



**УТВЕРЖДЕНО:**

**Ученым советом Института сервисных  
технологий**

**Протокол №24 от «16» января 2025г.**

**РАБОЧАЯ ПРОГРАММА  
ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

**ПМ. Программирование встраиваемых систем с использованием интегрированных сред  
разработки**

**основной профессиональной образовательной программы среднего профессионального  
образования – программы подготовки специалистов среднего звена**

**по специальности: 11.02.17 Разработка электронных устройств и систем**

**Квалификация: техник**

**год начала подготовки: 2025**

**Разработчики:**

должность	ученая степень и звание, ФИО
<i>преподаватель</i>	<i>Голубцов А.С.</i>

**Рабочая программа согласована и одобрена руководителем ППСЗ:**

должность	ученая степень и звание, ФИО
<i>преподаватель</i>	<i>Голубцов А.С.</i>

**Рабочая программа согласована и одобрена представителем работодателей:**

должность	должность, ФИО
<i>главный технолог ООО «Московский завод «ФИЗПРИБОР»</i>	<i>Онищенко Н.Н.</i>

	ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТУРИЗМА И СЕРВИСА»	СМК РГУТИС
		<i>Лист 2</i>

## *СОДЕРЖАНИЕ*

- 1    Общая характеристика рабочей программы профессионального модуля**
- 2    Структура и содержание профессионального модуля**
- 3    Методические указания по проведению практических занятий, занятий в форме практической подготовки и самостоятельной работе**
- 4    Фонд оценочных средств профессионального модуля**
- 5    Фонд оценочных средств для аттестации по модулю**
- 6    Условия реализации профессионального модуля**
- 7    Информационное обеспечение реализации программы профессионального модуля**

	ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТУРИЗМА И СЕРВИСА»	СМК РГУИС
		Лист 3

## 1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

### «ПМ.04 Программирование встраиваемых систем с использованием интегрированных сред разработки»

#### 1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающихся должен освоить основной вид деятельности - выполнение сборки, монтажа и демонтажа электронных устройств и систем в соответствии с технической документацией и соответствующие ему общие компетенции и профессиональные компетенции:

Перечень общих компетенций<sup>1</sup>

Код	Наименование общих компетенций
<b>ОК 01</b>	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
<b>ОК 02</b>	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
<b>ОК 03</b>	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях
<b>ОК 04</b>	Эффективно взаимодействовать и работать в коллективе и команде
<b>ОК 05</b>	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
<b>ОК 06</b>	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
<b>ОК 07</b>	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
<b>ОК 08</b>	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания

<sup>1</sup> В данном подразделе указываются только те компетенции, которые формируются в рамках данного модуля и результаты которых будут оцениваться в рамках оценочных процедур по модулю.

	ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТУРИЗМА И СЕРВИСА»	СМК РГУТИС
		Лист 4

	необходимого уровня физической подготовленности
<b>ОК 09</b>	Пользоваться профессиональной документацией на государственном и иностранном языках

### 1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
<b>ВД 4</b>	Программирование встраиваемых систем с использованием интегрированных сред разработки
<b>ПК 4.1.</b>	Составлять алгоритмы и структуры программного кода для микропроцессорных систем
<b>ПК 4.2.</b>	Проектировать и программировать встраиваемые системы и интерфейсы оборудования с использованием языков программирования

### 1.1.3. В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт	<ul style="list-style-type: none"> <li>- формализация и алгоритмизация поставленных задач;</li> <li>- написание программного кода с использованием языков программирования, определения и манипулирования данными;</li> <li>- оформление программного кода в соответствии с установленными требованиями;</li> <li>- проверка и отладка программного кода;</li> <li>- разработка процедур проверки работоспособности и измерения характеристик программного обеспечения;</li> <li>- разработка тестовых наборов данных;</li> <li>- проверка работоспособности программного обеспечения;</li> <li>- рефакторинг и оптимизация программного кода;</li> <li>- исправление дефектов, зафиксированных в базе данных дефектов.</li> </ul>
Уметь	<ul style="list-style-type: none"> <li>- составлять программы на языке программирования для встраиваемых систем;</li> <li>- применять стандартные алгоритмы и конструкции языка программирования;</li> <li>- выбирать микроконтроллер для конкретной задачи встраиваемой системы;</li> <li>- выполнять требования технического задания по программированию встраиваемых систем;</li> <li>- создавать и отлаживать программы реального времени средствами программной эмуляции и на аппаратных макетах;</li> <li>- находить ошибки в программном коде для встраиваемой системы и оценивать степень их критичности;</li> <li>- производить тестирование и отладку встраиваемых систем на базе микроконтроллеров;</li> </ul>

	ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТУРИЗМА И СЕРВИСА»	СМК РГУТИС
		<i>Лист 5</i>

	<ul style="list-style-type: none"> <li>- выявлять причины неисправностей периферийных модулей встраиваемых систем.</li> </ul>
Знать	<ul style="list-style-type: none"> <li>- базовая функциональная схема микропроцессорной системы;</li> <li>- назначение и принцип действия составных блоков МПС;</li> <li>- режимы работы МПС;</li> <li>- способы организации связи МПС с внешней средой (исполнительными устройствами);</li> <li>- структура типовой системы управления (микроконтроллер);</li> <li>- организация микроконтроллерных систем;</li> <li>- состав микроконтроллера, назначение его функциональных блоков;</li> <li>- синтаксис и основные конструкции языка программирования для встраиваемой системы;</li> <li>- структура типовой встраиваемой системы на базе микроконтроллера и организации таких систем;</li> <li>- особенности программирования встраиваемых систем реального времени;</li> <li>- методы программной реализации типовых функций управления;</li> <li>- классификация, общие принципы построения и физические основы работы периферийных модулей встраиваемых систем;</li> <li>- способы подключения стандартных и нестандартных программных библиотек при разработке программного кода;</li> <li>- базовая функциональная схема встраиваемых систем на базе микроконтроллера;</li> <li>- виды и назначение программного обеспечения для разработки программного обеспечения для встраиваемых систем – интегрированных сред разработки (IDE);</li> <li>- методы тестирования и способы отладки встраиваемых систем;</li> <li>- причины неисправностей и возможных сбоев программного кода;</li> <li>- способы информационного взаимодействия различных устройств встраиваемых систем через проводные и беспроводные каналы связи, в том числе и сеть Интернет;</li> <li>- общее состояние производства и тенденции использования встраиваемых систем.</li> </ul>

## 1.2. Количество часов, отводимое на освоение профессионального модуля

Всего часов 257

в том числе в форме практической подготовки 72 часа

Из них на освоение МДК 173 часа

в том числе самостоятельная работа 4 часа

практики, в том числе:

учебная практика 36 часов

производственная практика (по профилю специальности) 36 часов

Промежуточная аттестация 12 часов



## 2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

### 2.1. Структура профессионального модуля

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля	Всего, час.	В т.ч. в форме практической подготовки	Объем профессионального модуля, ак. час.						
				Всего	Обучение по МДК				Практики	
					В том числе				Учебная	Производственная
Лабораторных и практических занятий	Курсовых работ (проектов) <sup>2</sup>	Самостоятельная работа <sup>3</sup>	Промежуточная аттестация							
1	2	3	4	5	6	7	8	9	10	11
ПК 4.1 ОК 01 – ОК 09	Микроконтроллеры и встраиваемые системы	<b>129</b>	36	<b>93</b>	39	-	2	-	<b>36</b>	-
ПК 4.2 ОК 01 – ОК 09	Разработка программного обеспечения для встраиваемых систем	<b>116</b>	36	<b>80</b>	26	-	2	-	-	<b>36</b>
	Аттестация по модулю	<b>12</b>								-
	<b>Всего:</b>	<b>257</b>	<b>72</b>	<b>173</b>	<b>65</b>	-	<b>4</b>	-	<b>36</b>	<b>36</b>

<sup>2</sup> Данная колонка указывается только для специальностей СПО.

<sup>3</sup> Самостоятельная работа в рамках образовательной программы планируется образовательной организацией в соответствии с требованиями ФГОС СПО в пределах объема профессионального модуля в количестве часов, необходимом для выполнения заданий самостоятельной работы обучающихся, предусмотренных тематическим планом и содержанием междисциплинарного курса.

## 2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовой проект	Объем, акад. ч / в том числе в форме практической подготовки, акад. Ч.
1	2	3
<b>Раздел 1. Микроконтроллеры и встраиваемые системы</b>		<b>129</b>
<b>МДК. 04.01 Микроконтроллеры и встраиваемые системы</b>		<b>93</b>
<b>Тема 1.1. Общие сведения о микропроцессорных системах</b>	<b>Содержание</b>	<b>20</b>
	История развития микропроцессоров (МП), современный уровень и тенденции развития микропроцессорных систем (МПС). МП, классификация МП. Структура простейшей МПС.	20
	Назначение и особенности различных типов МПС. Принстонская и гарвардская архитектуры МПС.	
	Структура простейшего МП. Функции МП.	
	Устройства управления с жесткой логикой. Устройства управления с программируемой логикой. Микропрограммное управление.	
	Система команд МП. Рабочий цикл МП.	
	Режимы работы МПС. Программный обмен. Система прерываний МП. Механизм обмена по прерываниям. Обмен в режиме ПДП.	
	Классификация и функции памяти МПС. Классификация ОЗУ, типы и виды ОЗУ. КЭШ память. Классификация ПЗУ, типы и виды ПЗУ. Способы адресации в МПС.	
Организация связи МПС с внешней средой. Функции устройств ввода-вывода. Принципы построения портов ввода-вывода.		



<b>Тема 1.2. Встраиваемые системы на основе микроконтроллеров</b>	<b>Содержание</b>	<b>12</b>
	Обзор современных микроконтроллеров (МК). Классификация МК. Модульная организация МК.	12
	Структура процессорного ядра МК. Система команд МК. Память МК.	
	Порты ввода-вывода, таймеры, модуль прерываний МК	
	Минимизация энергопотребления в системах с МК. Тактовые генераторы МК.	
	Аппаратные средства обеспечения надежной работы МК.	
	Дополнительные модули МК: последовательного ввода-вывода, аналогового ввода-вывода.	
	Аппаратные и программные средства для разработки приложений на базе МК.	
Функциональные блоки микроконтроллера. Конфигурирование МК.		
<b>Тема 1.3. Структура программы и основные конструкции языка Си</b>	<b>Содержание</b>	<b>59</b>
	Вводные понятия языка С. Структура программы на С.	20
	Типы данных в С. Переменные в С. Константы в С.	
	Арифметические и логические операторы языка С.	
	Операторы ветвления в С.	
	Циклические конструкции в С.	
	Указатели и адреса переменных в С.	
	Работа с функциями в С. Особенности передачи данных при обращении к функции в С.	
	Структуры в С. Указатели и адреса переменных в С.	
	Массивы и строки в С.	
	Стандартные функции ввода/вывода в С.	
	<b>В том числе практических занятий и лабораторных работ</b>	<b>39</b>
	Основные характеристики и особенности архитектуры МК.	4
	Выполнение логических и арифметических команд.	7
Выполнение циклических конструкций и операторов ветвления.	6	
Работа с цифровыми портами ввода-вывода.	6	

	Организация циклов и временных задержек.	4
	Организация подпрограмм.	4
	Работа с макросами.	4
	Обработка прерываний.	4
<b>Примерная тематика самостоятельной учебной работы при изучении раздела 1</b>		
	1. Подготовка устного доклада на тему «История развития микроконтроллеров». 2. Разработка SWOT-анализа по теме «Основные семейства, особенности и характеристики микроконтроллеров фирм Atmel, Texas Instruments, Microchip, STMicroelectronics». 3. Современные интерфейсы и их применение в микроконтроллерных системах. 4. Выполнение коллективного проекта по теме: «Проектирование микроконтроллерной системы управления «Умный дом». 5. Подбор микроконтроллера по условиям технического задания.	2
<b>Учебная практика раздела 1</b>  <b>Виды работ (изучение микроконтроллера по выбору образовательной организации)</b>		
	1. Установка программного обеспечения. Конфигурирование микроконтроллера, создание проекта, компиляция, прошивка. 2. Работа с регистрами микроконтроллера. Библиотеки для разработчика.. 3. Управление портами ввода-вывода. 4. Конвертирование проекта для микроконтроллера на языке C в проект C++. 5. Обработка входных дискретных сигналов. Устранение дребезга контактов, борьба с импульсными помехами. 6. Разработка и использование классов в C++. Создание класса обработки дискретных сигналов. 7. Создание и использование библиотек для микроконтроллера. 8. Параллельные процессы. Выполнение задач в фоновом режиме при помощи прерывания от таймера. 9. Разработка программ, состоящих из нескольких исходных файлов. Определение и объявление переменных, область видимости. Режимы компиляции. 10. Установка конфигурации таймеров с помощью библиотек. Логика работы прерывания таймера. 11. Интерфейс UART в микроконтроллере. Использование прерывания UART.	36



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТУРИЗМА И СЕРВИСА»

