



КГУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГУ»)

УТВЕРЖДАЮ

И.о.директора _____ ИЭЭ

Наименование института

_____ Р.Р. Гибадуллин

«_____» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДЭ.02.03.04. Программирование управляющих устройств систем
автоматизации

Направление подготовки _____ 11.03.04 Электроника и нанoeлектроника

Профиль _____ Программирование и электроника информационных систем

Квалификация _____ Бакалавр

г. Казань, 2025

Программу разработал(и):

Наименование кафедры	Должность, уч. степень, уч. звание	ФИО разработчика
Промышленная электроника	Доцент, к.п.н., доцент	Ахметвалеева Л.В.

Согласование	Наименование подразделения	Дата	№ протокола	Подпись
Одобрена	ПЭ	12.02.2025	№ 12	_____ Зав. каф., д.т.н., доц. Иванов Д.А.
Согласована	ПЭ	12.02.2025	№ 12	_____ Зав. каф., д.т.н., доц. Иванов Д.А.
Согласована	Учебно- методический совет института	18.02.2025	№ 6	_____ И.о. директора ИЭЭ, к.т.н., доц. Гибадуллин Р.Р.
Одобрена	Ученый совет института	18.02.2025	№ 8	_____ И.о. директора ИЭЭ, к.т.н., доц. Гибадуллин Р.Р.

1. Цель, задачи и планируемые результаты обучения по дисциплине

Целью освоения дисциплины является формирование знаний, умений по применению и проектированию современных средств цифровой и микропроцессорной техники, формирование навыков по разработке микропроцессорных устройств на основе современной электронной базы.

Задачами дисциплины являются изучение структуры, архитектуры, функционирования, построения цифровых систем автоматизации, приобретение навыков по применению стандартных цифровых устройств и проектированию микропроцессорных устройств на базе современных микроконтроллеров, их технической реализации и отладки с использованием специализированных программных средств.

Компетенции и индикаторы, формируемые у обучающихся:

Код и наименование компетенции	Код и наименование индикатора
ПК-5 Способностью решать задачи по созданию и отладке программного обеспечения для микропроцессорных систем автоматизации и взаимодействия их модулей	ПК-5.1. Владеть навыками создания управляющих программ
	ПК-5.3. Владеть методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации.

2. Место дисциплины в структуре ОП

Предшествующие дисциплины (модули), практики, НИР, др. Информационные технологии, Схемотехника, Алгоритмизация и программирование.

Последующие дисциплины (модули), Производственная практика (преддипломная)

3. Структура и содержание дисциплины

3.1. Структура дисциплины

Для очной формы обучения

Вид учебной работы	Всего ЗЕ	Всего часов	Семестр(ы)	
			7	8
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	12	432	216	216
КОНТАКТНАЯ РАБОТА*	-	191	81	110
АУДИТОРНАЯ РАБОТА	3,33	120	60	60
Лекции	1,6	42	30	12
Практические (семинарские) занятия	1,15	30	14	16
Лабораторные работы	1,83	48	16	32
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ	8,67	312	156	156
Проработка учебного материала	4,67	168	120	48
Курсовой проект	2	72	0	72
Курсовая работа		0	0	0

Подготовка к промежуточной аттестации	2	72	36	36
Промежуточная аттестация:			Э	Э
				КП

3.2. Содержание дисциплины, структурированное по разделам и видам занятий

Разделы дисциплины	Всего часов	Распределение трудоемкости по видам учебной работы				Формы и вид контроля	Индексы индикаторов формируемых компетенций
		лекции	лаб. раб.	пр. зан.	сам. раб.		
Раздел 1	90	14	8	8	60	ТК1	ПК-5.13, ПК-5.33 ПК-5.3В
Раздел 2	88	16	8	6	60	ТК2	ПК-5.13, ПК-5.1В, ПК-5.3В
Экзамен	36				36	ОМ 1	ПК-5.13, ПК-5.33 ПК-5.1В, ПК-5.3В
Итого за 7 семестр	216	30	16	14	156		
Раздел 3	72	6	16	8	42	ТК3	ПК-5.13, ПК-5.33 ПК-5.3В
Раздел 4	72	6	16	8	42	ТК4	ПК-5.13, ПК-5.1В, ПК-5.3В
Курсовой проект	72				72	ОМкп	ПК-5.1У, ПК-5.3У ПК-5.1В, ПК-5.3В
Экзамен	36				36	ОМ 2	ПК-5.13, ПК-5.33 ПК-5.1У, ПК-5.3У ПК-5.1В, ПК-5.3В
Итого за 8 семестр	216	12	32	16	156		
ИТОГО	432	42	48	30	312		

3.3. Содержание дисциплины

Раздел 1. Основы программирования на языке C++ для микроконтроллеров с архитектурой ARM Cortex.

Тема 1.1. Знакомство со средой программирования IMLAB IDE.

Тема 1.2. Изучение принципов работы с ЖКИ и способов его программирования.

Тема 1.3. Таймеры/Счётчики.

Тема 1.4. Изучение принципов работы и способов микроконтроллерного управления шаговым электроприводом.

Раздел 2. Архитектура процессорного ядра ARM Cortex.

Тема 2.1. Системы команд ARM Cortex.

Тема 2.2. Функции и режимы работы портов ввода/вывода общего назначения.

Тема 2.3. Прерывания и события.

Раздел 3. Исследование режимов работы АЦП.

Тема 3.1. Прерывания АЦП.
Тема 3.2. Режимы работы АЦП.

Раздел 4. Исследование режимов работы Таймера реального времени (RTC).

Тема 4.1. Исследование основ RTC.
Тема 4.2. Исследование режима работы генерации сигналов.
Тема 4.3. Исследование режима энергосбережения RTC

**3.4. Тематический план практических занятий
Семестр 7.**

Практическое занятие 1. Переменные, типы данных и преобразования языка программирования C++.

Практическое занятие 2. Форматный вывод данных.

Практическое занятие 3. Арифметические и логические операторы.

Практическое занятие 4. Условные циклы while, do while, for.

Практическое занятие 5. Условные и безусловные операторы.

Практическое занятие 6. Модульное программирование.

Практическое занятие 7. Компиляция и отладка программ.

Семестр 8.

Практическое занятие 8. Порты ввода/вывода общего назначения (GPIO).

Практическое занятие 9. Универсальный синхронный/асинхронный приемник передатчик (USART/UART). Принцип работы.

Практическое занятие 10. Передатчик и приемник USART/UART.

Практическое занятие 11. Режимы работы USART/UART.

Практическое занятие 12. Прерывания USART/UART.

Практическое занятие 13. Интерфейсы физического уровня RS-485 и RS-232.

Практическое занятие 14. Контроллер прямого доступа к памяти (DMA). Режимы работы.

Практическое занятие 15. Прерывания и события DMA.

3.5. Тематический план лабораторных работ

Лабораторная работа 1. Знакомство со средой программирования IMLAB IDE.

Лабораторная работа 2. Изучение принципов работы с ЖКИ и способов его программирования.

Лабораторная работа 3. Таймеры/Счётчики.

Лабораторная работа 4. Изучение принципов работы и способов микроконтроллерного управления шаговым электроприводом.

Лабораторная работа 5. Системы команд процессорного ядра ARM Cortex.

Лабораторная работа 6. Функции и режимы работы портов ввода/вывода общего назначения.

Лабораторная работа 7. Прерывания и события.

Лабораторная работа 8. Прерывания АЦП

Лабораторная работа 9. Режимы работы АЦП

Лабораторная работа 10. Исследование основ RTC.

Лабораторная работа 11. Исследование режима работы генерации сигналов.

Лабораторная работа 12. Исследование режима энергосбережения RTC

3.6. Курсовой проект /курсовая работа

Примеры тем заданий на курсовое проектирование.

– Разработать генератор прямоугольных импульсов. Частота импульсов $f_{вых}$ на выходе генератора в герцах 1, 10 задается переключателями и в двоично-десятичном коде отображается на цифровом индикаторе. Длительность импульса 500мс, 50мс соответственно. Кнопка ПУСК запускает генератор. Кнопка СТОП – выключает.

– Разработать устройство измерения временного интервала между двумя событиями (включить конвейер, фотодатчик). Вывести на цифровой индикатор число деталей, прошедших через конвейер за время его включения. Фотодатчик формирует одиночный импульс в виде логического нуля при прохождении детали по конвейеру.

– Разработать устройство для выполнения опроса сигнала на входе с интервалом 90 мс, после тридцатого обнаружения сигнала «логического нуля» отобразить на цифровом индикаторе информацию количества опроса сигнала и выдать звуковой сигнал и включить зеленый светодиод. Опрос начинается при нажатии кнопок ПУСК и НУЛЬ.

