уже не столько преподавателю, сколько студенту. Зная основные методы научной организации умственного труда, можно при наименьших затратах времени, средств и трудовых усилий достичь наилучших результатов. Эффективность усвоения поступающей информации зависит от работоспособности человека в тот или иной момент его деятельности.

*Работоспособность* - способность человека к труду с высокой степенью напряженности в течение определенного времени. Различают внутренние и внешние факторы работоспособности.

К внутренним факторам работоспособности относятся интеллектуальные особенности, воля, состояние здоровья.

К внешним:

- организация рабочего места, режим труда и отдыха;
- уровень организации труда - умение получить справку и пользоваться информацией;

- величина умственной нагрузки.

Выдающийся русский физиолог Н. Е. Введенский выделил следующие условия продуктивности умственной деятельности:

- во всякий труд нужно входить постепенно;
- мерность и ритм работы. Разным людям присущ более или менее разный темп работы;
- привычная последовательность и систематичность деятельности;
- правильное чередование труда и отдыха.

Отдых не предполагает обязательного полного бездействия со стороны человека, он может быть достигнут простой переменой дела. В течение дня работоспособность изменяется. Наиболее плодотворным является *утреннее время* (с 8 до 14 часов), причем максимальная работоспособность приходится на период с 10 до 13 часов, затем *послеобеденное* - (с 16 до 19 часов) и *вечернее* (с 20 до 24 часов). Очень трудный для понимания материал лучше изучать в начале каждого отрезка времени (лучше всего утреннего) после хорошего отдыха. Через 1-1,5 часа нужны перерывы по 10 - 15 мин, через 3 - 4 часа работы отдых должен быть продолжительным - около часа.

Составной частью научной организации умственного труда является овладение техникой умственного труда.

Время, которым располагает студент для выполнения учебного плана, складывается из двух составляющих: одна из них - это аудиторная работа в вузе по расписанию занятий, другая - внеаудиторная самостоятельная работа. Задания и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль за самостоятельной работой, а также оказывает помощь студентам по правильной организации работы.

Самостоятельные занятия потребуют интенсивного умственного труда, который необходимо не только правильно организовать, но и стимулировать. При этом очень важно уметь поддерживать устойчивое внимание к изучаемому материалу. Выработка внимания требует значительных волевых усилий. Именно поэтому, если студент замечает, что он часто отвлекается во время самостоятельных занятий, ему надо заставить себя сосредоточиться. Подобную процедуру необходимо проделывать постоянно, так как это является тренировкой внимания. Устойчивое внимание появляется тогда, когда человек относится к делу с интересом.

Следует правильно организовать свои занятия по времени: 50 минут - работа, 5-10 минут - перерыв; после 3 часов работы перерыв - 20-25 минут. Иначе нарастающее утомление повлечет неустойчивость внимания. Очень существен-

ным фактором, влияющим на повышение умственной работоспособности, являются систематические занятия физической культурой. Организация активного отдыха предусматривает чередование умственной и физической деятельности, что полностью восстанавливает работоспособность человека.

## **6. Самостоятельная работа студента - необходимое звено становления исследователя**

Прогресс науки и техники, информационных технологий приводит к значительному увеличению научной информации, что предъявляет более высокие требования не только к моральным, нравственным свойствам человека, но и в особенности, постоянно возрастающие требования в области образования – обновление, модернизация общих и профессиональных знаний, умений специалиста.

Всякое образование должно выступать как динамический процесс, присущий человеку и продолжающийся всю его жизнь. Овладение научной мыслью и языком науки является необходимой составляющей в самоорганизации будущего специалиста исследователя. Под этим понимается не столько накопление знаний, сколько овладение научно обоснованными способами их приобретения. В этом, вообще говоря, состоит основная задача вуза.

Специфика вузовского учебного процесса, в организации которого самостоятельной работе студента отводятся все больше места, состоит в том, что он является как будто бы последним и самым адекватным звеном для реализации этой задачи. Ибо во время учебы в вузе происходит выработка стиля, навыков учебной (познавательной) деятельности, рациональный характер которых будет способствовать постоянному обновлению знаний высококвалифицированного выпускника вуза.

Однако до этого пути существуют определенные трудности, в частности, переход студента от синтетического процесса обучения в средней школе, к аналитическому в высшей. Это связано как с новым содержанием обучения (расширение общего образования и углубление профессиональной подготовки), так и с новыми, неизвестными до сих пор формами: обучения (лекции, семинары, лабораторные занятия и т.д.). Студент получает не только знания, предусмотренные программой и учебными пособиями, но он также должен познакомиться со способами приобретения знаний так, чтобы суметь оценить, что мы знаем, откуда мы это знаем и как этого знания мы достигли. Ко всему этому приходят через собственную самостоятельную работу.