СМК  
РГУТИС

Лист 10

12. Работа с АЦП		
<b>Раздел 2 Разработка программного обеспечения для встраиваемых систем</b>		<b>116</b>
<b>МДК. 04.02 Разработка программного обеспечения для встраиваемых систем</b>		<b>80</b>
<b>Тема 2.1. Инструментальные средства разработки программного обеспечения для встраиваемых систем</b>	<b>Содержание</b>	<b>26</b>
	Современный уровень и тенденции развития инструментальных сред разработки (IDE) для встраиваемых систем.	26
	Классификация средств разработки. Аппаратные и программные средства.	
	Особенности применения языков высокого уровня в разработке приложений пользователя.	
	Особенности разработки приложений работы в системе реального времени.	
	Библиотеки встроенных функций в составе IDE.	
	Программаторы и отладчики.	
Компиляторы языка C.		
<b>Тема 2.2. Тестирование и отладка разработанного программного кода</b>	<b>Содержание</b>	<b>52</b>
	Единая система программной документации. Назначение, виды документов.	26
	Понятие программного тестирования. Виды тестов.	
	Составление плана тестирования.	
	Разработка модулей тестирования. Моделирование ситуаций.	
	Создание и использование разнообразных входных данных.	
	Поиск вероятных ошибок и сбоев в функционировании ПО.	
	Нахождение несоответствия интерфейса программы техническому описанию.	
	Поиск ошибок в логике работы программы и в документации на программу.	
	Рефакторинг программного обеспечения.	
	Контроль версий программы.	
	Оформление результатов тестирования и отладки программного обеспечения.	
	<b>В том числе практических занятий и лабораторных работ</b>	
Подключение к микроконтроллеру семисегментного светодиодного индикатора.	2	
Подключение к микроконтроллеру светодиодной матрицы.	2	



	Подключение к микроконтроллеру RGB-светодиода.	2
	Подключение к микроконтроллеру светодиодного шкального индикатора.	2
	Подключение к микроконтроллеру аналогового датчика температуры.	2
	Подключение к микроконтроллеру энкодера.	2
	Построение программируемого счетчика-таймера на микроконтроллере.	2
	Подключение к микроконтроллеру модуля знакосинтезирующего ЖКИ.	2
	Подключение к микроконтроллеру модуля графического ЖКИ с сенсорным экраном.	2
	Подключение к микроконтроллеру серводвигателя.	2
	Подключение к микроконтроллеру шагового двигателя.	2
	Подключение к микроконтроллеру датчика по цифровому интерфейсу SPI.	2
	Подключение к микроконтроллеру датчика по цифровому интерфейсу I2C.	2
	<b>Примерная тематика самостоятельной учебной работы при изучении раздела № 2</b> 1. Обзор средств разработки ПО для встраиваемых систем. 2. Поиск ошибок в программном коде и оптимизация алгоритма предложенной программы. 3. Разработка SWOT-анализа по теме «Внутрисхемные эмуляторы, эмуляторы ПЗУ, мониторы отладки» 4. Разработка алгоритма для пользовательской функции работы с цифровым датчиком по интерфейсу 1-Wire. 5. Разработка алгоритма для пользовательской функции работы с цифровым датчиком по интерфейсу SPI. 6. Разработка алгоритма для пользовательской функции работы с цифровым датчиком по интерфейсу I2C.	2
	<b>Производственная практика раздела № 2</b> <b>Виды работ</b> 1. Установка инструментальной среды разработки программного обеспечения для встраиваемых микроконтроллерных систем. 2. Настройка интерфейса пользователя и параметров среды. Установка и настройка компилятора. 3. Анализ технического задания на разработку программного обеспечения. 4. Разработка алгоритма программы для встраиваемой микроконтроллерной системы. 5. Написание программы на специализированном языке для встраиваемой микроконтроллерной системы. 6. Подбор стандартных библиотек для реализации проекта. 7. Программирование встраиваемой микроконтроллерной системы. 8. Проведение отладки программного обеспечения микропроцессорных систем с помощью аппаратно-программных	36



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТУРИЗМА И СЕРВИСА»

СМК  
РГУТИС

*Лист 12*

средств.	
9. Проверка функциональности программного обеспечения.	
10. Составление отчетной программной документации.	
<b>Всего</b>	<b>257</b>



### **3. Методические указания по проведению практических занятий, занятий в форме практической подготовки и самостоятельной работе**

В рамках освоения профессионального модуля реализуются следующие виды занятий: лекционные занятия, практические занятия, в том числе в форме практической подготовки.

#### **3.1. Тематика и содержание практических занятий**

##### **Тема 1.3. Структура программы и основные конструкции языка Си**

Содержание: Практическое занятие №1 Основные характеристики и особенности архитектуры МК.

Содержание: Практическое занятие №2 Выполнение логических и арифметических команд.

Содержание: Практическое занятие №3 Выполнение циклических конструкций и операторов ветвления.

Содержание: Практическое занятие №4 Работа с цифровыми портами ввода-вывода.

Содержание: Практическое занятие №5 Организация циклов и временных задержек.

Содержание: Практическое занятие №6 Организация подпрограмм.

Содержание: Практическое занятие №7 Работа с макросами.

Содержание: Практическое занятие №8 Обработка прерываний.

##### **Тема 2.2. Тестирование и отладка разработанного программного кода**

Содержание: Практическое занятие №1 Подключение к микроконтроллеру семисегментного светодиодного индикатора.

Содержание: Практическое занятие №2 Подключение к микроконтроллеру светодиодной матрицы.

Содержание: Практическое занятие №3 Подключение к микроконтроллеру RGB-светодиода.

Содержание: Практическое занятие №4 Подключение к микроконтроллеру светодиодного шкального индикатора.

Содержание: Практическое занятие №5 Подключение к микроконтроллеру аналогового датчика температуры.

Содержание: Практическое занятие №6 Подключение к микроконтроллеру энкодера.

Содержание: Практическое занятие №7 Построение программируемого счетчика-таймера на микроконтроллере.



Содержание: Практическое занятие №8 Подключение к микроконтроллеру модуля знаковсинтезирующего ЖКИ.

Содержание: Практическое занятие №9 Подключение к микроконтроллеру модуля графического ЖКИ с сенсорным экраном.

Содержание: Практическое занятие №10 Подключение к микроконтроллеру серводвигателя.

Содержание: Практическое занятие №11 Подключение к микроконтроллеру шагового двигателя.

Содержание: Практическое занятие №12 Подключение к микроконтроллеру датчика по цифровому интерфейсу SPI.

Содержание: Практическое занятие №13 Подключение к микроконтроллеру датчика по цифровому интерфейсу I2C.

### **3.2. Тематика и содержание самостоятельной работы**

Самостоятельная работа является неотъемлемой частью образовательного процесса, связанного с формированием компетенций обучающихся.

Цель и задачи самостоятельной работы обучающегося: формирование и воспитание многогранной, творческой личности, со сложившимися приоритетами, правилами поведения, с системой ценностей и верными представлениями о мире в целом.

Целью самостоятельной (внеаудиторной) работы обучающихся является обучение навыкам работы с научно-теоретической, периодической, научно-технической литературой и технической документацией, необходимыми для углубленного изучения дисциплины, а также развитие у них устойчивых способностей к самостоятельному изучению и изложению полученной информации.

Самостоятельная работа выполняется в форме проработки конспектов занятий, учебной и специальной технической литературы (по вопросам к параграфам, главам учебных пособий, составленным преподавателем) и подготовки к практическим работам с использованием методических рекомендаций преподавателя; оформление практических работ; отчетов и подготовка к их защите.

### **Тематика и содержание**

#### **Раздел 1. Микроконтроллеры и встраиваемые системы**

Содержание: Самостоятельная работа 1. Подготовка устного доклада на тему «История развития микроконтроллеров».

Содержание: Самостоятельная работа 2. Разработка SWOT-анализа по теме «Основные семейства, особенности и характеристики микроконтроллеров фирм Atmel, Texas Instruments, Microchip, STMicroelectronics».



Содержание: Самостоятельная работа 3. Современные интерфейсы и их применение в микроконтроллерных системах.

Содержание: Самостоятельная работа 4. Выполнение коллективного проекта по теме: «Проектирование микроконтроллерной системы управления «Умный дом».

Содержание: Самостоятельная работа 5. Подбор микроконтроллера по условиям технического задания.

## **Раздел 2 Разработка программного обеспечения для встраиваемых систем**

Содержание: Самостоятельная работа 1. Обзор средств разработки ПО для встраиваемых систем.

Содержание: Самостоятельная работа 2. Поиск ошибок в программном коде и оптимизация алгоритма предложенной программы.

Содержание: Самостоятельная работа 3. Разработка SWOT-анализа по теме «Внутрисхемные эмуляторы, эмуляторы ПЗУ, мониторы отладки»

Содержание: Самостоятельная работа 4. Разработка алгоритма для пользовательской функции работы с цифровым датчиком по интерфейсу 1-Wire.

Содержание: Самостоятельная работа 5. Разработка алгоритма для пользовательской функции работы с цифровым датчиком по интерфейсу SPI.

Содержание: Самостоятельная работа 6. Разработка алгоритма для пользовательской функции работы с цифровым датчиком по интерфейсу I2C.

## **4. Фонд оценочных средств профессионального модуля**

### **4.1. Формы аттестации по профессиональному модулю**

Элемент модуля	Форма контроля и оценивания	
	Промежуточная аттестация	Текущий контроль
МДК.04.01 Микроконтроллеры и встраиваемые системы	Другие формы контроля (Контрольная работа)	оценка практических работ, оценка самостоятельной работы, устный опрос
МДК.04.02 Разработка программного обеспечения для встраиваемых систем	Другие формы контроля (Контрольная работа)	оценка практических работ, оценка самостоятельной работы, устный опрос
УП.04.01 Учебная практика	Дифференцированный зачет	Выполнение индивидуального задания
ПП.04.01 Производственная практика по профилю специальности	Дифференцированный зачет	Выполнение индивидуального задания

### **4.2. Результаты освоения профессионального модуля**

#### **Профессиональные, общие компетенции, личностные результаты**



В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Профессиональные компетенции	Показатели оценки результата
ПК 4.1. Составлять алгоритмы и структуры программного кода для микропроцессорных систем	<ul style="list-style-type: none"><li>- правильность написания программного кода с использованием языков программирования;</li><li>- правильность оформления программного кода в соответствии с установленными требованиями;</li><li>- верное осуществление проверки и отладки программного кода;</li><li>- верное составление программы на языке программирования для встраиваемых систем;</li><li>- правильность применения стандартных алгоритмов и конструкций языка программирования;</li><li>- правильность выбора микроконтроллера для конкретной задачи встраиваемой системы;</li><li>- правильность выполнения требования технического задания по программированию встраиваемых систем;</li><li>- правильность определения назначения и принципа действия составных блоков МПС и их режимов;</li><li>- верное определение состава микроконтроллера, назначения его функциональных блоков;</li><li>- правильность использования синтаксиса и основных конструкций языка программирования для встраиваемой системы;</li><li>- правильность понимания структуры типовой встраиваемой системы на базе микроконтроллера и организации таких систем;</li><li>- правильность выбора метода программной реализации типовых функций управления;</li><li>- правильность выбора способа подключения стандартных и нестандартных программных библиотек при разработке программного кода.</li></ul>
ПК 4.2. Проектировать и программировать встраиваемые системы и интерфейсы оборудования с использованием языков программирования	<ul style="list-style-type: none"><li>- правильность разработки процедур проверки работоспособности программного обеспечения;</li><li>- правильность разработки тестовых наборов данных для программы;</li><li>- правильность проведения процедуры тестирования и отладки встраиваемых систем на базе микроконтроллеров;</li><li>- правильность осуществления рефакторинга и оптимизации программного кода под требования встраиваемой системы;</li><li>- правильность нахождения ошибок в программном коде для встраиваемой системы;</li><li>- верное оценивание степени критичности ошибок в коде программы;</li><li>- правильность определения вида и назначения программного обеспечения для разработки программного обеспечения для встраиваемых систем;</li><li>- правильность применения методов тестирования и</li></ul>



	способов отладки встраиваемых систем; - верное определение причин неисправностей и возможных сбоев программного кода
--	--

Общие компетенции	Показатели оценки результата
ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	- обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач.
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	- демонстрация ответственности за принятые решения - обоснованность самоанализа и коррекция результатов собственной работы;
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	- взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; - обоснованность анализа работы членов команды (подчиненных)
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	- грамотность устной и письменной речи, - ясность формулирования и изложения мыслей
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	- соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик,
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	- эффективность выполнения правил техники безопасности и охраны труда во время учебных занятий, при прохождении учебной и производственной практик; - знание и использование ресурсосберегающих технологий в области электроники и приборостроения



ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	- эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;
ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на иностранном языке.

### 4.3. Требования к портфолио

Тип портфолио: *смешанный тип*

- Доклады и сообщения по отдельным темам междисциплинарного курса профессионального модуля, предусмотренные программой.
- Отчеты по практическим работам, выполненным при изучении междисциплинарного курса профессионального модуля.

#### Дополнительные материалы:

1. Грамоты, дипломы
2. Сертификаты за участие в мероприятиях колледжа и Московской области.
3. Приказы о поощрениях.