Курсовой проект включает следующие этапы разработки:

- создание исходного текста программы на языке C++;
- компиляция и отладка управляющей программы;
- разработка структурной и принципиальной электрической схемы цифровой системы в формате А3.

4. Оценивание результатов обучения

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлетворительно	неудовлетворительно
			зачтено			не зачтено

ПК-5	ПК-5.1	знать:				
		как использовать информационные технологии и методы обработки информации в области промышленной электроники	В полном объеме знает, как использовать информационные технологии и методы обработки информации в области промышленной электроники	Достаточно полно знает, как использовать информационные технологии и методы обработки информации в области промышленной электроники	Плохо ориентируется в использовании информационных технологий и методов обработки информации в области промышленной электроники	Не знает, как использовать информационные технологии и методы обработки информации в области промышленной электроники
		уметь:				
		использовать информационные технологии и методы обработки информации в области промышленной электроники	Свободно использует информационные технологии и методы обработки информации в области промышленной электроники	Умеет использовать информационные технологии и методы обработки информации в области промышленной электроники	Слабо ориентируется в использовании информационных технологий и методов обработки информации в области промышленной электроники	Не умеет использовать информационные технологии и методы обработки информации в области промышленной электроники
	ПК-5.3	владеть:				
		навыками создания управляющих программ	В полном объеме владеет навыками создания управляющих программ	Достаточно полно владеет навыками создания управляющих программ	Плохо описывает создание управляющих программ	Не владеет навыками создания управляющих программ
		знать:				
		методы использования цифровой и микропроцессорной техники,	В полном объеме знает методы использования	Достаточно полно знает методы использования	Плохо ориентируется в методах использования	Не знает методы использования цифрово

		программное обеспечение при проектировании и электронных устройств различного функционального назначения	ания цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения	ания цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения	ания цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения	й и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения
		уметь:				
		использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании и электронных устройств различного функционального назначения	Свободно использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании электронных устройств различного функционального назначения	Умеет использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании электронных устройств различного функционального назначения	Слабо ориентируется в использовании основы структуры и архитектуры цифровой и микропроцессорной техники, приемов программирования при проектировании электронных устройств различного функционального назначения	Не умеет использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании электронных устройств различного функционального назначения
		владеть:				

		методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	В полном объеме владеет методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	Достаточно полно владеет методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	Плохо описывает методы отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	Не владеет методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации
--	--	---	---	--	---	--

Оценочные материалы для проведения текущего контроля и промежуточной аттестации приведены в Приложении к рабочей программе дисциплины.

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика.

5. Учебно-методическое и информационное обеспечение дисциплины

5.1. Учебно-методическое обеспечение

5.1.1. Основная литература

1. Гусев, В. Г., Электроника и микропроцессорная техника : учебник / В. Г. Гусев, Ю. М. Гусев. - Москва : КноРус, 2022. - 798 с. - ISBN 978-5-406-08700-8. - URL: <https://book.ru/book/941129> - Текст : электронный.

2. Искусственный интеллект в задачах моделирования, управления, диагностики технологических процессов : монография / А. П. Веревкин, Т. М. Муртазин. - Вологда : Инфра-Инженерия, 2023. - 232 с. - ISBN 978-5-9729-1428-9. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/347132>

3. Гуров В. В. Архитектура микропроцессоров: учебное пособие / В.В.Гуров. - Москва: Национальный Открытый Университет ИНТУИТ, 2016. – 327 с. Текст - непосредственный.

4. Матюшин А. О. Программирование микроконтроллеров. Стратегия и тактика: учебное пособие / А. О. Матюшин. - Москва: ДМК Пресс, 2017. - 356 с. Текст - непосредственный.

5. Карпов А. Т. Цифровые системы автоматического регулирования: учебное пособие / А. Т. Карпов. - Томск: ТУСУР, 2015. - 216 с. Текст - непосредственный.

6. Рацеев С. М. Программирование на языке Си: учебное пособие / С. М. Рацеев. - Санкт-Петербург : Лань, 2023. - 329 с. - ISBN 978-5-8114-8585-7.- URL: <https://e.lanbook.com/book/193320> - Текст : электронный.

7. Павловская Т. А. Программирование на языке C++: учебное пособие /

Т. А. Павловская. - 2-е изд., испр. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. - 154 с. - URL: <https://e.lanbook.com/book/100409>. - Текст : электронный.

8. Васильев А. Е. Встраиваемые системы автоматики и вычислительной техники. Микроконтроллеры. - М.: Горячая линия - Телеком, 2021. - 590 с., ISBN 978-5-9912-0645-7. Текст : непосредственный.

9. Программирование микроконтроллера MICROCHIP PIC18F45K22 / Методические указания к выполнению лабораторных работ МПС-НН. - Челябинск : ЛАБСИС, 2023. - 71 с. - Текст : непосредственный.

5.1.2.Дополнительная литература

1. Покоев П. Н. Электроника. Расчет электронных схем : учебное пособие / П. Н. Покоев. - Ижевск : Ижевская ГСХА, 2019. - 82 с. - Текст : электронный // Лань : электронная-библиотечная система. - URL: <https://e.lanbook.com/book/178035>

2. Кангин В. В. Контроллера Arduino в мобильных роботах : учебное пособие / В. В. Кангин. - Старый Оскол : ТНТ, 2024. - 396 с.

3. Ефимов, И. Е. Основы микроэлектроники : учебник / И. Е. Ефимов, И. Я. Козырь. - 3-е изд. - Санкт-Петербург : Лань, 2022. - 384 с. - ISBN 978-5-8114-0866-5. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/210218>.

4. Муромцев Д. Ю. Проектирование функциональных узлов и модулей радиоэлектронных средств : учебное пособие для вузов / Д. Ю. Муромцев, И. В. Тюрин, О. А. Белоусов, Р. Ю. Курносов. - 2-е изд., стер. - Санкт-Петербург : Лань, 2021. - 251 с. - ISBN 978-5-8114-8814-8. - Текст : электронный // Лань: электронно-библиотечная система. - URL: <https://e.lanbook.com/book/181532>.

5. Трегубов С. И. Основы конструирования электронных средств: техническое задание : учебное пособие / С. И. Трегубов, А. А. Левицкий. - Красноярск : СФУ, 2020. - 180 с. - ISBN 978-5-7638-4257-9. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/181591>.

6. Муромцев Д. Ю. Конструирование блоков радиоэлектронных средств : учебное пособие для вузов / Д. Ю. Муромцев, О. А. Белоусов, И. В. Тюрин, Р. Ю. Курносов. - 2-е изд., стер. - Санкт-Петербург : Лань, 2022. - 288 с. - ISBN 978-5-507-44388-8. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/226472>.

7. Сонькин М. А. Микропроцессорные системы. Средства разработки программного обеспечения для микроконтроллеров семейства AVR [Электронный ресурс] : учебное пособие / М.А. Сонькин, А.А. Шамин. - Электрон.дан. - Томск : ТПУ, 2016. - 90 с. - Режим доступа: <https://e.lanbook.com/book/107725>.

8. Основы цифровой электроники : учебно-методическое пособие / Л. В. Ахметвалеева, Л. Г. Кулагина. - Казань : КГЭУ, 2018. - 100 с. - URL: <https://lib.kgeu.ru>. - Текст : электронный

9. Ахметвалеева Л. В. Основы микропроцессорной техники. Лабораторный практикум по дисциплинам «Информационная электроника», «Основы микропроцессорной техники», «Программирование цифровых систем автоматизации» : учебно-методическое пособие / Л. В. Ахметвалеева. - Казань: Казан. гос. энерг. ун-т, 2015. - 108 с.

10. Ахметвалеева Л. В. Цифровые устройства : учебное пособие / Казань : Казан. гос. энерг. ун-т, 2002. - 171 с.