Это и потому, что самостоятельно приобретенные знания являются более оперативными, они становятся личной собственностью, а также мотивом поведения, развивают интеллектуальные черты, внимание, наблюдательность, критичность, умение оценивать. Роль преподавателя в основном заключается в руководстве накопления знаний (по отношению к первокурсникам), а в последующие годы учебы, на старших курсах, в совместном установлении проблем и заботе о самостоятельных поисках студента, а также контролировании за их деятельностью. Отметим, что нельзя ограничиваться только приобретением знаний предусмотренных программой изучаемой дисциплины, надо постоянно углублять полученные знания, сосредотачивая их на какой-нибудь узкой определенной области, соответствующей интересам студента. Углубленное изучение всех предметов, предусмотренных программой, на практике является возможным, и хорошая организация работы позволяет экономить время, что создает условия для глубокого, систематического, заинтересованного изучения самостоятельно выбранной студентом темы.

Конечно, все советы, примеры, рекомендации в этой области, даваемые преподавателем, или определенными публикациями, или другими источниками, не гарантируют никакого успеха без проявления собственной активности в этом деле, т. е. они не дают готовых рецептов, а должны способствовать анализу собственной работы, ее целей, организации в соответствии с индивидуальными особенностями. Учитывая личные возможности, существующие условия жизни и работы, навыки, на основе этих рекомендаций, возможно, выработать индивидуально обоснованную совокупность методов, способов, найти свой стиль или усовершенствовать его, чтобы изучив определенный материал, иметь время оценить его значимость, пригодность и возможности его применения, чтобы, в конечном счете, обеспечить успешность своей учебы с будущей профессиональной деятельности

## **7. Методические рекомендации для студентов по отдельным формам самостоятельной работы**

Система вузовского обучения подразумевает значительно большую самостоятельность студентов в планировании и организации своей деятельности.

### **Работа с литературой**

При работе с литературой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Различают два вида чтения; первичное и вторичное. *Первичное* - это внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятого слова. Содержание не всегда может быть понятно после первичного чтения.

Задача *вторичного* чтения полное усвоение смысла целого (по счету это чтение может быть и не вторым, а третьим или четвертым).

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того насколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют четыре основные установки в чтении научного текста:

1. информационно-поисковый (задача – найти, выделить искомую информацию)

2. усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

3. аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

4. творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

С наличием различных установок обращения к научному тексту связано существование и нескольких **видов чтения**:

1. библиографическое – просматривание карточек каталога, рекомендательных списков, сводных списков журналов и статей за год и т.п.;

2. просмотрное – используется для поиска материалов, содержащих нужную информацию, обычно к нему прибегают сразу после работы со списками литературы и каталогами, в результате такого просмотра читатель устанавливает, какие из источников будут использованы в дальнейшей работе;

3. ознакомительное – подразумевает сплошное, достаточно подробное прочтение отобранных статей, глав, отдельных страниц, цель – познакомиться с характером информации, узнать, какие вопросы вынесены автором на рассмотрение, провести сортировку материала;

4. изучающее – предполагает доскональное освоение материала; в ходе такого чтения проявляется доверие читателя к автору, готовность принять изложенную информацию, реализуется установка на предельно полное понимание материала;

5. аналитико-критическое и творческое чтение – два вида чтения близкие между собой тем, что участвуют в решении исследовательских задач. Первый из них предполагает направленный критический анализ, как самой информации, так и способов ее получения и подачи автором; второе – поиск тех суждений, фактов, по которым или в связи с которыми, читатель считает нужным высказать собственные мысли.

Из всех рассмотренных видов чтения основным для студентов является изучающее – именно оно позволяет в работе с учебной литературой накапливать знания в различных областях. Вот почему именно этот вид чтения в рамках учебной деятельности должен быть освоен в первую очередь. Кроме того, при овладении данным видом чтения формируются основные приемы, повышающие эффективность работы с научным текстом.

### **Основные виды систематизированной записи прочитанного:**

1. Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;
2. Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;
3. Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;
4. Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;
5. Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

### **Методические рекомендации по составлению конспекта:**

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.
2. Выделите главное, составьте план.
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

### **Лабораторные занятия**

Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на лабораторных занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью выполнения заданий лабораторной работы, поставленных задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном выполнении заданий нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что выполнение каждого учебного задания должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Выполнение заданий данного типа нужно продолжать до приобретения твердых навыков в их решении.

### **Самопроверка**

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих заданий на лабораторных занятиях и самостоятельно студенту рекомендуется, используя лист опорных материалов, воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение за-

дачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

### **Консультации**

Если в процессе самостоятельной работы над изучением теоретического материала или при выполнении заданий у студента возникают вопросы, разрешить которые самостоятельно не удастся, необходимо обратиться к преподавателю для получения у него разъяснений или указаний. В своих вопросах студент должен четко выразить, в чем он испытывает затруднения, характер этого затруднения. За консультацией следует обращаться и в случае, если возникнут сомнения в правильности ответов на вопросы самопроверки.

### **Подготовка к зачету по дисциплине «Программирование микропроцессорных систем».**

Изучение многих общепрофессиональных и специальных дисциплин завершается зачетом. Подготовка к зачету способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к нему, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания.