#### Требования:

Требования к презентации и защите портфолио: - не предусмотрено

Требования к структуре и оформлению портфолио: - не предусмотрено

Обязательно наличие всего перечня, входящего в состав обязательной части портфолио.

Специальных требований к оформлению нет.

#### Показатели оценки портфолио на аттестации по модулю:

Коды проверяемых компетенций или их сочетаний	Показатели оценки результата	Оценка (да / нет)
ОК1-ОК9; ПК 4.1-ПК 4.2	Наличие характеристики с места практики.	Да/нет
ОК1-ОК9; ПК 4.1-ПК 4.2	Наличие аттестационного листа	Да/нет
ОК1-ОК9; ПК 4.1-ПК 4.2	Наличие и качественное выполнение отчетов по практикам, содержание которых соответствует выданному заданию.	Да/нет
ОК1-ОК9; ПК 4.1-ПК 4.2	Оформление отчетов по практикам Положения об оформлении текстовых документов.	Да/нет
ОК1-ОК9; ПК 4.1-ПК 4.2	Защита отчетов о прохождении практики.	Да/нет



**4.4. Оценка освоения теоретического курса профессионального модуля – МДК:**  
**4.4.1 Типовые задания для оценки освоения МДК.04.01 Микроконтроллеры и встраиваемые системы**  
**Примерный перечень вопросов для формирования заданий текущего контроля:**

1. Что такое микроконтроллеры, микропроцессоры и сигнальные процессоры
2. Области применения микроконтроллеров
3. Целочисленные двоичные коды
4. Запись текстов двоичным кодом
5. Запись десятичных чисел двоичным кодом
6. Представление чисел в двоичном коде с плавающей запятой
7. Масочные ПЗУ, ППЗУ, РПЗУ
8. EEPROM и flash память
9. Внутреннее устройство статического ОЗУ
10. Команды микропроцессора
11. Системная шина микропроцессора
12. Принципы построения параллельного порта. Подключение внешних устройств к микропроцессору
13. Принципы построения последовательных портов. Виды последовательных портов
14. Принципы построения схем таймеров микропроцессоров
15. Архитектура микроконтроллеров MCS-51
16. Система команд микроконтроллеров MCS-51
17. Виды адресации
18. Инструкции микроконтроллеров MCS-51
19. Особенности построения параллельных портов микроконтроллеров MCS-51
20. Особенности построения памяти микроконтроллеров семейства MCS-51
21. Внутренние таймеры микроконтроллера, особенности их применения
22. Устройство и особенности применения последовательного порта микроконтроллеров семейства MCS-51
23. Особенности проектирования схем на микроконтроллерах
24. Особенности проектирования системы питания для устройств на микроконтроллерах
25. Языки программирования для микроконтроллеров
26. Применение подпрограмм при программировании. Понятие подпрограммы процедуры и подпрограммы функции
27. Написание программ для микропроцессоров. Понятие программы-монитора и операционной системы реального времени
28. Понятие структурного программирования. Применение комментариев
29. Понятие многофайлового и многомодульного программирования

**Наименование проверяемой компетенции**  
**ОК01-ОК09, ПК 4.1. Составлять алгоритмы и структуры программного кода**  
**для микропроцессорных систем**

*Часть 1: Вопросы с выбором ответа*

1. Что такое микроконтроллер?
  - а) Устройство для управления компьютером



- б) Миниатюрный компьютер на одном кристалле, включающий процессор, память и периферию
  - с) Устройство для усиления сигналов
  - d) Программа для разработки ПО
2. Какие из перечисленных компонентов обычно входят в состав микроконтроллера?
- а) CPU, ОЗУ, ПЗУ, таймеры, GPIO
  - б) GPU, жесткий диск, монитор
  - с) Сеть Wi-Fi, Bluetooth, динамики
  - d) Клавиатура, мышь, принтер
3. Что такое GPIO в микроконтроллере?
- а) Графический процессор
  - б) Программное обеспечение для управления
  - с) Порты ввода-вывода общего назначения
  - d) Устройство для хранения данных
4. Какая архитектура чаще всего используется в микроконтроллерах?
- а) x86
  - б) ARM
  - с) RISC-V
  - d) MIPS
5. Что такое встраиваемая система?
- а) Компьютер общего назначения
  - б) Специализированная система, предназначенная для выполнения конкретных задач
  - с) Сервер для хранения данных
  - d) Игровая консоль

**Задание 1: Соотнесите компонент микроконтроллера с его описанием**

Компонент	Описание
1. CPU	а) Память, которая хранит данные только для чтения (например, прошивку)
2. ОЗУ (RAM)	б) Память для временного хранения данных во время выполнения программы
3. ПЗУ (ROM)	с) Центральный процессор, выполняющий инструкции программы
4. GPIO	д) Устройство для измерения аналоговых сигналов и преобразования их в цифровые
5. АЦП (ADC)	е) Порты ввода-вывода общего назначения для подключения внешних устройств

**Задание 2: Соотнесите термин с его определением**

Термин	Определение
1. Прерывание	а) Устройство для управления временными интервалами и событиями
2. Таймер	б) Сигнал, который временно приостанавливает выполнение программы для обработки события
3. ШИМ (PWM)	с) Метод управления аналоговыми устройствами с помощью цифровых сигналов



Термин	Определение
4. UART	d) Интерфейс для последовательной передачи данных между устройствами
5. Встраиваемая система	e) Специализированная система, предназначенная для выполнения конкретных задач

**Задание 3: Соотнесите микроконтроллер с его особенностями**

**Микроконтроллер Особенности**

- |                      |  |
|----------------------|--|
| 1. Arduino Uno       | a) Основан на архитектуре ARM, часто используется в IoT-устройствах  |
| 2. STM32             | b) Популярный микроконтроллер для начинающих, основан на ATmega328P  |
| 3. ESP32             | c) Имеет встроенный Wi-Fi и Bluetooth, подходит для IoT-проектов     |
| 4. Raspberry Pi Pico | d) Основан на RP2040, поддерживает MicroPython и C/C++               |
| 5. PIC16F877A        | e) Микроконтроллер от Microchip, часто используется в промышленности |

**Задание 4: Соотнесите тип памяти с его характеристикой**

**Тип памяти Характеристика**

- |                 |  |
|-----------------|--|
| 1. ОЗУ (RAM)    | a) Энергонезависимая память, используется для хранения прошивки            |
| 2. ПЗУ (ROM)    | b) Быстрая память для временного хранения данных во время работы программы |
| 3. EEPROM       | c) Память, которая может быть перезаписана, но медленнее, чем ОЗУ          |
| 4. Flash-память | d) Энергонезависимая память, используется для хранения данных и программ   |
| 5. Кэш-память   | e) Быстрая память, используемая процессором для ускорения доступа к данным |

**Задание 5: Соотнесите интерфейс с его назначением**

**Интерфейс Назначение**

- |         |  |
|---------|--|
| 1. I2C  | a) Интерфейс для подключения датчиков и устройств с использованием двух проводов |
| 2. SPI  | b) Интерфейс для высокоскоростной передачи данных между устройствами             |
| 3. UART | c) Интерфейс для последовательной передачи данных между двумя устройствами       |
| 4. CAN  | d) Интерфейс для обмена данными в автомобильных и промышленных системах          |
| 5. USB  | e) Интерфейс для подключения периферийных устройств к компьютеру                 |

**Задача 1: Основы микроконтроллеров**

**Вопрос:** Что из перечисленного является основной функцией микроконтроллера?

- Управление графическим интерфейсом пользователя
- Выполнение специализированных задач с использованием встроенных периферийных устройств
- Хранение больших объемов данных
- Подключение к интернету через Wi-Fi



---

**Задача 2: Архитектура микроконтроллеров**

**Вопрос:** Какая архитектура чаще всего используется в современных микроконтроллерах?

- a) x86
- b) ARM
- c) PowerPC
- d) SPARC

---

**Задача 3: Память микроконтроллеров**

**Вопрос:** Какая память в микроконтроллере используется для хранения программы (прошивки)?

- a) ОЗУ (RAM)
- b) ПЗУ (ROM)
- c) Кэш-память
- d) Регистры процессора

---

**Задача 4: Периферия микроконтроллеров**

**Вопрос:** Что такое GPIO в микроконтроллере?

- a) Графический процессор
- b) Порты ввода-вывода общего назначения
- c) Устройство для хранения данных
- d) Интерфейс для подключения дисплея

---

**Задача 5: Прерывания**

**Вопрос:** Для чего используются прерывания в микроконтроллерах?

- a) Для увеличения тактовой частоты процессора
- b) Для временного приостановления основной программы и обработки внешних событий
- c) Для хранения данных в энергонезависимой памяти
- d) Для подключения к интернету

---

**Задача 6: АЦП (Аналого-цифровой преобразователь)**

**Вопрос:** Какое устройство используется для преобразования аналогового сигнала в цифровой?

- a) ЦАП (DAC)
- b) АЦП (ADC)
- c) Таймер
- d) ШИМ (PWM)

---

**Задача 7: ШИМ (PWM)**

**Вопрос:** Что такое ШИМ (PWM) в микроконтроллере?

- a) Метод управления аналоговыми устройствами с помощью цифровых сигналов
- b) Устройство для хранения данных



- c) Интерфейс для подключения дисплея
  - d) Метод шифрования данных
- 

#### ***Задача 8: Встраиваемые системы***

**Вопрос:** Что из перечисленного является примером встраиваемой системы?

- a) Ноутбук
  - b) Смартфон
  - c) Стиральная машина с микроконтроллером
  - d) Сервер
- 

#### ***Задача 9: Интерфейсы связи***

**Вопрос:** Какой интерфейс используется для подключения датчиков с использованием двух проводов?

- a) SPI
  - b) I2C
  - c) UART
  - d) USB
- 

#### ***Задача 10: Языки программирования***

**Вопрос:** Какой язык программирования чаще всего используется для разработки ПО для микроконтроллеров?

- a) Python
  - b) Java
  - c) C/C++
  - d) JavaScript
- 

#### ***Задача 11: Таймеры***

**Вопрос:** Для чего используются таймеры в микроконтроллерах?

- a) Для хранения данных
  - b) Для измерения временных интервалов и генерации сигналов
  - c) Для подключения к интернету
  - d) Для управления графическим интерфейсом
- 

#### ***Задача 12: Энергопотребление***

**Вопрос:** Какая особенность микроконтроллеров делает их подходящими для использования в устройствах с батарейным питанием?

- a) Высокая тактовая частота
  - b) Низкое энергопотребление
  - c) Большой объем памяти
  - d) Поддержка Wi-Fi
-



### **Задача 13: Flash-память**

**Вопрос:** Для чего используется Flash-память в микроконтроллерах?

- a) Для временного хранения данных
- b) Для хранения программы (прошивки)
- c) Для увеличения тактовой частоты
- d) Для подключения к интернету

---

### **Задача 14: UART**

**Вопрос:** Что такое UART?

- a) Интерфейс для последовательной передачи данных между двумя устройствами
- b) Устройство для управления графическим интерфейсом
- c) Метод шифрования данных
- d) Интерфейс для подключения к интернету

---

### **Задача 15: CAN**

**Вопрос:** Где чаще всего используется интерфейс CAN?

- a) В автомобильных системах
- b) В смартфонах
- c) В игровых консолях
- d) В ноутбуках

**Пример задания на практическое занятие:**

#### **Основные характеристики и особенности архитектуры МК.**

##### **1 Цель работы**

Целью работы является изучение архитектуры МК ATmega16 и приобретение базовых навыков по созданию проектов, компиляции и настройке программ в интегрированной среде разработки Atmel Studio

##### **2 Основные положения**

###### **2.1 Общая характеристика МК семейства Mega компании Atmel**

Представители семейства Mega являются 8-розрядными МК, предназначенными для встроенных добавлений. Они добываются CMOS-технологией с низким использованием энергии, которая в симбиозе с усовершенствованной RISC-архитектурой позволяет достичь наилучшего соотношения быстродействия/энергопотребления.

МК данного семейства являются более развитыми представителями МК AVR компании Atmel

###### **Особенности МК AVR семейства Mega:**

- FLASH-память программ объемом 8 ... 128 Кбайт (число циклов стирания/записи не меньше 1000);
- оперативная память объемом 1 ... 4 Кбайт;
- память данных на основе ЕСПЗП (EEPROM) объемом 512 байт ... 4Кбайт (число циклов стирания/записи не меньше 100000);
- возможность защиты от считывания и модификации памяти программ и данных;



- возможность программирования непосредственно в системе через последовательный интерфейс SPI и JTAG;
- возможность самопрограммирования;
- возможность внутрисхемной настройки относительно к стандарту IEEE 1149.1 (JTAG);
- разные способы синхронизации: встроенный RC-генератор с внутренней либо внешней RC-цепью или с внешним резонатором;
- наличие нескольких режимов пониженного энергопотребления;
- наличие детектора снижения напряжения питания (brown-out detector, BOD);
- возможность программного понижения частоты тактового генератора.