5.2. Информационное обеспечение

5.2.1. Электронные и интернет-ресурсы

№ п/п	Наименование электронных и интернет-ресурсов	Ссылка
1	Электронно-библиотечная система «Лань»	https://e.lanbook.com/
2	Электронно-библиотечная система «ibooks.ru»	https://ibooks.ru/
3	Электронно-библиотечная система «book.ru»	https://www.book.ru/
4	Энциклопедии, словари, справочники	http://www.rubricon.com
5	Портал "Открытое образование"	http://npoad.ru

5.2.2. Профессиональные базы данных / Информационно-справочные системы

№ п/п	Наименование профессиональных баз данных	Адрес	Режим доступа
1	Российская национальная библиотека	http://nlr.ru/	http://nlr.ru/
2	Единое окно доступа к образовательным ресурсам	http://window.edu.ru/	http://window.edu.ru/
3	Web of Science	https://webofknowledge.com/	https://webofknowledge.com/
4	Scopus	https://www.scopus.com	https://www.scopus.com
5	КиберЛенинка	https://cyberleninka.ru/	https://cyberleninka.ru/
6	Научная электронная библиотека eLIBRARY.RU	http://elibrary.ru	http://elibrary.ru
7	Национальная электронная библиотека (НЭБ)	https://rusneb.ru/	https://rusneb.ru/
8	Техническая библиотека	http://techlibrary.ru	http://techlibrary.ru
9	IEEE Xplore	www.ieeeexplore.ieee.org	www.ieeeexplore.ieee.org
10	Springer	www.springer.com	www.springer.com
11	«Консультант плюс»	http://www.consultant.ru/	http://www.consultant.ru/

5.2.3. Лицензионное и свободно распространяемое программное обеспечение дисциплины

№ п/п	Наименование программного обеспечения	Описание
1	Windows 10 Домашняя / Pro / Для образовательных учреждений	Пользовательская операционная система
2	Windows 7 Профессиональная (Pro)	Пользовательская операционная система
3	Браузер Chrome	Система поиска информации в сети интернет
4	Office Standard 2007 Russian OLP NL AcademicEdition+	Пакет программных продуктов, содержащий в себе необходимые офисные программы
5	MS Office 2010 Russian OLP NL	Пакет программных продуктов, содержащий в себе необходимые офисные программы
6	OpenOffice	Пакет офисных приложений
7	Adobe Acrobat	Пакет программ для создания и просмотра файлов формата PDF
8	LMS Moodle	ПО для эффективного онлайн-взаимодействия преподавателя и студента
9	STM32_CooCox_Installation	ПО для отладки управляющей программы
10	MPLAB	Среда разработки от Microchip, содержащая компилятор с языка ассемблера, позволяющая провести полный цикл разработки в том числе и прошивку
11	Proteus ISIS	Симулятор-отладчик.

6. Материально-техническое обеспечение дисциплины

Наименование вида учебной работы	Наименование учебной аудитории, специализированной лаборатории	Перечень необходимого оборудования и технических средств обучения
Лекции	Учебная аудитория для проведения занятий лекционного типа	Специализированная учебная мебель, технические средства обучения, служащие для представления учебной информации большой аудитории (мультимедийный проектор, компьютер (ноутбук), экран), демонстрационное оборудование, учебно-наглядные пособия
Практические занятия	Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и	Специализированная учебная мебель, технические средства обучения (мультимедийный проектор, компьютер (ноутбук), экран) и др.

	промежуточной аттестации А-405, А-404	
Лабораторные работы	Учебная лаборатория А-410	Специализированное лабораторное оборудование по профилю лаборатории:
	Компьютерный класс с выходом в Интернет А-410, А-405	Специализированная учебная мебель, технические средства обучения (мультимедийный проектор, компьютер (ноутбук, экран), лицензионное программное обеспечение Специализированное лабораторное оборудование «ЛАБСИС» модуль «Микроконтроллер AVR», модуль «Микроконтроллер архитектуры PIC» по профилю лаборатории «Микропроцессорные системы»
Самостоятельная работа	Компьютерный класс с выходом в Интернет В-600а	Специализированная учебная мебель на 30 посадочных мест, 30 компьютеров, технические средства обучения (мультимедийный проектор, компьютер (ноутбук), экран), видеокамеры, программное обеспечение
	Читальный зал библиотеки	Специализированная мебель, компьютерная техника с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, экран, мультимедийный проектор, программное обеспечение

7. Особенности организации образовательной деятельности для лиц с ограниченными возможностями здоровья и инвалидов

Лица с ограниченными возможностями здоровья (ОВЗ) и инвалиды имеют возможность беспрепятственно перемещаться из одного учебно-лабораторного корпуса в другой, подняться на все этажи учебно-лабораторных корпусов, заниматься в учебных и иных помещениях с учетом особенностей психофизического развития и состояния здоровья.

Для обучения лиц с ОВЗ и инвалидов, имеющих нарушения опорно-двигательного аппарата, обеспечены условия беспрепятственного доступа во все учебные помещения. Информация о специальных условиях, созданных для обучающихся с ОВЗ и инвалидов, размещена на сайте университета www/kgeu.ru. Имеется возможность оказания технической помощи ассистентом, а также услуг сурдопереводчиков и тифлосурдопереводчиков.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушенным слухом справочного, учебного материала по дисциплине обеспечиваются следующие условия:

- для лучшей ориентации в аудитории, применяются сигналы оповещения о начале и конце занятия (слово «звонок» пишется на доске);
- внимание слабослышащего обучающегося привлекается педагогом

жестом (на плечо кладется рука, осуществляется нерезкое похлопывание);

- разговаривая с обучающимся, педагогический работник смотрит на него, говорит ясно, короткими предложениями, обеспечивая возможность чтения по губам.

Компенсация затруднений речевого и интеллектуального развития слабослышащих обучающихся проводится путем:

- использования схем, диаграмм, рисунков, компьютерных презентаций с гиперссылками, комментирующими отдельные компоненты изображения;
- регулярного применения упражнений на графическое выделение существенных признаков предметов и явлений;
- обеспечения возможности для обучающегося получить адресную консультацию по электронной почте по мере необходимости.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушениями зрения справочного, учебного, просветительского материала, предусмотренного образовательной программой по выбранному направлению подготовки, обеспечиваются следующие условия:

- ведется адаптация официального сайта в сети Интернет с учетом особых потребностей инвалидов по зрению, обеспечивается наличие крупношрифтовой справочной информации о расписании учебных занятий;
- педагогический работник, его собеседник (при необходимости), присутствующие на занятии, представляются обучающимся, при этом каждый раз называется тот, к кому педагогический работник обращается;
- действия, жесты, перемещения педагогического работника коротко и ясно комментируются;
- печатная информация предоставляется крупным шрифтом (от 18 пунктов), тотально озвучивается;
- обеспечивается необходимый уровень освещенности помещений;
- предоставляется возможность использовать компьютеры во время занятий и право записи объяснений на диктофон (по желанию обучающихся).

Форма проведения текущей и промежуточной аттестации для обучающихся с ОВЗ и инвалидов определяется педагогическим работником в соответствии с учебным планом. При необходимости обучающемуся с ОВЗ, инвалиду с учетом их индивидуальных психофизических особенностей дается возможность пройти промежуточную аттестацию устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п., либо предоставляется дополнительное время для подготовки ответа.

8. Методические рекомендации для преподавателей по организации воспитательной работы с обучающимися.

Методическое обеспечение процесса воспитания обучающихся выступает одним из определяющих факторов высокого качества образования. Преподаватель вуза, демонстрируя высокий профессионализм, эрудицию, четкую гражданскую позицию, самодисциплину, творческий подход в решении профессиональных задач, в ходе образовательного процесса способствует формированию гармоничной личности.

При реализации дисциплины преподаватель может использовать следующие методы воспитательной работы:

- методы формирования сознания личности (беседа, диспут, внушение, инструктаж, контроль, объяснение, пример, самоконтроль, рассказ, совет, убеждение и др.);

- методы организации деятельности и формирования опыта поведения (задание, общественное мнение, педагогическое требование, поручение, приучение, создание воспитывающих ситуаций, тренинг, упражнение, и др.);

- методы мотивации деятельности и поведения (одобрение, поощрение социальной активности, порицание, создание ситуаций успеха, создание ситуаций для эмоционально-нравственных переживаний, соревнование и др.)

При реализации дисциплины преподаватель должен учитывать следующие направления воспитательной деятельности:

Гражданское и патриотическое воспитание:

- формирование у обучающихся целостного мировоззрения, российской идентичности, уважения к своей семье, обществу, государству, принятым в семье и обществе духовно-нравственным и социокультурным ценностям, к национальному, культурному и историческому наследию, формирование стремления к его сохранению и развитию;

- формирование у обучающихся активной гражданской позиции, основанной на традиционных культурных, духовных и нравственных ценностях российского общества, для повышения способности ответственно реализовывать свои конституционные права и обязанности;

- развитие правовой и политической культуры обучающихся, расширение конструктивного участия в принятии решений, затрагивающих их права и интересы, в том числе в различных формах самоорганизации, самоуправления, общественно-значимой деятельности;

- формирование мотивов, нравственных и смысловых установок личности, позволяющих противостоять экстремизму, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам, межэтнической и межконфессиональной нетерпимости, другим негативным социальным явлениям.