Для сдачи зачета студенту необходимо полностью выполнить задания лабораторных работ занятий, подготовить отчет по каждой лабораторной работе и отчет по результатам самостоятельного изучения тем, согласно учебной программе. При наличии задолженностей по текущей аттестации по данной дисциплине студент может получить оценку «не зачтено».

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Текущая аттестация студентов проводится преподавателями, ведущими лабораторные занятия по дисциплине в следующих формах: собеседование при защите лабораторной работы с предоставлением письменного отчета. Допуск к лабораторным работам происходит при наличии у студентов печатного варианта отчета. Защита отчета проходит в форме собеседования.

## **8. Учебно-методическое и информационное обеспечение дисциплины**

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Основная литература | <ol style="list-style-type: none"><li>1. Баженов, А. В. Программирование встраиваемых микропроцессорных систем : учебник / А. В. Баженов, Н. Ю. Братченко, Н. В. Гривенная. – Ставрополь : Изд-во СКФУ, 2022. – 302 с., экземпляров неограниченно.</li><li>2. Новиков, Ю. В.; Основы микропроцессорной техники : учебное пособие / Ю. В. Новиков, П. К. Скоробогатов. - Основы микропроцессорной тех-ники, 2022-07-28. - Электрон. дан. (1 файл). - Москва : Интернет-Университет Информа-ционных Технологий (ИНТУИТ), Ай Пи Ар</li></ol> |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | <p>Медиа, 2020. - 405 с. - электронный. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-4497-0677-5, экземпляров неограничено</p> <p>3.Водовозов, А. М. Микроконтроллеры для систем автоматики [Электронный ресурс] : учебное пособие / А. М. Водовозов. — Электрон. текстовые данные. — М. : Инфра-Инженерия, 2016. — 164 с. — 978-5-9729-0138-8. — Режим доступа: <a href="http://www.iprbookshop.ru/51727.html">http://www.iprbookshop.ru/51727.html</a></p> <p>4.Основы программирования микропроцессорных контроллеров в цифровых системах управления технологическими процессами [Электронный ресурс] : учебное пособие / В. С. Кудряшов, А. В. Иванов, М. В. Алексеев [и др.]. — Электрон. текстовые данные. — Воронеж : Воронежский государственный университет инженерных технологий, 2014. — 144 с. — 978-5-00032-054-9. — Режим доступа: <a href="http://www.iprbookshop.ru/47437.html">http://www.iprbookshop.ru/47437.html</a></p> <p>5.Шегал, А. А. Применение программного комплекса Multisim для проектирования устройств на микроконтроллерах [Электронный ресурс] : лабораторный практикум / А. А. Шегал ; под ред. В. И. Иевлев. — Электрон. текстовые данные. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2014. — 116 с. — 978-5-7996-1117-0. — Режим доступа: <a href="http://www.iprbookshop.ru/65968.html">http://www.iprbookshop.ru/65968.html</a></p> |
| Дополнительная литература | <p>1. Алиев, М. Т.; Микропроцессоры и микропроцессорные системы управления. 8-разрядные процессоры семейства AVR : лабораторный практикум / М.Т. Алиев, Т.С. Буканова ; Поволжский государственный технологический университет. - Йошкар-Ола : ПГТУ, 2016. - 64 с. : схем., табл., ил. - <a href="http://biblioclub.ru/">http://biblioclub.ru/</a>. - Библиогр. в кн. - ISBN 978-5-8158-1775-3, экземпляров неограничено</p> <p>2. Безуглов, Д. А. Цифровые устройства и микропроцессоры : учеб. пособие / Д. А. Безуглов, И. В. Калиенко. - Изд. 2-е. - Ростов н/Д : Феникс, 2008. - 468 с. : ил. - (Высшее образование). - Библиогр.: с. 464-465. - ISBN 978-5-222-13917-2</p> <p>3.Хартов, В. Я. Микроконтроллеры AVR : практикум для начинающих : учеб. по-собие / В. Я. Хартов. - М. : МГТУ, 2007. - 240 с. : ил. - Прил.: с. 231-238. - Библиогр.: с. 239. - ISBN 978-5-7038-3051-2</p> <p>4. Мельников, Е. В.; Основы микропроцессорной техники : лабораторный практикум / Е. В. Мельников. - Основы микропроцессорной техни-ки, 2026-09-20. - Электрон. дан. (1 файл). - Самара : Самарский государственный техниче-ский университет, ЭБС АСВ, 2020. - 47 с. - электронный. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397, экземпляров неограничено</p>                                                                                                                           |

|                            |                                                                                                        |
|----------------------------|--------------------------------------------------------------------------------------------------------|
|                            |                                                                                                        |
| Программное<br>обеспечение | Альт Рабочая станция 10<br>Альт Рабочая станция К<br>Альт «Сервер»<br>Пакет офисных программ - Р7-Офис |