Основные характеристики процессора МК семейства Mega являются такие же как и МК других семейств – Classic и Tiny:

- полностью статическая архитектура; минимальная тактовая частота может равняться нулю;
- АЛУ подключено непосредственно к регистрам общего назначения;
- большинство команд выполняются за один машинный цикл;

Также процессор МК семейства Mega имеет низкие характеристики, присущие именно этому семейству:

- наибольшее число источников прерываний (до 27 источников)
- наличие программного стека во всех моделях семейства;
- наличие аппаратного умножителя.

Все характеристики подсистемы ввода/вывода МК семейства Mega такие, как и у МК других семейств:

- программное конфигурирование и выбор портов ввода/вывода;
- выводы могут быть запрограммированы как входные или как выходные независимо друг от друга;
- входные буферы с триггером Шмидта на всех выходах;
- возможность подключения ко всем входам внутренних резисторов подтягивания (сопротивление резисторов составляет 35 ... 120 кОм).

МК семейства Mega имеют наиболее богатый набор периферийных устройств (ПП). При этом в большинстве моделей есть все ПП, которые вообще встречаются в составе МК AVR. Этими устройствами являются:

- 8-разрядные таймеры/счетчики (таймеры T0 и T2). У некоторых моделей эти таймеры/счетчики могут работать как часы реального времени (в асинхронном режиме);
- 16-разрядные таймеры/счетчики (таймеры T1 и T3);
- сторожевой таймер WDT;
- генераторы сигнала с ШИМ разрядностью 8 бит (один из режимов работы 8-разрядных таймеров/счетчиков T0 и T2);
- одно-, двух- и трехканальные генераторы сигнала с ШИМ регулируемой разрядностью (один из режимов работы 16-разрядных таймеров T1 и T3). Разрешение ШИМ-сигнала для разных моделей составляет 8 ... 10 бит или 1 ... 16 бит;
- аналоговый компаратор;



- многоканальный 10-разрядный АЦП как с несимметричными, так и с дифференциальными входами;
  - полнодуплексный универсальный асинхронный приемник (UART);
  - полнодуплексный универсальный синхронный/асинхронный приемник (USART);
  - последовательный синхронный интерфейс SPI;
  - последовательный двухпроводный интерфейс TWI (аналог интерфейса I2C).
- Ядро МК AVR семейства Mega, как и ядро МК семейств Classic и Tiny, выполнено по усовершенствованной RISC-архитектуре (enhanced RISC). АЛП, что выполняет все вычисления, есть подключен напрямую до 32-х рабочих регистров, которые объединены в регистровый файл.
- Благодаря этому АЛП выполняет одну операцию (чтение содержимого регистров, выполнение операции и запись результата обратно в регистровый файл) за один машинный цикл. Практически каждая из команд (за исключением команд, у которых одним из операндов является 16-разрядный адрес) занимает одну ячейку памяти программ.

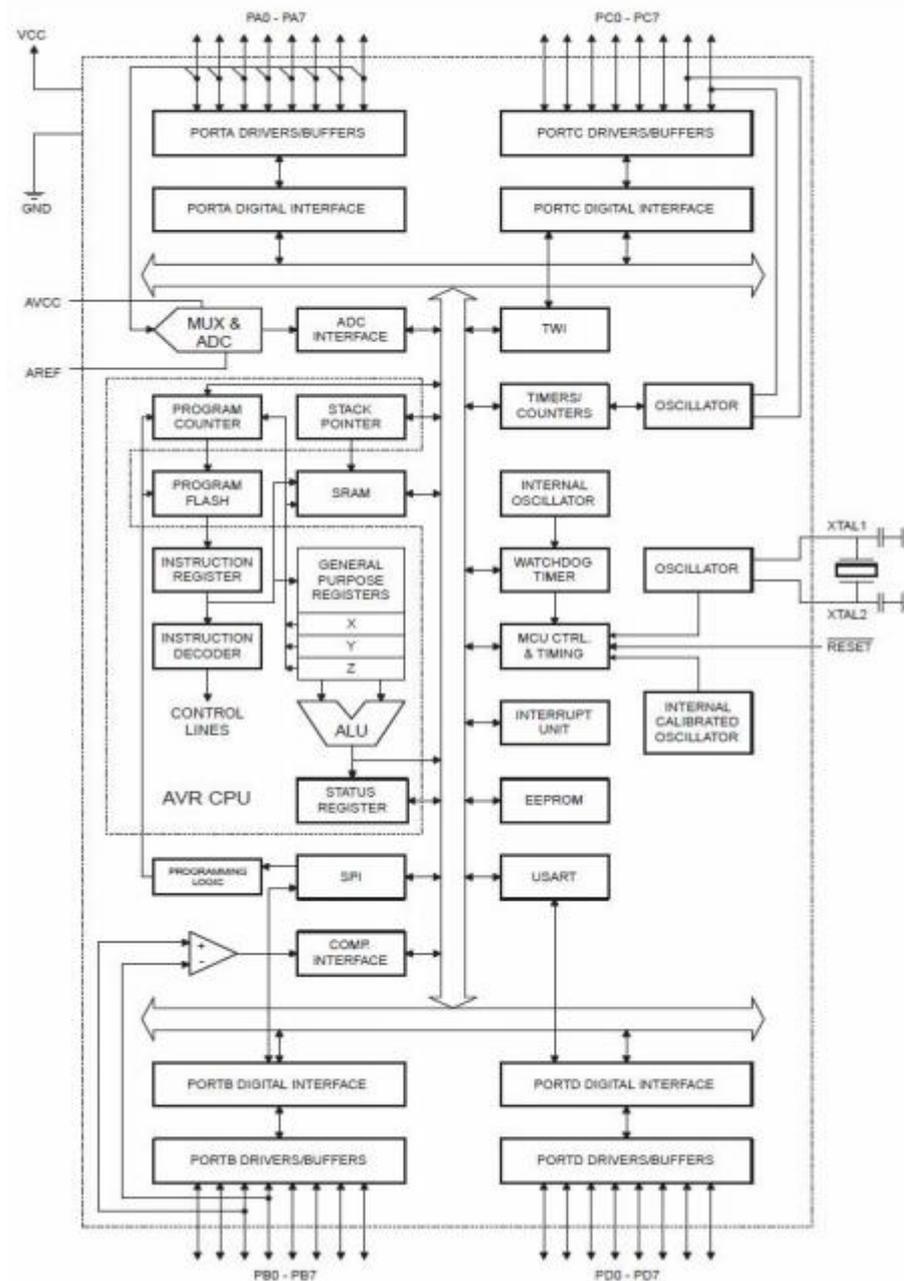


Рисунок 1.1 – Структурная схема МК ATmega 16

В МК AVR реализована Гарвардская архитектура, которая характеризуется отдельной памятью программ и данных, каждая из которых имеет собственные шины доступа к ним. Такая организация позволяет одновременно работать как с памятью программ, так и с памятью данных. Разделение шин доступа позволяет использовать для каждого типа шины памяти различной разрядности, причем способы адресации и доступа к каждому типу памяти также различны.

Еще одним решением, направленным на повышение быстродействия, является использование технологии конвейеризации. Конвейеризация заключается в том, что во

время выполнения текущей команды проводится выборка из памяти и дешифрование кода следующей команды.

Причем, поскольку длительность машинного цикла МК AVR составляет всего один период тактового генератора, они могут обеспечивать такую же производительность, что и RISC-МК других фирм, но по более низкой тактовой частоте.

Далее будем рассматривать только МК ATmega16, структурная схема которого показана на рис. 1.1.

### Это типичный представитель семейства Mega. 2.2 Архитектура МК ATmega16

Микросхема изготавливается в корпусах формата PDIP и TQFP. Расположение и обозначение выводов микросхемы ATmega16 в PDIP-корпусе показано на рис. 1.2.

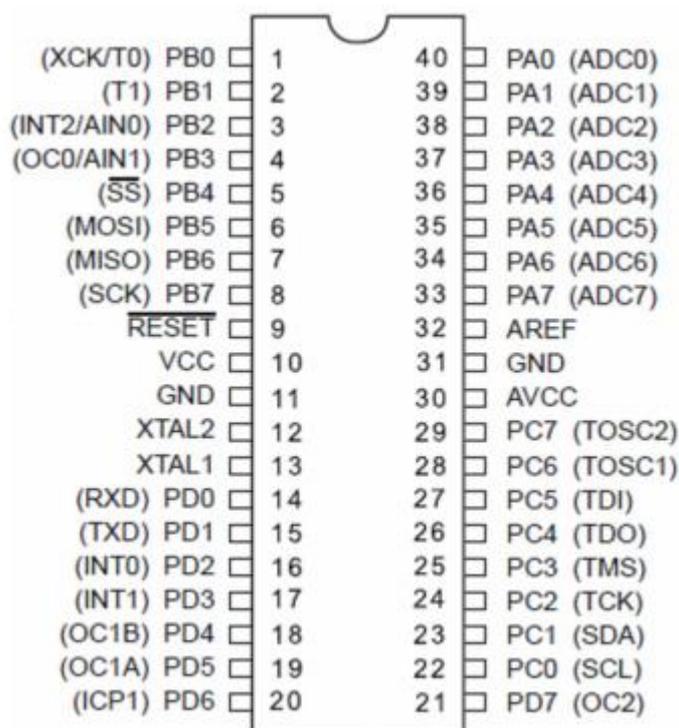


Рисунок 1.2 – Выводы микросхемы ATmega 16 в корпусе PDIP

### 2.3 Работа в среде программирования Atmel Studio

Atmel Studio IDE (Integrated Development Environment – интегрированная среда разработки – профессиональный программный продукт от компании AVR).

До версии 5 назывался AVR Studio. Распространяется бесплатно. Среда позволяет создавать программы для различных моделей 8-разрядных МК AVR язык Assembler/C/C++, а также осуществлять их компиляцию, симуляцию и налаживание. Далее рассматривается версия Atmel Studio. Дальнейшее описание работы в среде практически не отличается от предыдущих версий. Итак, для начала работы необходимо создать новый проект. Проект состоит из одного или нескольких файлов с кодом с возможностью подключение библиотек. Выбираем пункт меню File > New > Project. В следующем окне выбираем язык Assembler и задаем название и расположение папки в соответствии с указаниями преподавателя. В следующем окне необходимо выбрать

модель МК. Во всех дальнейших работах будем использовать модель ATmega16. Новый проект имеет вид рис 1.3.

Автоматически создается файл – main.asm с минимальным шаблоном ассемблерного кода для МК, который представляет собой вечный цикл:

```
start: // комментарии:
```

```
inc r16 // r16=r16+1
```

```
rjmp start // переход к метке start
```

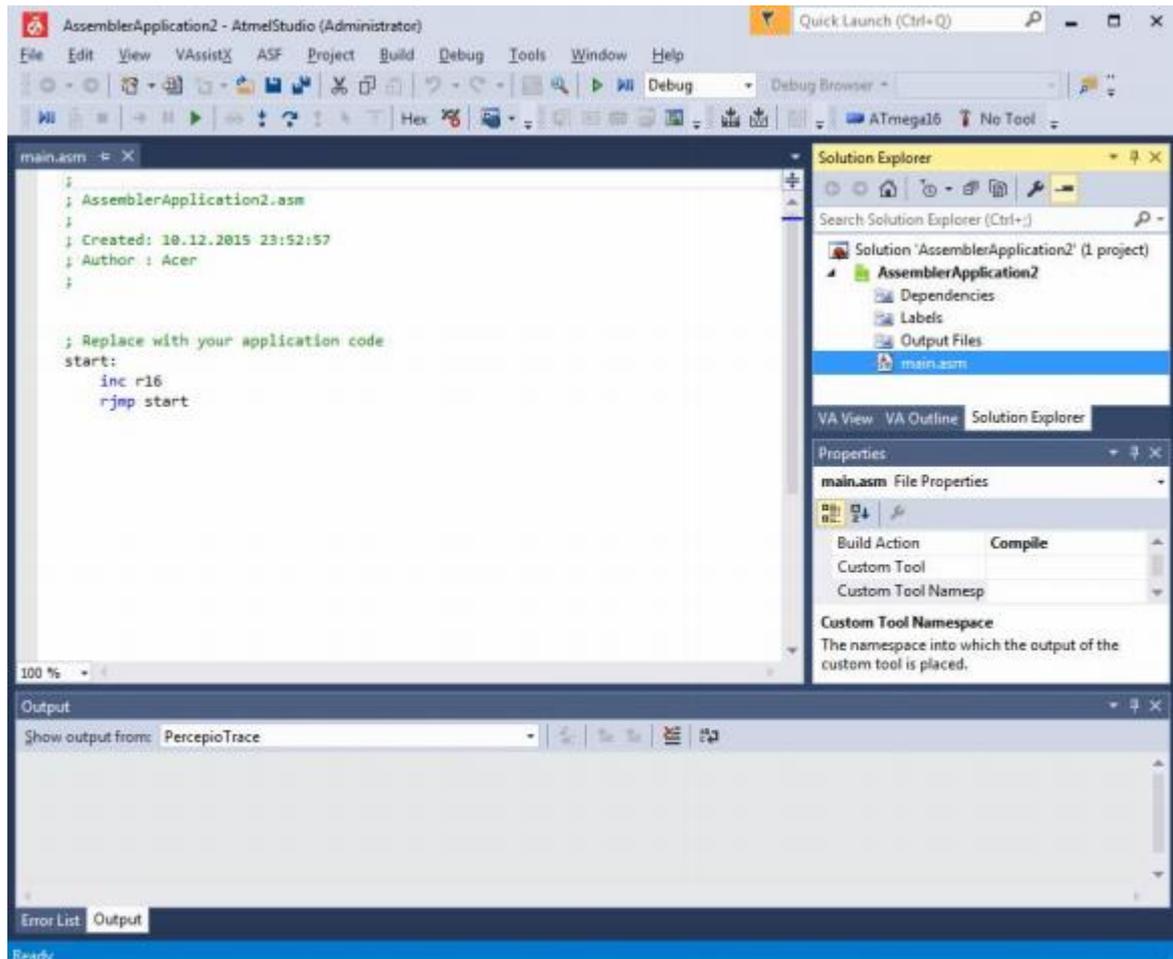


Рисунок 1.3 – Новый проект Atmel Studio

В правой части проекта следует отметить окно Solution Explorer, которое содержит список файлов проекта, и где возможно подключать внешние библиотеки функций, определений и т. п. Когда текст программы готов, можно начинать сборка проекта (Build), состоящего из двух этапов: – компиляция (Compiling) – процесс перевода текста программы, написанной языком программирования, в объектный модуль, который содержит машинные команды процессора; - компоновка (Linking) - сборка выполняемого модуля с одного или нескольких объектных модулей.