Духовно-нравственное воспитание:

- воспитание чувства достоинства, чести и честности, совестливости, уважения к родителям, учителям, людям старшего поколения;

- формирование принципов коллективизма и солидарности, духа милосердия и сострадания, привычки заботиться о людях, находящихся в трудной жизненной ситуации;

- формирование солидарности и чувства социальной ответственности по отношению к людям с ограниченными возможностями здоровья, преодоление психологических барьеров по отношению к людям с ограниченными возможностями;

- формирование эмоционально насыщенного и духовно возвышенного отношения к миру, способности и умения передавать другим свой эстетический опыт.

Культурно-просветительское воспитание:

- формирование эстетической картины мира;
- формирование уважения к культурным ценностям родного города, края, страны;
- повышение познавательной активности обучающихся.

Научно-образовательное воспитание:

- формирование у обучающихся научного мировоззрения;
- формирование умения получать знания;
- формирование навыков анализа и синтеза информации, в том числе в профессиональной области.

Вносимые изменения и утверждения на новый учебный год

№ п/п	№ раздела внесения изменений	Дата внесения изменений	Содержание изменений	«Согласовано» Зав. каф. реализующей дисциплину	«Согласовано» председатель УМК института (факультета), в состав которого входит выпускающая
1	2	3	4	5	6
1					
2					
3					



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГУ»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
по дисциплине**

Б1.В.ДЭ.02.03.04. Программирование управляющих устройств систем
автоматизации

Направление подготовки	<u>11.03.04 Электроника и нанoeлектроника</u>
Профиль	<u>Программирование и электроника информационных систем</u>
Квалификация	<u>Бакалавр</u>

Оценочные материалы по дисциплине «Цифровая и микропроцессорная техника», предназначенны для оценивания результатов обучения на соответствие индикаторам достижения компетенций.

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля (ТК) и промежуточной аттестации, проводимых по балльно-рейтинговой системе (БРС).

1.Технологическая карта

Семестр 7, 8

Наименование раздела	Формы и вид контроля	Рейтинговые показатели									
		I текущий контроль	Дополнительные баллы к ТК1	II текущий контроль	Дополнительные баллы к ТК2	III текущий контроль	Дополнительные баллы к ТК3	IV текущий контроль	Дополнительные баллы к ТК4	Итого	Промежуточная аттестация
Раздел 1	ТК1	20	0-10							20-30	20-30
Письменный опрос		6									
Защита лабораторной работы		7									
Практические занятия		7									
Раздел 2	ТК2			20	0-5					20-25	20-25
Письменный опрос				6							
Защита лабораторной работы				7							
Практические занятия				7							
Промежуточная аттестация, 7 семестр (экзамен)	ОМ1										0-45
Задание промежуточной аттестации											0-20
В письменной форме по билетам											0-25
Раздел 3	ТК3					20	0-10			20-30	20-30
Тест или письменный опрос						6					
Практические занятия						7					
Защита лабораторной работы						7					
Раздел 4	ТК4							20	0-5	20-25	20-250

Тест или письменный опрос								6			
Защита лабораторной работы								7			
Практические занятия								7			
Промежуточная аттестация, 8 семестр (экзамен, КП)	ОМ2										0-45
Задание промежуточной аттестации											0-20
В письменной форме по билетам											0-25

2. Оценочные материалы текущего контроля и промежуточной аттестации

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции				
			Высокий	Средний	Ниже среднего	Низкий	
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54	
			Шкала оценивания				
			отлично	хорошо	удовлетворительно	неудовлетворительно	
			зачтено			не зачтено	
ПК-5	ПК-5.1	знать:					
		как использовать информационные технологии и методы обработки информации в области промышленной электроники	В полном объеме знает, как использовать информационные технологии и методы обработки информации в области промышленной электроники	Достаточно полно знает, как использовать информационные технологии и методы обработки информации в области промышленной электроники	Плохо ориентируется в использовании информационных технологий и методов обработки информации в области промышленной электроники	Не знает, как использовать информационные технологии и методы обработки информации в области промышленной электроники	
		уметь:					
		использовать	Свободно	Умеет	Слабо	Не умеет	

		информационные технологии и методы обработки информации в области промышленной электроники	используют информационные технологии и методы обработки информации в области промышленной электроники	использовать информационные технологии и методы обработки информации в области промышленной электроники	ориентир уется в использовании информационных технологий и методов обработки информации в области промышленной электроники	использовать информационные технологии и методы обработки информации в области промышленной электроники
		владеть:				
		навыками создания управляющих программ	В полном объеме владеет навыками создания управляющих программ	Достаточно полно владеет навыками создания управляющих программ	Плохо описывает создание управляющих программ	Не владеет навыками создания управляющих программ
	ПК-5.3	знать:				
		методы использования цифровой и микропроцессорной техники, программное обеспечение при проектировании и электронных устройств различного функционального назначения	В полном объеме знает методы использования цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения	Достаточно полно знает методы использования цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения	Плохо ориентируется в методах использования цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения	Не знает методы использования цифровой и микропроцессорной техники, программное обеспечение при проектировании электронных устройств различного функционального назначения

		уметь:				
		использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании и электронных устройств различного функционального назначения	Свободно использует основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании электронных устройств различного функционального назначения	Умеет использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании электронных устройств различного функционального назначения	Слабо ориентируется в использовании основы структуры и архитектуры цифровой и микропроцессорной техники, приемов программирования при проектировании электронных устройств различного функционального назначения	Не умеет использовать основы структуры и архитектуры цифровой и микропроцессорной техники, приемы программирования при проектировании электронных устройств различного функционального назначения
		владеть:				
		методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	В полном объеме владеет методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	Достаточно полно владеет методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	Плохо описывает методы отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации	Не владеет методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации

Оценка **«отлично»** выставляется за выполнение *практических и лабораторных работ в семестре; тестовых заданий; выполнение курсового проекта, глубокое понимание программирования управляющих устройств*

систем автоматизации и микроконтроллерных устройств, полные и содержательные ответы на вопросы билета (теоретическое и практическое задание).

Оценка **«хорошо»** выставляется за выполнение практических и лабораторных работ в семестре; тестовых заданий; выполнение курсового проекта, понимание основ программирования управляющих устройств систем автоматизации и микроконтроллерных устройств, ответы на вопросы билета (теоретическое и практическое задание).

Оценка **«удовлетворительно»** выставляется за выполнение практических и лабораторных работ в семестре; тестовых заданий; за небрежное выполнение курсового проекта и приблизительные правильные ответы на вопросы билета на экзамене.

Оценка **«неудовлетворительно»** выставляется за слабое и неполное выполнение практических и лабораторных работ в семестре и невыполнение курсового проекта.

3. Перечень оценочных средств

Краткая характеристика оценочных средств, используемых при текущем контроле успеваемости и промежуточной аттестации обучающегося по дисциплине:

Наименование оценочного средства	Краткая характеристика оценочного средства	Описание оценочного средства
Тест (Тест)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Комплект тестовых заданий
Практическое задание (ПЗ)	Средство оценки умения применять полученные теоретические знания в практической ситуации. Задание направлено на оценивание компетенций по дисциплине, содержит четкую инструкцию по выполнению или алгоритм действий.	Комплект задач и заданий
Отчет по лабораторной работе (ОЛР)	Выполнение лабораторной работы, обработка результатов испытаний, измерений, эксперимента. Оформление отчета, защита результатов лабораторной работы по отчету.	Перечень заданий и вопросов для защиты лабораторной работы, перечень требований к отчету
Курсовой проект (КП),	Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления. Может	Темы проектов

	выполняться в индивиду-альном порядке или группой обучающихся	
--	---	--

4. Перечень контрольных заданий или иные материалы, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций в процессе освоения дисциплины

Требования к оформлению отчета по лабораторным и практическим работам:

Отчет о проделанной работе должен быть выполнен на листах формата А4 и содержать:

- название;
- цель работы;
- задание;
- изображение функциональной схемы;
- ход выполнения работы в отладочной среде программирования (в работе приводится скрин с экрана каждого шага выполнения работы);
- основные выводы;
- краткие ответы на контрольные вопросы.

Требования к содержанию и оформлению курсового проекта:

Курсовой проект состоит из пояснительной записки и графического материала. Объем текстовой части курсового проекта – до 50 страниц машинописного текста на писчей бумаге формата А4 (210×297). Графическая часть включает 3-5 листов стандартного формата.

Текстовая часть типового курсового проекта имеет следующую структуру:

1. Титульный лист
2. Задание на курсовой проект
3. Пояснительная записка, в составе:
 - 3.1. Содержание
 - 3.2. Перечень сокращений, условных обозначений, символов, единиц и терминов (оформляется согласно ЕСКД)
 - 3.3. Введение
 - 3.4. Основная часть
 - 3.5. Заключение
 - 3.6. Список использованных источников
 - 3.7. Приложения (3-5 – листы графической части КП, выполненные на А3)

Задание на курсовой проект «Разработка микропроцессорного устройства на базе RISC – контроллера».