Собирание проекта запускается нажатием клавиши F7 или кнопки Build solution на панели инструментов. Отчет о процессе сбора выводится в нижнем окне Output. В данном случае, после успешной сборки появится сообщение:

Build succeeded.

===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====

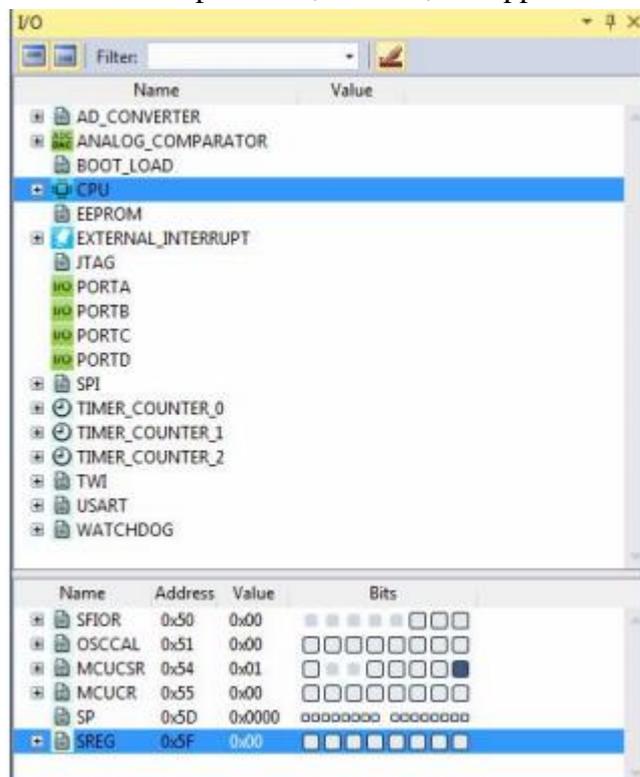


Рисунок 1.4 – Окно I/O

Отображение окна с содержимым регистров R0-R31: меню Debug >> Windows > Registers (Alt + 5).

В том же меню можно выбрать окно Processor status (рис.1.5), где показано текущее состояние регистров процессора, его текущая тактовая частота и сколько времени прошло от запуска МК до последней точки останова (Stop).

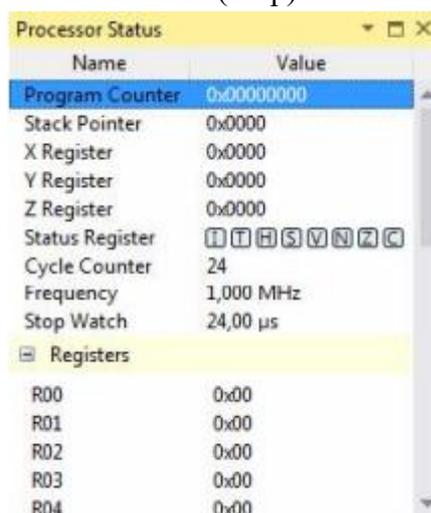


Рисунок 1.5 – Окно Processor Status

Также доступно окно Memory, где отображаются ячейки выбранного типа памяти с возможностью изменения их содержимого вручную.



Окно дизасемблера (Alt + 8) показывает обратную трансляцию скомпилированных программы из машинных кодов в Ассемблер.

Таким образом можно оценить работу компилятора при написании программы на языке C/C++.

Далее пошаговое выполнение каждой команды происходит при повторном нажатии клавиши F11. Выполнение первой команды «inc r16» увеличит значение регистра R16 на единицу. Проверить результат ее работы можно в окнах Registers или Processor status. Следующая команда «jmp start» осуществит возврат на метку start. При этом изменится значение регистра PC (окно Processor status).

### **3 Контрольные вопросы**

- 3.1. Назовите основные характеристики МК AVR.
- 3.2. Какое практическое значение достоинств Гарвардской RISC-архитектуры в МК AVR?
- 3.3. Назовите набор периферийных устройств МК семейства Mega.
- 3.4. Как построена система памяти МК семейства Mega?
- 3.5. Опишите регистры общего назначения.
- 3.6. Опишите регистры ввода/вывода.
- 3.7. Опишите биты регистра состояния SREG.
- 3.8. Назовите основные возможности отладки в среде Atmel Studio.

#### **4.4.2 Типовые задания для оценки освоения МДК.04.02 Разработка программного обеспечения для встраиваемых систем**

##### **Примерный перечень вопросов для формирования заданий текущего контроля:**

1. Стратегии разработки программных средств и систем.
2. Система реального времени. Принципиальное отличие обычных информационных систем от систем реального времени (СРВ). Система мягкого реального времени. Система жесткого реального времени.
3. Информационно-управляющая система (ИУС). Отличие СРВ от ИУС.
4. Варианты построения ИУС. Состав ИУС.
5. Встроенная (или встраиваемая) система (ВС). Категории ВС. Примеры встраиваемых систем.
6. Распределенная встроенная система.
7. Элементная база микропроцессорной техники для встраиваемых применений. Процессор. Классификация процессоров. Микропроцессор и микроконтроллер. Классификация микроконтроллеров.
8. Модульный принцип организации процессора ВВС. Типовая структура процессора для встраиваемых систем. Процессорное ядро.
9. Организация прерываний в управляющих процессорах. Модули памяти. Порты ввода-вывода.
10. Сетевые интерфейсы ВС.



11. Особенности организации и использования ВВС. Влияние особенностей аппаратного обеспечения ВВС на организацию ПО.
12. Особенности контроллеров. Анализ блоков микроконтроллеров с точки зрения программирования (центральный процессор, система контроля питания, подключение внешней памяти и т.д.).
13. Варианты организации ПО ВВС.
14. Стиль программирования, модель вычислений, платформа. Классификация стилей программирования. Стиль, метод, методология, методика.
15. Разработка встроенных систем с использованием модели вычислений.
16. Особенности реализации аппаратно-зависимого ПО.
17. Данные, поток данных, информация, процесс. Поточковая модель вычислений. Диаграмма потоков управления. Применение потоковых моделей на практике. Проблемы реализации потоковых моделей.
18. Реализация потоковой модели в Операционной системе реального времени (ОС РВ).
19. Обобщенная модель ОС РВ. Классификация ОС РВ. Основополагающие компоненты ОС РВ.
20. Компиляторы языков высокого уровня. Компилятор и транслятор. Язык программирования Си. Коммерческие компиляторы.

#### **Наименование проверяемой компетенции**

**ОК01-ОК09, ПК 4.2. Проектировать и программировать встраиваемые системы и интерфейсы оборудования с использованием языков программирования**  
*Открытые вопросы*

1. **Опишите основные этапы разработки программного обеспечения для встраиваемых систем.**
2. **Какие типы памяти используются в встраиваемых системах? Опишите их назначение.**
3. **Что такое прерывания в встраиваемых системах? Как они используются?**
4. **Назовите несколько примеров встраиваемых систем, с которыми вы сталкивались в повседневной жизни.**
5. **Какие языки программирования чаще всего используются для разработки ПО для встраиваемых систем? Почему?**

**Задание 1: Сопоставьте этап разработки ПО с его описанием**

<b>Этап разработки</b>	<b>Описание</b>
1. Анализ требований	a) Написание кода на выбранном языке программирования
2. Проектирование	b) Определение функциональных и нефункциональных требований к системе
3. Реализация	c) Создание архитектуры системы и выбор компонентов
4. Тестирование	d) Проверка корректности работы программы и устранение ошибок
5. Внедрение	e) Установка и настройка программы на целевом устройстве

**Задание 2: Сопоставьте инструмент разработки с его назначением**

<b>Инструмент</b>	<b>Назначение</b>
1. Компилятор	а) Отладка программы, поиск и исправление ошибок
2. Отладчик	б) Преобразование исходного кода в машинный код
3. Симулятор	в) Эмуляция работы программы на целевом устройстве
4. Профилировщик	г) Анализ производительности программы и оптимизация
5. Система контроля версий	д) Анализ производительности программы и оптимизация е) Управление изменениями в исходном коде и совместная работа

**Задание 3: Сопоставьте язык программирования с его особенностью**

<b>Язык программирования</b>	<b>Особенность</b>
1. C	а) Высокоуровневый язык с поддержкой ООП, используется в сложных системах
2. C++	б) Низкоуровневый язык, подходит для работы с аппаратурой
3. Python	в) Интерпретируемый язык, используется для прототипирования и скриптов
4. Assembler	г) Язык низкого уровня, используется для оптимизации критических участков
5. Rust	д) Язык низкого уровня, используется для оптимизации критических участков е) Современный язык с акцентом на безопасность и производительность

**Задание 4: Сопоставьте метод тестирования с его описанием**

<b>Метод тестирования</b>	<b>Описание</b>
1. Модульное тестирование	а) Проверка взаимодействия между компонентами системы
2. Интеграционное тестирование	б) Проверка корректности работы отдельных модулей программы
3. Системное тестирование	в) Проверка работы системы в целом
4. Регрессионное тестирование	г) Проверка, что новые изменения не нарушили существующую функциональность
5. Юнит-тестирование	д) Проверка, что новые изменения не нарушили существующую функциональность е) Проверка работы отдельных функций или методов

**Задание 5: Сопоставьте концепцию разработки с ее описанием**

<b>Концепция</b>	<b>Описание</b>
1. Реактивное программирование	а) Программа реагирует на внешние события в реальном времени
2. ООП (ООП)	б) Использование классов и объектов для моделирования системы
3. RTOS (Real-Time OS)	в) Операционная система, обеспечивающая выполнение задач в реальном времени
4. Агрегатное программирование	г) Программа разбивается на независимые модули, которые взаимодействуют через интерфейсы
5. Конечный автомат	д) Программа разбивается на независимые модули, которые взаимодействуют через интерфейсы е) Модель поведения системы, основанная на состояниях и переходах

**Задание 6: Сопоставьте тип памяти с его использованием в разработке ПО**

Тип памяти	Использование
1. ОЗУ (RAM)	a) Хранение данных, которые должны сохраняться после выключения устройства
2. ПЗУ (ROM)	b) Временное хранение данных во время выполнения программы
3. Flash-память	c) Хранение программы (прошивки) и данных, которые редко изменяются
4. EEPROM	d) Хранение настроек и данных, которые могут изменяться
5. Кэш-память	e) Ускорение доступа к часто используемым данным

**Задание 1: Разработка алгоритма для управления светодиодом**

- **Условие:**  
Напишите алгоритм и программный код для микроконтроллера, который управляет светодиодом (включение/выключение с интервалом 1 секунда).
- **Алгоритм:**
  1. Настройка порта ввода/вывода (GPIO) на выход.
  2. Бесконечный цикл:
    - Установить высокий уровень на выходе (включить светодиод).
    - Задержка 1 секунда.
    - Установить низкий уровень на выходе (выключить светодиод).
    - Задержка 1 секунда.
- **Пример кода (на C для AVR):**

```
c
Сору
#include <avr/io.h>
#include <util/delay.h>

int main(void) {
    DDRB |= (1 << PB0); // Настройка PB0 как выход
    while (1) {
        PORTB |= (1 << PB0); // Включить светодиод
        _delay_ms(1000); // Задержка 1 секунда
        PORTB &= ~(1 << PB0); // Выключить светодиод
        _delay_ms(1000); // Задержка 1 секунда
    }
}
```

**Задание 2: Управление двигателем с помощью ШИМ (PWM)**

- **Условие:**  
Разработайте алгоритм и программный код для управления скоростью двигателя с использованием ШИМ (PWM).
- **Алгоритм:**
  1. Настройка таймера для генерации ШИМ-сигнала.
  2. Установка коэффициента заполнения (duty cycle) для регулировки скорости.
  3. Бесконечный цикл для поддержания работы.

- **Пример кода (на C для AVR):**

```
c
Copy

#include <avr/io.h>

int main(void) {
    DDRD |= (1 << PD6); // Настройка PD6 как выход (PWM)
    TCCR0A |= (1 << COM0A1) | (1 << WGM01) | (1 << WGM00); // Режим Fast PWM
    TCCR0B |= (1 << CS01); // Предделитель таймера (CLK/8)
    OCR0A = 128; // Коэффициент заполнения 50%

    while (1) {
        // Основной цикл
    }
}
```

### Задание 3: Обработка прерываний от кнопки

- **Условие:**  
Напишите алгоритм и программный код для обработки прерывания от кнопки. При нажатии на кнопку состояние светодиода должно изменяться.
- **Алгоритм:**
  1. Настройка порта для кнопки (вход) и светодиода (выход).
  2. Настройка прерывания по фронту (например, по нажатию кнопки).
  3. В обработчике прерывания изменить состояние светодиода.
- **Пример кода (на C для AVR):**

```
c
Copy

#include <avr/io.h>
#include <avr/interrupt.h>

ISR(INT0_vect) {
    PORTB ^= (1 << PB0); // Инвертировать состояние светодиода
}

int main(void) {
    DDRB |= (1 << PB0); // Настройка PB0 как выход
    DDRD &= ~(1 << PD2); // Настройка PD2 как вход (кнопка)
    EICRA |= (1 << ISC01); // Прерывание по спаду
    EIMSK |= (1 << INT0); // Разрешить прерывание INT0
    sei(); // Глобально разрешить прерывания

    while (1) {
        // Основной цикл
    }
}
```

#### Задание 4: Мигание светодиодом

- **Условие:**  
Напишите программу для микроконтроллера, которая мигает светодиодом с интервалом 500 мс.
- **Решение:**  
Используйте GPIO для управления светодиодом и таймер для задержки.
- **Пример кода (AVR):**

```
c
Copy

#include <avr/io.h>
#include <util/delay.h>

int main(void) {
    DDRB |= (1 << PB0); // Настройка PB0 как выход
    while (1) {
        PORTB ^= (1 << PB0); // Инвертировать состояние светодиода
        _delay_ms(500); // Задержка 500 мс
    }
}
```

#### Задание 5: Управление кнопкой

- **Условие:**  
Напишите программу, которая включает светодиод при нажатии кнопки и выключает его при отпускании.
- **Решение:**  
Используйте GPIO для чтения состояния кнопки и управления светодиодом.
- **Пример кода (AVR):**

```
c
Copy

#include <avr/io.h>

int main(void) {
    DDRB |= (1 << PB0); // Настройка PB0 как выход (светодиод)
    DDRD &= ~(1 << PD2); // Настройка PD2 как вход (кнопка)
    while (1) {
        if (PIND & (1 << PD2)) { // Если кнопка нажата
            PORTB |= (1 << PB0); // Включить светодиод
        } else {
            PORTB &= ~(1 << PB0); // Выключить светодиод
        }
    }
}
```

#### Задание 6: Генерация ШИМ-сигнала

- **Условие:**  
Напишите программу для генерации ШИМ-сигнала с коэффициентом заполнения 75% на выводе микроконтроллера.

- **Решение:**  
Используйте таймер и режим Fast PWM.
- **Пример кода (AVR):**

```
c
Copy

#include <avr/io.h>

int main(void) {
    DDRD |= (1 << PD6); // Настройка PD6 как выход (PWM)
    TCCR0A |= (1 << COM0A1) | (1 << WGM01) | (1 << WGM00); // Режим Fast PWM
    TCCR0B |= (1 << CS01); // Предделитель таймера (CLK/8)
    OCR0A = 192; // Коэффициент заполнения 75% (192/256)

    while (1) {
        // Основной цикл
    }
}
```

### Задание 7: Обработка прерываний

- **Условие:**  
Напишите программу, которая изменяет состояние светодиода при каждом нажатии кнопки, используя прерывания.
- **Решение:**  
Настройте прерывание по фронту сигнала от кнопки.
- **Пример кода (AVR):**

```
c
Copy

#include <avr/io.h>
#include <avr/interrupt.h>

ISR(INT0_vect) {
    PORTB ^= (1 << PB0); // Инvertировать состояние светодиода
}

int main(void) {
    DDRB |= (1 << PB0); // Настройка PB0 как выход (светодиод)
    DDRD &= ~(1 << PD2); // Настройка PD2 как вход (кнопка)
    EICRA |= (1 << ISC01); // Прерывание по спаду
    EIMSK |= (1 << INT0); // Разрешить прерывание INT0
    sei(); // Глобально разрешить прерывания

    while (1) {
        // Основной цикл
    }
}
```

### Задание 8: Управление сервоприводом

- **Условие:**  
Напишите программу для управления сервоприводом с помощью ШИМ. Угол поворота сервопривода должен изменяться от 0° до 180°.
- **Решение:**  
Используйте таймер для генерации ШИМ-сигнала с длительностью импульса от 1 мс (0°) до 2 мс (180°).
- **Пример кода (AVR):**

```
с
Copy
#include <avr/io.h>
#include <util/delay.h>

void setServoAngle(uint8_t angle) {
    uint16_t pulseWidth = 1000 + (angle * 1000 / 180); // Расчет длительности импульса
    OCR1A = pulseWidth; // Установка значения для таймера
}

int main(void) {
    DDRB |= (1 << PB1); // Настройка PB1 как выход (PWM)
    TCCR1A |= (1 << COM1A1) | (1 << WGM11); // Режим Fast PWM, 10-бит
    TCCR1B |= (1 << WGM12) | (1 << WGM13) | (1 << CS11); // Предделитель CLK/8
    ICR1 = 19999; // Период 20 мс (50 Гц)

    while (1) {
        for (uint8_t angle = 0; angle <= 180; angle += 10) {
            setServoAngle(angle); // Установка угла
            _delay_ms(500); // Задержка
        }
    }
}
```

### Задание 9: Работа с АЦП (аналого-цифровым преобразователем)

- **Условие:**  
Напишите программу, которая считывает значение с потенциометра через АЦП и выводит его на светодиодную шкалу (например, 8 светодиодов).
- **Решение:**  
Используйте АЦП для чтения аналогового сигнала и GPIO для управления светодиодами.
- **Пример кода (AVR):**

```
с
Copy
#include <avr/io.h>
#include <util/delay.h>

void initADC() {
    ADMUX |= (1 << REFS0); // опорное напряжение AVCC
```

```
ADCSRA |= (1 << ADEN) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // Включение АЦП,
пределитель 128
}

uint16_t readADC(uint8_t channel) {
    ADMUX = (ADMUX & 0xF0) | (channel & 0x0F); // Выбор канала
    ADCSRA |= (1 << ADSC); // Запуск преобразования
    while (ADCSRA & (1 << ADSC)); // Ожидание завершения
    return ADC; // Возврат результата
}

int main(void) {
    DDRB = 0xFF; // Настройка порта В как выход (светодиоды)
    initADC(); // Инициализация АЦП

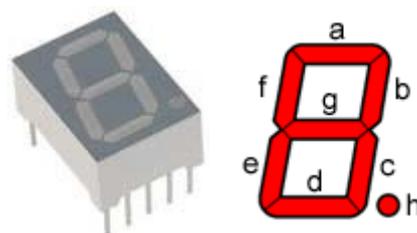
    while (1) {
        uint16_t adcValue = readADC(0); // Чтение значения с канала 0
        PORTB = adcValue >> 2; // Вывод на светодиоды (масштабирование)
        _delay_ms(100); // Задержка
    }
}
```

### Пример задания на практическое занятие

Подключение к микроконтроллеру семисегментного светодиодного индикатора.

**Семисегментные индикаторы** широко применяются в цифровой технике: в бытовых приборах, измерительной технике, в промышленных устройствах. По сравнению с жидкокристаллическими индикаторами светодиодные имеют свои преимущества, это контрастность отображения информации, малое потребление энергии. Семисегментный индикатор представляет собой матрицу из семи светодиодов, размещенных таким образом, чтобы зажигая их в разных сочетаниях, можно было бы отобразить любую десятичную цифру, а также специальные символы. Кроме этого индикатор дополняется еще одним сегментом, который предназначен для отображения десятичной точки.

На рисунке 1 изображен внешний вид индикатора. Принято каждый сегмент индикатора обозначать латинской буквой: a, b, c, d, e, f, g. Точка обозначается буквой h.



По схеме включения семисегментные индикаторы подразделяются на индикаторы с общим катодом и с общим анодом. Схемы включения приведены на рисунке 2.

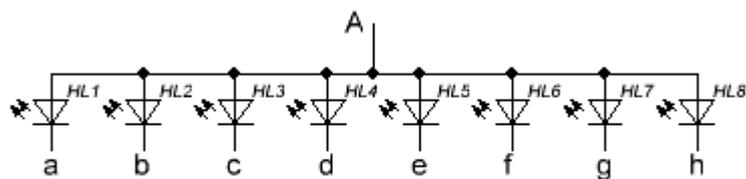


Схема индикатора с общим анодом

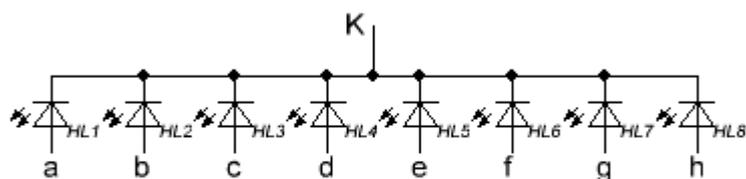
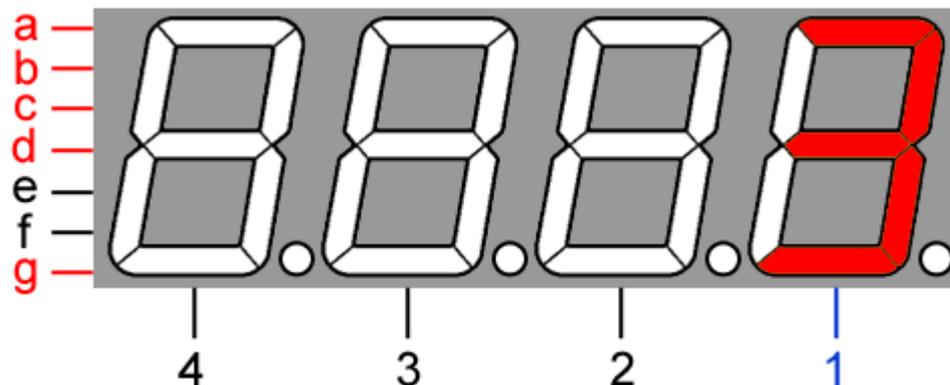


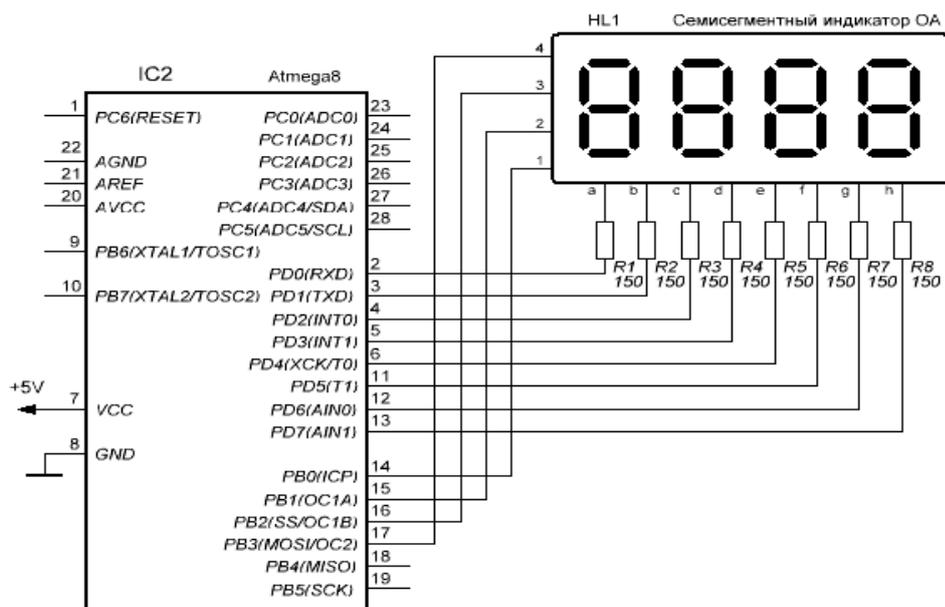
Схема индикатора с общим катодом

Подключить один семисегментный индикатор и управлять им с помощью микроконтроллера процедура несложная. Для этого достаточно сегменты индикатора подключить к порту микроконтроллера через токоограничительные резисторы по 150 Ом. Общий вывод подключить к линии другого порта микроконтроллера. В зависимости от того какую цифру надо вывести, в порт выводим двоичный код этой цифры, ссылаясь на тип подключенного индикатора (с общим анодом или катодом) на общий провод подаем плюс или минус. Для удобства можно сделать таблицу кодов для индикатора. Если подключение такое: PD7-h, PD6-g, PD5-f, PD4-e, PD3-d, PD2-c, PD1-b, PD0-a, то для отображения цифры 1 в порт D нужно вывести такой двоичный код: **0b00000110**.