Целью курсового проектирования является закрепление у студентов навыков по построению встраиваемых цифровых систем на базе RISC микроконтроллера, оптимизации выбора и распределения функций между

аппаратными средствами и программным обеспечением разрабатываемого микроконтроллерного устройства по разработке управляющих программ на языках программирования СИ++, ассемблера RISC-контроллера.

Курсовой проект включает следующие разделы:

1. Разработка алгоритма функционирования, структурной схемы цифровой системы на базе RISC- микроконтроллера. Включает описание состава, блоков , из которых состоит управляющий блок цифровой системы, а также описание функционирования. Содержит краткое изложение назначения, особенности и функционирования каждого блока и алгоритма работы: управляющий блок (МК, особенности). блок питания, блок синхронизации, блок управления режимами работы, блок индикации, блок ввода, блок вывода и т. д. в зависимости от конкретного задания.

2. Обоснование выбора и схемотехнического решения аппаратной части управляющего блока цифровой системы. Описание элементной базы - описать стандартные схемотехнические решения, выбранные микросхемы, электронные устройства, пассивные элементы схемы с расчетными решениями и указанием электрических параметров (взять из справочников, даташитов), элементы зарисовать согласно УГО (условное графическое обозначение). Рассчитать электрическое сопряжение элементов цифровой системы. Рассчитать параметры пассивных элементов, указать выбранные стандартные значения.

3. Построение и описание принципиальной электрической схемы разрабатываемой цифровой системы на базе RISC- микроконтроллера.

4. Разработка программного обеспечения - управляющей программы МКУ. Включает описание исходных данных, результатов выходных данных, алгоритма управляющей программы, блок-схему управляющей программы, реализацию ее на языке программирования СИ++ или ассемблера для заданного микропроцессора. Также следует описать режимы, этапы процесса отладки программ в выбранной интегрированной среде разработки.

СЕМЕСТР 7

Для текущего контроля ТК1:

Проверяемая компетенция:

ПК-5 Способностью решать задачи по созданию и отладке программного обеспечения для микропроцессорных систем автоматизации и взаимодействия их модулей (ПК-5.1 Владеть навыками создания управляющих программ; ПК-5.3 Владеть методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации)

Тест

<i>Вопрос</i>	<i>Варианты ответа</i>
<i>В чём заключается основная роль цифровой техники в</i>	<i>увеличение массы оборудования</i>
	<i>автоматизация и повышение точности управления</i>

управлении процессами?	уменьшение расхода электроэнергии
	обеспечение аналогового сигнала
Какое преимущество характерно для RISC-контроллеров?	сложные и длинные команды
	высокое энергопотребление
	простая архитектура и высокая скорость
	использование только внешней памяти
Что входит в архитектуру современных микроконтроллеров?	только арифметико-логическое устройство (АЛУ)
	процессор, память и периферийные модули
	только регистры и таймеры
	ПЗУ и механические реле
К какому типу можно отнести команду перехода (jump) в классификации команд процессора?	арифметическая
	логическая
	команд управления
	графическая
Что характерно для программирования на языке C++?	только процедурное программирование
	использование исключительно для веб-разработки
	поддержка объектно-ориентированного программирования
	не требует компиляции перед запуском

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика и содержит 50 тестовых вопросов на каждую компетенцию, из них 20% - закрытого типа, 80% - открытого типа.

Практическое занятие 1. Переменные, типы данных и преобразования языка программирования C++.

Примеры задач:

1) Напишите программу, которая объявляет две целочисленные переменные a и b, присваивает им значения 5 и 10 соответственно, а затем меняет их значения местами (без использования дополнительных переменных, если возможно). Выведите результат до и после обмена.

2) Напишите программу, которая запрашивает у пользователя число с плавающей точкой (например, 3.14), сохраняет его в переменную типа float, а затем преобразует его в int и double. Выведите все три значения.

3) Напишите программу, которая запрашивает у пользователя три целых числа, сохраняет их в переменные, а затем вычисляет их среднее арифметическое в виде double. Выведите результат.

4) Объявите переменные следующих типов: int, char, bool, float, double. Присвойте им значения, а затем выведите их размеры в байтах с помощью sizeof().

5) Напишите программу, которая запрашивает у пользователя символ, а затем выводит его ASCII-код.

Практическое занятие 2. Форматный вывод данных.

Примеры задач:

1) Напишите программу, которая выводит таблицу умножения для числа 5 в формате:

5 * 1 = 5

5 * 2 = 10

...

$$5 * 10 = 50$$

Условие: Используйте `setw()` для выравнивания чисел по правому краю.

2) Напишите программу, которая запрашивает у пользователя число с плавающей точкой и выводит его:

- С точностью до 2 знаков после запятой.
- В научной нотации (scientific).
- С фиксированной точностью (fixed) и 5 знаками после запятой.

3) Напишите программу, которая выводит данные в виде таблицы (3 столбца: название товара, количество, цена). Используйте `setw()`, `left` и `right` для форматирования.

4) Напишите программу, которая выводит одно и то же целое число (например, 255) в разных системах счисления:

- Десятичной.
- Шестнадцатеричной (с префиксом 0x).
- Восьмеричной (с префиксом 0).

5) Напишите программу, которая выводит строку текста (например, "Hello, C++!") по центру экрана, предполагая ширину вывода 30 символов. Используйте `setw()` и вычисление отступов.

Практическое занятие 3. Арифметические и логические операторы.

Примеры задач:

1) Напишите программу, которая запрашивает у пользователя два числа и символ операции (+, -, *, /, %), а затем выводит результат этой операции.

Условия:

- Если введена операция / и второе число 0, вывести "Ошибка: деление на ноль!".
- Если введен неизвестный оператор, вывести "Неверная операция!".

2) Напишите программу, которая запрашивает у пользователя целое число и проверяет:

- Является ли оно чётным.
- Делится ли оно одновременно на 3 и на 5.

Требования:

- Использовать логические операторы (&&, ||, !).
- Выводить "Да" или "Нет" вместо true/false.

3) Напишите программу, которая проверяет, попадает ли введённое пользователем число в один из диапазонов:

- От 10 до 20 (включительно).
- Меньше 0 или больше 100.

Использовать только логические операторы (без if).

Практическое занятие 4. Условные циклы while, do while, for.

Примеры задач:

1) Напишите программу, которая запрашивает у пользователя целое положительное число N, а затем выводит обратный отсчёт от этого числа до 1 с помощью цикла while. Если введено отрицательное число или 0, вывести

сообщение: "Число должно быть положительным!".

2) Напишите программу, которая запрашивает у пользователя целое число (может быть отрицательным) и вычисляет сумму его цифр с использованием цикла do-while. Для числа -1234 сумма цифр: $1 + 2 + 3 + 4 = 10$ (знак игнорируется). Используйте `abs()` для работы с модулем числа.

3) Напишите программу, которая выводит таблицу степеней числа 2 от 2^0 до 2^N , где N вводит пользователь. Используйте цикл `for`. Если введено $N < 0$, вывести "N должно быть неотрицательным!".

Лабораторная работа 1. Знакомство со средой программирования IMLAB IDE.

Перечень заданий и вопросов для защиты лабораторной работы:

1) Какие основные элементы интерфейса IMLAB IDE можно выделить? Перечислите хотя бы три (например, редактор кода, монитор порта и т. д.).

2) Как в IMLAB IDE выбрать конкретный микроконтроллер и порт для загрузки программы? Опишите шаги.

3) Какие кнопки отвечают за проверку кода (без загрузки) и загрузку программы в микроконтроллер? Что делает каждая из них?

4) Для чего нужен монитор последовательного порта в IMLAB IDE? Как его открыть и какие параметры нужно настроить?

5) Где в IMLAB IDE отображаются ошибки компиляции? Приведите пример синтаксической ошибки, которую может обнаружить среда.

6) Какая минимальная структура программы требуется в IMLAB IDE для работы с микроконтроллером? Напишите шаблон.

7) Как изменить стандартные настройки IMLAB IDE (например, шрифт, тему оформления или путь к проектам)? Где искать эти опции?

Лабораторная работа 2. Изучение принципов работы с ЖКИ и способов его программирования.

Перечень заданий и вопросов для защиты лабораторной работы:

1) Какие основные типы ЖК-дисплеев (LCD) используются в микроконтроллерных проектах? Чем отличается символьный LCD от графического?

2) Какие основные сигнальные линии используются для подключения символьного LCD по параллельному интерфейсу (HD44780)?