Для отображения цифровых данных одного семисегментного индикатора обычно недостаточно. В таких случаях к микроконтроллеру подключают сразу несколько индикаторов. Однако, из-за отсутствия достаточного количества выводов у микроконтроллера применяют специальные методы. Один из таких методов это динамическая индикация. Режим динамической индикации применяют для построения **многоразрядных индикаторов**. При таком режиме разряды индикатора работают не одновременно, а по очереди. Переключение разрядов происходит с большой скоростью (**50 Гц**), из-за этого человеческий глаз не замечает, что индикаторы работают по очереди. Так как у светодиодов очень малая инерционность, сменяющиеся разряды сливаются в одно изображение. В этом режиме в каждый момент времени работает только один разряд, включаются по очереди начиная с первого заканчивая последним, затем все начинается сначала.



Сделаем простой секундомер. Отсчет секунд будет производиться на четырехразрядном индикаторе (с общим анодом) от 0 до 9999. В нашей программе используем процедуру прерывания по таймеру, т.е. смена разряда индикатора будет происходить каждый раз когда таймер досчитает до конца (до 255). Используем восьмиразрядный таймер/счетчик **T2**, он будет работать в нормальном режиме. Но обычно для реализации динамической индикации используют режим **CTC** (сброс при совпадении), это режим, при котором частота возникновения прерываний по совпадению значений счетчика таймера и регистра **OCR2** определяется содержимым **OCR2** и предделителем тактовой частоты таймера. При таком режиме работы таймера можно легко изменять частоту обновления разрядов, записывая в регистр сравнения **OCR2** необходимое значение, предварительно рассчитанное. Частоту обновления разрядов делают обычно **50Hz** или больше, так как у нас 4 разряда, частота обновления будет равна **200Hz**. Подсчитаем частоту обновления для нашего примера: тактовая частота равна **8MHz**, предделитель сделаем на 8. На вход таймера будут поступать импульсы частотой **1MHz**. Тогда таймер будет увеличивать значение каждые 1 микросекунду, переполняться он будет каждые  $255 * 0,000001 = 255 \text{ мкс}$ . Частота обновления будет равна  $1/255 \text{ мкс} = 3921 \text{ Hz}$ .



Каждый раз по прерыванию мы должны в обработчике сначала погасить все индикаторы, затем выбрать из заранее подготовленного массива выводимых символов очередной

символ, вывести его в порт **D**, а потом установить лог. 1 на линию порта **B**, которая соответствует следующему индикатору. Таким образом мы сможем обновлять поочередно информацию на индикаторах, что создаст эффект их непрерывного свечения. Выводить двоичный код в порт **D** будем согласно таблице, приведенной ниже.

Цифра	PD7 (H)	PD6 (G)	PD5 (F)	PD4 (E)	PD3 (D)	PD2 (C)	PD1 (B)	PD0 (A)	HEX
0	0	0	1	1	1	1	1	1	0x3F
1	0	0	0	0	0	1	1	0	0x06
2	0	1	0	1	1	0	1	1	0x5B
3	0	1	0	0	1	1	1	1	0x4F
4	0	1	1	0	0	1	1	0	0x66
5	0	1	1	0	1	1	0	1	0x6D
6	0	1	1	1	1	1	0	1	0x7D
7	0	0	0	0	0	1	1	1	0x07
8	0	1	1	1	1	1	1	1	0x7F
9	0	1	1	0	1	1	1	1	0x6F
h	1	0	0	0	0	0	0	0	0x80

Массив с кодами цифр получится такой:

```
1 unsigned char SEGMENTE[ ] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D,  
0x07, 0x7F, 0x6F};
```

В обработчике прерываний мы используем оператор **switch**, этот оператор позволяет заменить сложную функцию из операторов **if**. В общем виде он выглядит так:

```
1 switch ( выражение )  
  {  
2 case значение1:  
  .....  
3 break;  
  case значение2:  
4 .....  
  break;
```



```
5     .....
6     default:
7     .....
8     }
```

Данный оператор производит выбор по выражению, обычно это число. Если выражение присутствует в значении **case**, то выполняются команды после **case** до **break**, иначе выполняется код после **default**.

Ниже приведен полный текст программы:

```
1 // Подключение семисегментных индикаторов к AVR. Динамическая индикация
2 #include <avr/io.h>
3 #include <avr/interrupt.h>
4 #include <util/delay.h>
5 //-----0-----1-----2-----3-----4-----5-----6-----7-
6 ----8-----9
7 unsigned char SEGMENTE[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D,
8 0x07, 0x7F, 0x6F};
9
10 unsigned char segcounter = 0;
11 unsigned int display = 0;
12
13 // Обработчик прерывания по переполнению таймера 2
14 ISR(TIMER2_OVF_vect)
15 {
16 PORTD = 0xFF; // Гасим все разряды
17 PORTB = (1 << segcounter); // Выбираем следующий разряд
18
19 switch(segcounter)
20 {
21 case 0:
22 PORTD = ~(SEGMENTE[display % 10000 / 1000]); // Раскладываем число на
23 разряды
24 break;
25 case 1:
26 PORTD = ~(SEGMENTE[display % 1000 / 100]);
27 break;
28 case 2:
29 PORTD = ~(SEGMENTE[display % 100 / 10]);
30 break;
31 case 3:
32 PORTD = ~(SEGMENTE[display % 10]);
33 break;
34 }
35 if(segcounter++ > 2) segcounter = 0;
36 }
37
38 // Главная функция
39 int main(void)
40 {
41 DDRB = 0xFF; // Порт B - выход
42 PORTB = 0x00;
43 DDRD = 0xFF; // Порт D - выход
44 PORTD = 0x00;
45 }
```



```
40 TCCR2 |= (1 << CS21); // Предделитель на 8
41 TIMSK |= (1 << TOIE2); // Разрешение прерывания по таймеру 2
42
43 sei(); // Глобально разрешаем прерывания
44
45 while(1)
46 {
47   display++; // Увеличиваем счет от 0 до 9999
48   if(display > 9999) display = 0;
49   _delay_ms(10); // Задержка
50 }
51 }
```

#### 4.5. Оценка по учебной и (или) производственной (по профилю специальности) практике

##### 4.5.1 Виды работ учебной практики и проверяемые результаты обучения по профессиональному модулю:

Таблица 6

Виды работ	Проверяемые результаты (ПК, ОК, профессиональный опыт, умения)
1. Установка программного обеспечения. Конфигурирование микроконтроллера, создании проекта, компиляции, прошивка. 2. Работа с регистрами микроконтроллера. Библиотеки для разработчика. 3. Система тактирования микроконтроллера. 4. Порты ввода-вывода микроконтроллера. 5. Управление портами ввода-вывода через регистры. 6. Управление портами ввода-вывода через функции библиотеки. 7. Типы данных языка C для микроконтроллера. 8. Конвертирование проекта для микроконтроллера на языке C в проект C++. 9. Обработка входных дискретных сигналов. Устранение дребезга контактов, борьба с импульсными помехами. 10. Разработка и использование классов в C++. Создание класса обработки дискретных сигналов. 11. Создание и использование библиотек для микроконтроллера. 12. Параллельные процессы. Выполнение задач в фоновом режиме при помощи прерывания от таймера.	- формализация и алгоритмизация поставленных задач; - написание программного кода с использованием языков программирования, определения и манипулирования данными; - оформление программного кода в соответствии с установленными требованиями; - проверка и отладка программного кода; - разработка процедур проверки работоспособности и измерения характеристик программного обеспечения; - разработка тестовых наборов данных; - проверка работоспособности программного обеспечения; - рефакторинг и оптимизация программного кода; - исправление дефектов, зафиксированных в базе данных дефектов. - составлять программы на языке программирования для встраиваемых систем; - применять стандартные алгоритмы и конструкции языка программирования; - выбирать микроконтроллер для конкретной задачи встраиваемой системы; - выполнять требования технического



13. Таймеры микроконтроллера в режиме счетчиков. Генерация циклических прерываний от таймеров.

14. Разработка программ, состоящих из нескольких исходных файлов. Определение и объявление переменных, область видимости. Режимы компиляции.

15. Система прерываний микроконтроллера. Организация и управление прерываниями.

16. Установка конфигурации таймеров с помощью библиотек. Логика работы прерывания таймера.

17. Интерфейс UART в микроконтроллере. Использование прерывания UART.

18. Работа с UART через библиотеку. Инициализация интерфейса и передача данных в блокирующем режиме. Отладка программ с помощью UART. Функция `sprintf`.

19. Работа с UART через библиотеку. Прием данных в блокирующем режиме.

20. Работа с UART через библиотеку с использованием прерываний.

21. Организация коротких временных задержек.

22. АЦП микроконтроллера. Общие сведения, режимы. Установка конфигурации через регистры.

23. Работа с АЦП через регистры. Основные режимы преобразования.

24. Работа с АЦП в различных режимах. Запуск от таймера, чтение результата с использованием прерываний.

25. Работа АЦП в режиме оконного компаратора. Внутренние датчик температуры и ИОН. Основные электрические и метрологические характеристики АЦП.

26. Работа с АЦП через функции библиотеки.

27. Прямой доступ к памяти в микроконтроллере. Контроллер DMA.

задания по программированию встраиваемых систем;

- создавать и отлаживать программы реального времени средствами программной эмуляции и на аппаратных макетах;
- находить ошибки в программном коде для встраиваемой системы и оценивать степень их критичности;
- производить тестирование и отладку встраиваемых систем на базе микроконтроллеров;
- выявлять причины неисправностей периферийных модулей встраиваемых систем.

ОК.01-ОК.09, ПК.4.1



**4.5.2. Виды работ производственной (по профилю специальности) практики и проверяемые результаты обучения по профессиональному модулю:**

Виды работ	Проверяемые результаты (ПК, ОК, профессиональный опыт, умения)
<p>1. Установка инструментальной среды разработки программного обеспечения для встраиваемых микроконтроллерных систем.</p> <p>2. Настройка интерфейса пользователя и параметров среды. Установка и настройка компилятора.</p> <p>3. Анализ технического задания на разработку программного обеспечения.</p> <p>4. Разработка алгоритма программы для встраиваемой микроконтроллерной системы.</p> <p>5. Написание программы на специализированном языке для встраиваемой микроконтроллерной системы.</p> <p>6. Подбор стандартных библиотек для реализации проекта.</p> <p>7. Программирование встраиваемой микроконтроллерной системы.</p> <p>8. Проведение отладки программного обеспечения микропроцессорных систем с помощью аппаратно-программных средств.</p> <p>9. Проверка функциональности программного обеспечения.</p> <p>10. Составление отчетной программной документации.</p>	<ul style="list-style-type: none"><li>- формализация и алгоритмизация поставленных задач;</li><li>- написание программного кода с использованием языков программирования, определения и манипулирования данными;</li><li>- оформление программного кода в соответствии с установленными требованиями;</li><li>- проверка и отладка программного кода;</li><li>- разработка процедур проверки работоспособности и измерения характеристик программного обеспечения;</li><li>- разработка тестовых наборов данных;</li><li>- проверка работоспособности программного обеспечения;</li><li>- рефакторинг и оптимизация программного кода;</li><li>- исправление дефектов, зафиксированных в базе данных дефектов.</li><li>- составлять программы на языке программирования для встраиваемых систем;</li><li>- применять стандартные алгоритмы и конструкции языка программирования;</li><li>- выбирать микроконтроллер для конкретной задачи встраиваемой системы;</li><li>- выполнять требования технического задания по программированию встраиваемых систем;</li><li>- создавать и отлаживать программы реального времени средствами программной эмуляции и на аппаратных макетах;</li><li>- находить ошибки в программном коде для встраиваемой системы и оценивать степень их критичности;</li><li>- производить тестирование и отладку встраиваемых систем на базе микроконтроллеров;</li><li>- выявлять причины неисправностей</li></ul>



периферийных модулей встраиваемых систем.  
ОК.01-ОК.09, ПК.4.2

**5. Фонд оценочных средств для аттестации по модулю (экзамен (квалификационный)):**

**ПАСПОРТ**

**Назначение:**

ФОС предназначен для контроля и оценки результатов освоения профессионального модуля