3) Какие стандартные библиотеки в IMLAB IDE используются для управления символьным LCD? Как инициализировать дисплей 16x2?

4) Как вывести текст и числа на символьный LCD? Приведите пример кода для вывода "Temp: 25.5C" на второй строке дисплея.

5) Как регулируется контрастность ЖКИ? Почему иногда дисплей показывает "заполненные" квадраты вместо текста?

6) Как вывести на LCD кастомный символ (например, значок градуса или стрелку)? Опишите процесс.

7) Чем отличается программирование графического LCD от символьного? Какие библиотеки используются?

Вопросы к комплексному заданию ТК1

1. Роль цифровой техники в управлении процессами.
2. Понятие цифровые системы автоматизации. Примеры приложений.
3. Организация цифровых систем управления.
4. Архитектура современных микроконтроллеров.
5. RISC- контроллеры. Особенности, характеристики, преимущества.
6. Особенности, преимущества RISC - микроконтроллеров.
7. Логическая модель процессорного ядра RISC – микроконтроллера.
8. Параметры системы команд процессора.
9. Классификация. команд процессора.
10. Особенности программирования на языке C++. Директивы C++.

Для текущего контроля ТК2:

Проверяемая компетенция:

ПК-5 Способностью решать задачи по созданию и отладке программного обеспечения для микропроцессорных систем автоматизации и взаимодействия их модулей (ПК-5.1 Владеть навыками создания управляющих программ; ПК-5.3 Владеть методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации)

Тест

Вопрос	Варианты ответа
Какой из перечисленных интерфейсов является синхронным последовательным?	UART
	I2C
	SPI
	USB
Что из перечисленного является особенностью программирования на C++?	отсутствие поддержки ООП
	использование классов и объектов
	только процедурное программирование
	отсутствие указателей
Какой метод чаще всего используется для отсчёта временных интервалов в RISC-микроконтроллерах?	использование таймеров
	бесконечные циклы
	аппаратные прерывания
	внешние генераторы сигналов
Какой из алгоритмов относится к типовым алгоритмам управления?	сортировка пузырьком
	ПИД-регулятор
	быстрая сортировка
	двоичный поиск
Какой оператор в C++ используется для динамического выделения памяти?	malloc
	calloc
	new
	alloc

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика и содержит 50 тестовых вопросов на каждую компетенцию, из них 20% - закрытого типа, 80% - открытого типа.

Практическое занятие 5. Условные и безусловные операторы.

Примеры задач:

1) Напишите программу, которая запрашивает у пользователя целое число и определяет:

- Кратно ли оно 3, но не кратно 5,
- Кратно ли 5, но не кратно 3,
- Кратно ли 3 и 5 одновременно,
- Не кратно ни одному из них.

2) Напишите программу, которая:

- Запрашивает у пользователя два числа (a и b),
- Предлагает выбрать операцию (+, -, *, /, %),
- Выводит результат с помощью оператора switch.

При делении на 0 выводите "Ошибка: деление на ноль!".

3) Напишите программу, которая:

- Запрашивает у пользователя три числа,
- Используя тернарный оператор, находит максимальное из них,
- Выводит результат.

4) Напишите программу, которая:

- В бесконечном цикле (while (true)) запрашивает у пользователя числа,
- При вводе числа 0 завершает работу с помощью break или goto,
- Считает сумму всех введенных чисел до нуля.

Практическое занятие 6. Модульное программирование.

Примеры задач:

1) Создайте модуль string_utils, содержащий функции:

- int strlen(const char* str) — длина строки,
- void reverseString(char* str) — переворот строки.

Структура проекта:

- string_utils.h — объявления,
- string_utils.cpp — реализации,
- main.cpp — тестирование функций.

2) Разработайте программу-калькулятор, разбитую на модули:

- calculator.h — объявляет функции:

cpp

Copy

```
double add(double a, double b);
```

```
double subtract(double a, double b);
```

```
double multiply(double a, double b);
```

```
double divide(double a, double b);
```

calculator.cpp — реализует эти функции.

- main.cpp — предоставляет меню выбора операций.

Обработайте деление на ноль. Используйте namespace для группировки функций калькулятора.

Практическое занятие 7. Компиляция и отладка программ.

Примеры задач:

1) Напишите простую программу, которая выводит "Hello, World!" и выполняет деление двух чисел (10 / введенное пользователем число).

Требования:

1. Скомпилируйте программу с флагами:

```
bash
Copy
g++ program.cpp -o basic
g++ -Wall -Wextra -pedantic program.cpp -o strict
g++ -O2 program.cpp -o optimized
```

2. Запустите все три версии и сравните:

- Размер получившихся исполняемых файлов (используйте `ls -lh`)
- Поведение при вводе 0 (деление на ноль)

Цель: понять влияние флагов компиляции на размер программы и проверку ошибок.

2) Дан код с 5 ошибками (синтаксис, типы данных, область видимости):

```
cpp
Copy
#include <iostream>

int main() {
    int x = 5
    double y = 3.14;

    std::cout << "Сумма: " << x + y << std::endl

    for (int i = 0; i < 10; i++) {
        std::cout i * i;
    }

    return 0
}
```

Требования:

1. Найдите все ошибки без запуска программы
2. Скомпилируйте с флагом `-Wall` и проверьте, какие ошибки обнаружил компилятор
3. Исправьте программу и проверьте ее работу

Ошибки:

- Отсутствует ; после `int x = 5`
- Несоответствие типов в `x + y`
- Отсутствует ; после `std::cout`
- Пропущены `<<` в `std::cout i * i`
- Отсутствует ; после `return 0`

3) Дан код, вызывающий segmentation fault:

```
cpp
Copy
#include <iostream>

int main() {
    int* ptr = nullptr;
    *ptr = 10; // Ошибка

    int arr[5] = {1, 2, 3, 4, 5};
    for (int i = 0; i <= 5; i++) { // Выход за границы массива
        std::cout << arr[i] << " ";
    }

    return 0;
}
```

Требования:

- Скомпилируйте с флагом отладки:

```
bash
Copy
g++ -g debug.cpp -o debug
```

- Запустите в GDB:

```
bash
Copy
gdb ./debug
```

- Найдите строки, вызывающие ошибки, используя:
 - `run` — запуск программы
 - `backtrace` — вывод стека вызовов
 - `print` — вывод значений переменных

Цель: научиться находить runtime-ошибки с помощью отладчика.

4) Программа должна вычислять факториал числа, но содержит логическую ошибку:

```
cpp
Copy
#include <iostream>

int factorial(int n) {
    int result = 0; // Ошибка инициализации
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    return result;
}

int main() {
    std::cout << "5! = " << factorial(5) << std::endl; // Выведет 0
    return 0;
}
```

Требования:

1. Скомпилируйте и запустите программу
 2. С помощью отладчика (GDB) или пошагового вывода найдите ошибку:
 - Добавьте `std::cout << "i: " << i << ", result: " << result << std::endl;` в цикл
 3. Исправьте ошибку (инициализация `result = 1`)
- Дополнительно скомпилируйте с оптимизацией (-O2) и проверьте, изменилось ли поведение.

Лабораторная работа 3. Таймеры/Счётчики.

Перечень заданий и вопросов для защиты лабораторной работы:

- 1) Что такое таймер/счётчик в микроконтроллере? Чем отличается таймер от счётчика?
- 2) Какие основные режимы работы таймера вы знаете? Кратко опишите каждый.
- 3) Как рассчитать частоту переполнения таймера (например, для режима CTC) при:
 - Тактовой частоте МК = 16 МГц,
 - Прескалер = 256,
 - Значение OCR1A = 62499?
- 4) Как настроить таймер (например, Timer1) на генерацию прерывания при переполнении? Приведите пример кода
- 5) Как настроить таймер (например, Timer0) для генерации ШИМ-сигнала

на ножке OC0A.

6) Как использовать таймер для измерения длительности положительного импульса на входе?

Лабораторная работа 4. Изучение принципов работы и способов микроконтроллерного управления шаговым электроприводом.

Перечень заданий и вопросов для защиты лабораторной работы:

1) Как устроен шаговый двигатель? Какие основные типы обмоток существуют и чем они отличаются?

2) Какие микросхемы/драйверы используются для управления шаговым двигателем?

3) Чем отличается управление в режимах полного шага, полушага и микрошага? Как это влияет на точность и момент?

4) Как сгенерировать последовательность импульсов для шагового двигателя в биполярном режиме?

5) Как использовать таймер микроконтроллера для плавного вращения шагового двигателя? Как рассчитать частоту STEP-сигнала для скорости 60 об/мин?

6) Что делать, если шаговый двигатель перегревается или пропускает шаги? Какие могут быть причины?