**ПАКЕТ ЭКЗАМЕНАТОРА**

**Примерный перечень вопросов для формирования заданий письменной экзаменационной работы.**

**5.1 Вопросы экзамену квалификационному**

- 1 Что такое микроконтроллеры, микропроцессоры и сигнальные процессоры
- 2 Области применения микроконтроллеров
- 3 Целочисленные двоичные коды
- 4 Запись текстов двоичным кодом
- 5 Запись десятичных чисел двоичным кодом
- 6 Представление чисел в двоичном коде с плавающей запятой
- 7 Масочные ПЗУ, ППЗУ, РПЗУ
- 8 EEPROM и flash память
- 9 Внутреннее устройство статического ОЗУ
- 10 Команды микропроцессора
- 11 Системная шина микропроцессора
- 12 Принципы построения параллельного порта. Подключение внешних устройств к микропроцессору
- 13 Принципы построения последовательных портов. Виды последовательных портов
- 14 Принципы построения схем таймеров микропроцессоров
- 15 Архитектура микроконтроллеров MCS-51
- 16 Система команд микроконтроллеров MCS-51
- 17 Виды адресации
- 18 Инструкции микроконтроллеров MCS-51
- 19 Особенности построения параллельных портов микроконтроллеров MCS-51
- 20 Особенности построения памяти микроконтроллеров семейства MCS-51
- 21 Внутренние таймеры микроконтроллера, особенности их применения
- 22 Устройство и особенности применения последовательного порта микроконтроллеров семейства MCS-51
- 23 Особенности проектирования схем на микроконтроллерах
- 24 Особенности проектирования системы питания для устройств на микроконтроллерах
- 25 Языки программирования для микроконтроллеров



- 26 Применение подпрограмм при программировании. Понятие подпрограммы процедуры и подпрограммы функции
- 27 Написание программ для микропроцессоров. Понятие программы-монитора и операционной системы реального времени
- 28 Понятие структурного программирования. Применение комментариев
- 29 Понятие многофайлового и многомодульного программирования
- 30 Стратегии разработки программных средств и систем.
- 31 Система реального времени. Принципиальное отличие обычных информационных систем от систем реального времени (СРВ). Система мягкого реального времени. Система жесткого реального времени.
- 32 Информационно-управляющая система (ИУС). Отличие СРВ от ИУС.
- 33 Варианты построения ИУС. Состав ИУС.
- 34 Встроенная (или встраиваемая) система (ВС). Категории ВС. Примеры встраиваемых систем.
- 35 Распределенная встроенная система.
- 36 Элементная база микропроцессорной техники для встраиваемых применений. Процессор. Классификация процессоров. Микропроцессор и микроконтроллер. Классификация микроконтроллеров.
- 37 Модульный принцип организации процессора ВВС. Типовая структура процессора для встраиваемых систем. Процессорное ядро.
- 38 Организация прерываний в управляющих процессорах. Модули памяти. Порты ввода-вывода.
- 39 Сетевые интерфейсы ВС.
- 40 Особенности организации и использования ВВС. Влияние особенностей аппаратного обеспечения ВВС на организацию ПО.
- 41 Особенности контроллеров. Анализ блоков микроконтроллеров с точки зрения программирования (центральный процессор, система контроля питания, подключение внешней памяти и т.д.).
- 42 Варианты организации ПО ВВС.
- 43 Стиль программирования, модель вычислений, платформа. Классификация стилей программирования. Стиль, метод, методология, методика.
- 44 Разработка встроенных систем с использованием модели вычислений.
- 45 Особенности реализации аппаратно-зависимого ПО.
- 46 Данные, поток данных, информация, процесс. Поточковая модель вычислений. Диаграмма потоков управления. Применение потоковых моделей на практике. Проблемы реализации потоковых моделей.
- 47 Реализация потоковой модели в Операционной системе реального времени (ОС РВ).
- 48 Обобщенная модель ОС РВ. Классификация ОС РВ. Основополагающие компоненты ОС РВ.

49 Компиляторы языков высокого уровня. Компилятор и транслятор. Язык программирования Си. Коммерческие компиляторы.

## 5.2 Задание для контроля практических навыков

### Пример практического задания к экзаменационному билету.

Разработка приложения контроллера домашней погодной станции

**Краткая теория** Метеостанция предназначена для наблюдения за погодой, просмотром текущей температуры, влажности и атмосферного давления. Минимально необходимый набор функций домашней погодной станции: – измерение температуры окружающего воздуха; – измерение влажности окружающего воздуха; – измерение атмосферного давления; – формирование прогноза погоды на различные периоды на основе измеренных данных.

Порядок выполнения работы:

1. Для построения схемы, подключите контакты датчиков, громкоговорителя и LCD дисплея к выбранным контактам на отладочной плате.

2. В среде разработки Arduino IDE наберите программу, выполняющую следующие функции: – измерение температуры и влажности окружающего воздуха; – измерение атмосферного давления; исходя из полученных данных; – индикация полученного прогноза реализована с помощью LCD дисплея.



УСЛОВИЯ

К экзамену квалификационному допускаются обучающиеся, освоившие все составляющие профессионального модуля:

МДК.04.01 Микроконтроллеры и встраиваемые системы

МДК.04.02 Разработка программного обеспечения для встраиваемых систем

УП.04.01 Учебная практика

ПП.04.01 Производственная практика (по профилю специальности)



Место проведения экзамена квалификационного определяется расписанием занятий.

Для проведения экзамена квалификационного создается экзаменационная комиссия, состоящая из 3 человек. В состав экзаменационной комиссии входят: представитель работодателя, руководитель ОПОП СПО ППССЗ и ведущие преподаватели специальности.

Экзамен квалификационный проводится в виде демонстрационного экзамена в состав, которого входят три теоретических вопроса из курса МДК.04.01 Микроконтроллеры и встраиваемые системы и МДК.04.02 Разработка программного обеспечения для встраиваемых систем, а также практическое задание из перечня выполняемых работ на учебной и (или) производственной практике.

На выполнение практической части дается 2 часа.

Подготовку билета 45 минут.

На ответ отводится до 7 минут.

#### КРИТЕРИИ ОЦЕНКИ

---

##### 1) Ход выполнения задания

При проведении экзамена квалификационного знания и умения каждого слушателя оцениваются за:

- а) качество доклада (по существу рассматриваемых вопросов);
- б) правильность ответов на дополнительные вопросы;
- в) качество выполнения проектного задания.

Итоговая оценка за экзамен квалификационный определяется:

- «отлично» – если выполнение проектного задания оценено на «отлично», а остальные показатели по среднему баллу оценены не ниже «хорошо»;
- «хорошо» – если выполнение проектного задания оценено не ниже «хорошо», а остальные показатели по среднему баллу оценены не ниже «удовлетворительно»;
- «удовлетворительно» – если выполнение проектного задания оценено на «удовлетворительно», а остальные показатели по среднему баллу оценены не ниже «удовлетворительно»;
- «неудовлетворительно» – если не выполнены условия получения положительной оценки.

Обсуждение результатов экзамена квалификационного студента производится сразу после ответов на поставленные ему вопросы. Решение об оценке по каждому обучающемуся принимается комиссией на закрытом заседании большинством голосов.



В случае получения студентом положительной оценки за экзамен квалификационный модуль ПМ.04 Программирование встраиваемых систем с использованием интегрированных сред разработки **считается освоенным.**

Коды проверяемых компетенций или их сочетаний	Показатели оценки результата	Оценка (да / нет)
ОК1-ОК9	Полнота и правильность письменного ответа на вопросы экзаменационного билета	Да/нет
ПК 4.1-ПК 4.2	Применение в процессе выполнения проектного задания технической документации	Да/нет
ПК 4.1-ПК 4.2	Применение в процессе выполнения проектного задания интегрированных сред программирования.	Да/нет
ОК1-ОК9	Качество доклада (по существу вопросов экзаменационного билета)	Да/нет
ОК1-ОК9	Правильность ответов на дополнительные вопросы;	Да/нет
ПК 4.1-ПК 4.2	Качество выполнения проектного задания	Да/нет

## 6. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

**6.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:**

**Учебная аудитория, оборудованная:**

Учебная мебель, доска, мультимедийное презентационное оборудование.

Лаборатория «Микропроцессорной техники и встраиваемых устройств», оборудованная:

Учебная мебель

ПК-13

Мультимедийное презентационное оборудование,

Маршрутизатор

Доска

Лабораторные стенды по вычислительной технике

Установка учебная РТЦУЛ-10К

Образцы плат Arduino Mega, Uno, Nano



## 7. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организации выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список может быть дополнен новыми изданиями.

### 7.1. Основные печатные издания

1. Варламов, Д. О. Работа с основными периферийными модулями микроконтроллеров Atmega на примерах и заданиях : учебное пособие / Д.О. Варламов, С.М. Зуев, А.А. Тимошенко. — Москва : ИНФРА-М, 2021. — 186 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-109712-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1816758>
2. Водовозов, А. М. Микроконтроллеры для систем автоматики : учебное пособие / А. М. Водовозов. - 2-е изд., испр. и доп. - Москва ; Вологда : Инфра-Инженерия, 2022. - 168 с. - ISBN 978-5-9729-1071-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1903136>
3. Arduino® Полный учебный курс. От игры к инженерному проекту. 3 - е изд. : Учебное пособие / А.А. Салахова, О.А. Феоктистова, Н.А. Александрова, М.В. Храмова . — Москва : Лаборатория знаний, 2024. — 176 с. — ISBN 978-5-93208-670-4. — URL: <https://book.ru/book/953242>
4. Салахова, А. А. Конструируем роботов на Arduino®. Электронный домашний питомец : учебно-практическое пособие / А. А. Салахова. - 2-е изд. - Москва : Лаборатория знаний, 2022. - 68 с. - (РОБОФИШКИ). - ISBN 978-5-00101-968-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1906359>
5. Макаров, С. Л. Arduino Uno и Raspberry Pi 4: от схемотехники к интернету вещей : практическое руководство / С. Л. Макаров. – 2-е изд., перераб. и доп. - Москва : ДМК Пресс, 2024. - 242 с. – ISBN 978-5-93700-272-3. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2205061>
6. Программирование в Интернете вещей : учебное пособие / Л. Б. Филиппова, Р. А. Филиппов, А. С. Сазонова [и др.]. — Москва : Русайнс, 2024. — 174 с. — ISBN 978-5-466-04116-3. — URL: <https://book.ru/book/951112>

### 7.2. Дополнительные источники

1. Винницкий, Ю. А. Конструируем роботов на ScratchDuino®. Первые шаги : практическое руководство / Ю. А. Винницкий, К. Ю. Поляков. - 3-е изд. - Москва : Лаборатория знаний, 2020. - 119 с. - (РОБОФИШКИ). - ISBN 978-5-00101-901-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1203943>
2. Виноградов, В. М. Технологические процессы автоматизированных производств : учебник для студентов высших учебных заведений / В.М. Виноградов, А.А. Черепяхин, В.В. Клепиков. — Москва : КУРС : ИНФРА-М, 2023. — 272 с. — (Бакалавриат). - ISBN 978-5-906818-69-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1941738>
3. Бахвалова, С. А. Основы моделирования и проектирования радиотехнических устройств в Microwave Office : учебно-методическое пособие / С. А. Бахвалова, В. А. Романюк. - Москва : СОЛОН-ПРЕСС, 2024. - 152 с. - ISBN 978-5-91359-206-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2185095>



4. Трубочкина, Н. К. Моделирование 3D наносхемотехники : монография / Н. К. Трубочкина. - 4-е изд. - Москва : Лаборатория знаний, 2024. - 526 с. - ISBN 978-5-93208-756-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2178479>
5. Богатырев, А. В. Электронные системы мобильных машин : учебное пособие / А.В. Богатырев. — Москва : ИНФРА-М, 2022. — 224 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/13026. - ISBN 978-5-16-006638-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1844423>
6. Наноэлектроника. Состояние и перспективы развития : учебное пособие / А. Н. Игнатов. - 3-е изд., стер. - Москва : Флинта, 2022. - 360 с. - ISBN 978-5-9765-1619-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1985742>
7. Проектирование цифровых устройств : учебник / А.В. Кистрин, Б.В. Костров, М.Б. Никифоров, Д.И. Устюков. — Москва : КУРС : ИНФРА-М, 2022. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-59-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1495622>
8. Фортыгина, С. Н., ЛЕГО-конструирование и робототехника в ДОУ : практикум / С. Н. Фортыгина, М. Н. Забродина, А. Н. Корниенко, ; под общ. ред. С. Н. Фортыгиной. — Москва : Русайнс, 2025. — 118 с. — ISBN 978-5-466-08609-6. — URL: <https://book.ru/book/957465>
9. Дмитриев, В. Г. Радиоэлектронная борьба: функциональное поражение радиоэлектронных средств : монография / В. Г. Дмитриев. - Москва ; Вологда : Инфра-Инженерия, 2021. - 268 с. - ISBN 978-5-9729-0700-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1832020>

### 7.3 Интернет ресурсы

1. «РадиоЛоцман»: сайт. [Электронный ресурс]. URL: [www.rlocman.com.ru/indexs.htm](http://www.rlocman.com.ru/indexs.htm)
2. RadioRadar - электронный портал: Datasheets, service manuals, схемы, электроника, компоненты, САПР, CAD. [Электронный ресурс]. – URL: [https://www.radioradar.net/about\\_project/index.html/](https://www.radioradar.net/about_project/index.html/)
3. Паяльник: сайт. [Электронный ресурс]. – URL: <http://cxem.net>