Вопросы к комплексному заданию ТК2

1. Приемы программирования RISC - микроконтроллера .
2. Программирование портов ввода/вывода RISC - микроконтроллера.
3. Синхронный последовательный интерфейс SPI.
4. Операторы, функции языка C++.
5. Особенности программирования на языке C++.
6. Поясните типовые алгоритмы управления.
7. Сопряжение микроконтроллеров с устройствами ввода/вывода.
8. Обработка прерываний в RISC - микропроцессорах.
9. Приемы программирования отсчета временных интервалов.
10. Метод программных циклов RISC - микроконтроллеров.

Для промежуточной аттестации в 7 –м семестре:

Вопросы к экзамену:

1. Структура микропроцессорных систем. Шинная структура связей. Архитектуры микропроцессорных систем. Особенности построения управляющих, контрольно-измерительных устройств на микроконтроллерах.
2. Однокристальные микроконтроллеры (структура, характеристики, применение).
3. Архитектура микроконтроллеров. (Гарвардская, CISC, RISC, RISC-V архитектуры. Особенности, преимущества).

4. Процессорное ядро микроконтроллера. (Основные характеристики, функции. Назначение основных узлов).
5. Архитектура центрального процессора микроконтроллера RISC - архитектуры.
6. Программно-логическая модель процессора.
7. Регистр процессора – регистр признаков (флажков) RISC - микроконтроллера.
8. Система команд микропроцессора. На примере RISC - микроконтроллера. Формат команд. Характеристики команд.
9. Способы адресации операндов. Прямая, непосредственная адресация. Привести примеры.
10. Способы адресации команд МП. Индексная адресация. Привести примеры.

СЕМЕСТР 8

Примеры заданий

Для текущего контроля ТКЗ:

Проверяемая компетенция:

ПК-5 Способностью решать задачи по созданию и отладке программного обеспечения для микропроцессорных систем автоматизации и взаимодействия их модулей (ПК-5.1 Владеть навыками создания управляющих программ; ПК-5.3 Владеть методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации)

Тест

Вопрос	Варианты ответа
Какой модуль микроконтроллера отвечает за работу с временными интервалами в режиме реального времени?	АЦП (Аналого-цифровой преобразователь)
	таймер/счетчик
	UART
	SPI
Что измеряет режим «входного захвата» таймера/счетчика?	частоту генератора
	длительность импульса или момент события
	напряжение на входе
	количество циклов процессов
Какой тип данных в C++ используется для хранения целых чисел со знаком?	unsigned int
	float
	int
	char
Какая функция таймера-счетчика позволяет генерировать ШИМ-сигнал?	режим сравнения
	режим захвата
	режим переполнения
	режим сна
Как классифицируются команды процессора по выполняемым действиям?	арифметические, логические, передачи данных, управления
	только арифметического и логического
	ввод, вывод, обработка
	линейные, циклические, условные

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика и содержит *50 тестовых вопросов на каждую компетенцию, из них 20% - закрытого типа, 80% - открытого типа.*

Практическое занятие 8. Порты вводы/вывода общего назначения (GPIO).

Примеры задач:

1) Настроить вывод микроконтроллера (например, PC13 в STM32) как выход и запрограммировать мигание светодиода с частотой 1 Гц (1 сек – включён, 1 сек – выключен).

2) Настроить вывод (например, PD2) как вход с внутренней подтяжкой к питанию (+VCC). При нажатии кнопки (подключённой к земле) включать светодиод на другом выводе (PB0).

3) Используя GPIO и программный ШИМ (без таймеров), плавно изменять яркость светодиода (например, на PB1) от 0 до 100% и обратно.

Алгоритм:

- Настроить PB1 как выход.
- В цикле изменять коэффициент заполнения ($0 \rightarrow 255 \rightarrow 0$).
- Использовать `_delay_us()` для генерации ШИМ.

4) Настроить кнопку на выводе PD3 и светодиод на PD4. При нажатии кнопки светодиод должен менять состояние (вкл/выкл) без ложных срабатываний из-задребезга. Метод: Аппаратная или программная задержка (например, 50 мс).

5) Реализовать «бегущий огонь» на 4 светодиодах (подключённых к выводам PB0–PB3). Использовать битовые операции для циклического сдвига.

Практическое занятие 9. Универсальный синхронный/асинхронный приемник передатчик (USART/UART). Принцип работы.

Примеры задач:

1) Напишите код для настройки USART1 в микроконтроллере STM32 (или UART в AVR) со следующими параметрами:

- Скорость: 9600 бод,
- 8 бит данных,
- 1 стоп-бит,
- Без контроля чётности.

2) Напишите функцию, которая отправляет строку "Hello, UART!" через USART1 с использованием блокирующей передачи.

3) Настройте USART2 на приём данных по прерыванию. При получении символа 'A' включайте светодиод на PC13, при 'B' — выключайте.

4) Соедините два микроконтроллера (например, STM32 и AVR) через UART. Первый МК отправляет числа 0..9, второй принимает их и возвращает квадрат числа.

5) Настройте USART3 для обнаружения ошибок (переполнение, шум, фрейм). При ошибке отправляйте сообщение "UART Error!" и мигайте

светодиодом.

Практическое занятие 10. Передатчик и приемник USART/UART.

Примеры задач:

1) Настройте UART (например, USART1 в STM32) для передачи строки "Hello, UART!" каждую секунду.

Требования:

- Скорость: 115200 бод,
- Формат: 8 бит данных, 1 стоп-бит, без контроля чётности.

2) Настройте UART на приём одного символа и отправку его обратно (эхо). При получении символа 'A' дополнительно включайте светодиод.

3) Передавайте через UART значение переменной counter (инкрементируется каждые 100 мс) в виде строки: "Counter: XX\r\n".

4) Настройте приём строки (до символа '\n') через прерывания. При получении строки "ON" включайте светодиод, при "OFF" — выключайте.

5) Соедините два микроконтроллера (например, STM32 и ESP32) через UART. Первый отправляет команды:

- "LED ON", "LED OFF" — управление светодиодом на втором МК,
- "GET TEMP" — запрос температуры (второй МК отвечает случайным числом 20..25).

Практическое занятие 11. Режимы работы USART/UART.

Примеры задач:

1) Настройте USART1 в асинхронном режиме (9600 бод, 8 бит данных, 1 стоп-бит, без чётности) с обработкой ошибок:

- При обнаружении ошибки фрейма (FE) или переполнения (ORE) отправлять сообщение "ERROR" и мигать светодиодом.

2) Настройте USART2 в синхронном режиме (CLK на выход, скорость 1 МГц) для обмена данными с внешним АЦП. Отправьте команду 0xA1 и примите 2 байта ответа.

3) Настройте UART0 (9600 бод, 7 бит данных, чётность) для приёма данных. При обнаружении ошибки чётности отправлять "PARITY ERROR".

4) Настройте USART1 в однопроводном режиме (TX и RX на одном выводе PA9) для обмена с датчиком. Реализуйте запрос-ответ:

- МК отправляет 0x55,
 - Переключается на приём,
 - Ждёт ответ 0xAA.
- 5) Настройте 3 устройства (ведущий и 2 ведомых) с адресацией:
- Ведущий отправляет "ADDR1" или "ADDR2",
 - Только выбранное устройство отвечает "ACK".

Лабораторная работа 5. Системы команд процессорного ядра ARM Cortex.

Перечень заданий и вопросов для защиты лабораторной работы:

1) Какие три основных типа инструкций поддерживаются в ARM Cortex-M? Приведите по одному примеру для каждого типа.

2) Какие режимы адресации поддерживаются в инструкции LDR (на примере Cortex-M)? Приведите примеры.

3) Как реализуется условное выполнение инструкций в ARM Cortex-M? Приведите пример инструкции ADD с условием EQ.

4) Чем отличается Thumb-2 от классического Thumb? Какие преимущества дает эта технология?

5) Какие регистры отвечают за обработку прерываний в ARM Cortex-M? Как процессор определяет адрес ISR?

6) Как выполняется системный вызов (SVC) в ARM Cortex-M? Опишите шаги от вызова SVC #0x12 до перехода в обработчик.

7) Какие DSP-инструкции доступны в Cortex-M4/M7? Приведите примеры операций для сигнальных процессоров.

Лабораторная работа 6. Функции и режимы работы портов ввода/вывода общего назначения.

Перечень заданий и вопросов для защиты лабораторной работы:

1) Какие основные функции выполняют GPIO-порты в микроконтроллерах? Приведите примеры использования.

2) Какие режимы работы GPIO поддерживаются в STM32? Чем отличается push-pull от open-drain?

3) Зачем нужны внутренние подтягивающие резисторы? Как их включить в STM32 и AVR?

4) Как настроить вывод PA5 в режим вывода (push-pull) с максимальной скоростью в STM32? Приведите код на HAL.

5) Как назначить вывод PA9 для работы в режиме UART1_TX в STM32?

6) Какие методы устранениядребезга кнопки существуют? Приведите пример программной реализации.

Лабораторная работа 7. Прерывания и события.

Перечень заданий и вопросов для защиты лабораторной работы:

1) Что такое прерывание в микроконтроллере? Чем отличается аппаратное прерывание от программного (например, SWI)?

2) Как микроконтроллер определяет, какой обработчик прерывания (ISR) нужно выполнить? Что такое векторная таблица прерываний?

3) Как работают приоритеты прерываний в ARM Cortex-M? Что такое подпрерывания (nested interrupts)?

4) Как настроить прерывание по нажатию кнопки на выводе PA0 в STM32? Приведите код инициализации EXTI и обработчика.

5) Как защитить данные от повреждения при обработке прерываний? Что такое критическая секция?

6) Как настроить прерывание таймера TIM2 на частоту 1 кГц в STM32?

7) Чем события (Events) отличаются от прерываний? Когда их используют?

Лабораторная работа 8. Прерывания АЦП

Перечень заданий и вопросов для защиты лабораторной работы:

- 1) Что такое прерывание АЦП и в каких случаях его используют?
- 2) Как настроить АЦП в STM32 для работы с прерываниями? Приведите пример кода (HAL/CubeMX).
- 3) Как организовать прерывания для последовательного чтения 3 каналов АЦП (PA0, PA1, PA2) в STM32?
- 4) Какие ошибки АЦП можно обрабатывать через прерывания? Как их настроить?
- 5) Как уменьшить задержки при обработке прерываний АЦП? Какие методы использовать?
- 6) В чем преимущества и недостатки использования прерываний АЦП по сравнению с опросом флага (polling)?
- 7) Как использовать АЦП с DMA в STM32?
- 8) Как настроить Analog Watchdog для мониторинга границ напряжения?
- 9) Почему важно учитывать время установления АЦП?

Вопросы к комплексному заданию ТК3

1. Подсистема реального времени – модуль таймера/счетчика, назначение, функции.
2. Программирование модуля таймера/счетчика.
3. Программирование временных функций.
4. Режим «входного захвата» модуля таймера/счетчика.
5. Режим «входного захвата» модуля таймера/счетчика.
6. Режимы работы процессора событий.
7. Параметры системы команд процессора.
8. Классификация команд процессора.
9. Особенности программирования на языке C++.
10. Переменные, типы данных и преобразования языка программирования C++.

Для текущего контроля ТК4:

Проверяемая компетенция:

ПК-5 Способностью решать задачи по созданию и отладке программного обеспечения для микропроцессорных систем автоматизации и взаимодействия их модулей (ПК-5.1 Владеть навыками создания управляющих программ; ПК-5.3 Владеть методами отладки и тестирования программного обеспечения для микропроцессорных систем автоматизации)

Тест

Вопрос	Варианты ответа
Что делает оператор delete в C++?	освобождает память, выделенную через new
	удаляет файл

	закрывает программу
	удаляет переменную
Как объявить константу в C++?	let x = 5;
	const int x = 5;
	final int x = 5;
	#define x 5
Что делает ключевое слово virtual в C++?	запрещает переопределение метода
	позволяет переопределение метода в дочерних классах
	делает метод статическим
	удаляет метод
Характерная особенность системы команд RISC?	большое разнообразие команд
	фиксированная длина команд
	команды разной длины
	много сложных многоцикловых команд
Основное преимущество RISC-архитектуры?	большой набор сложных команд
	выполнение большинства команд за один такт
	переменная длина команд
	минимальное количество регистров
Какой элемент микроконтроллера отвечает за хранение адреса возврата при обработке прерываний?	регистр общего назначения
	стек
	кэш-память
	АЛУ
Какой оператор в C++ используется для работы с динамической памятью?	malloc()
	new
	alloc()
	create()
Какой способ адресации использует смещение относительно текущего значения счётчика команд?	прямая адресация
	непосредственная адресация
	относительная адресация
	косвенная адресация
Как объявить указатель на целое число в C++?	int * ptr = &x;
	int ptr = &x;
	pointer ptr = x;
	Int ptr = *x;
Какой заголовочный файл нужен для работы с прерываниями в embedded C++?	<iostream>
	<stdio.h>
	<avr/interrupt.h>
	<interrupt>
Какой режим таймера используется для измерения временных интервалов?	режим счётчика
	режим ШИМ
	режим захвата
	режим сравнения
Какая директива в C++ используется для подключения заголовочных файлов?	#import
	#include
	#require
	#attach
Какой тип порта ввода-вывода позволяет передать несколько бит одновременно?	последовательный
	параллельный
	SPI
	I2C
Какой элемент таймерного модуля хранит текущее значение счёта?	регистр сравнения
	счётный регистр
	регистр управления
	регистр делителя
Какой оператор C++	bit()

используется для работы с битами?	<code>bitwise()</code>
	битовые операторы (&, , ^)
	<code>bittest()</code>

Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре разработчика и содержит *50 тестовых вопросов на каждую компетенцию, из них 20% - закрытого типа, 80% - открытого типа.*

Практическое занятие 12. Прерывания USART/UART.

Примеры задач:

1) Настроить USART1 на приём строки, заканчивающейся символом '\n', с использованием прерываний. При получении строки выводить её обратно в терминал.

2) Настроить USART2 на:

- Приём команды "ON" или "OFF" через прерывания.
- При "ON" включать светодиод (PC13) и отправлять "LED ON\r\n".
- При "OFF" — выключать и отправлять "LED OFF\r\n".

3) Настроить USART3 с обработкой ошибок (фрейм, переполнение). При ошибке мигать светодиодом (PA5) и отправлять "UART ERROR!\r\n".

4) Настроить USART1 с DMA для приёма данных в кольцевой буфер. При заполнении половины буфера отправлять "BUFFER HALF!\r\n", при полном — "BUFFER FULL!\r\n".

Практическое занятие 13. Интерфейсы физического уровня RS-485 и RS-232.

Примеры задач:

1) Настроить USART1 микроконтроллера STM32 для обмена данными через интерфейс RS-232 (с использованием преобразователя уровня, например, MAX232). Параметры:

- Скорость: 9600 бод,
- 8 бит данных, 1 стоп-бит, без контроля чётности.

Требования:

- Реализовать передачу строки "RS-232 Test" каждые 2 секунды.
- Принять ответ от ПК (например, через терминал) и отобразить его на LCD.

2) Настроить USART2 + управление линией DE/RE (направление передачи) для работы с RS-485. Реализовать:

- Передачу команды "GET_DATA" на устройство-ведомый.
- Приём ответа "DATA:XX" и вывод в терминал.

Схема подключения:

- USART2_TX → MAX485 DI,
- USART2_RX ← MAX485 RO,
- PA1 → MAX485 DE/RE (управление направлением).

3) Создать сеть из 3 устройств (1 ведущий, 2 ведомых) с адресацией:

- Ведущий отправляет "ADDR1:CMD" или "ADDR2:CMD",

- Только устройство с совпадающим адресом отвечает "ACK".
Алгоритм:
- Ведущий отправляет сообщение с адресом.
- Ведомые сравнивают адрес и реагируют, если он совпадает.
- 4) Реализовать механизм обнаружения коллизий в сети RS-485:
- При одновременной передаче от нескольких устройств фиксировать ошибку.
- Отправлять повторный запрос через случайную задержку (алгоритм CSMA/CA).

Практическое занятие 14. Контроллер прямого доступа к памяти (DMA). Режимы работы.

Примеры задач:

1) Настроить DMA для передачи массива `uint8_t data[] = "DMA TEST"` через USART1 без участия CPU. Требования:

- Использовать нормальный режим (не циклический).
- После передачи вывести сообщение "Transfer Complete" через прерывание DMA.

2) Настроить DMA в циклическом режиме для непрерывного чтения АЦП (канал ADC1) в массив `adc_buffer[32]`. При заполнении половины буфера отправлять "HALF", при полном — "FULL".

3) Настроить DMA для приёма данных с USART2 в кольцевой буфер `rx_buffer[64]`. При обнаружении символа '\n' выводить строку в терминал.

4) Настроить DMA для АЦП с двойным буфером (`adc_buf1[32]` и `adc_buf2[32]`). При заполнении первого буфера — обрабатывать данные, при заполнении второго — активировать прерывание.

Практическое занятие 15. Прерывания и события DMA.

Примеры задач:

1) Настроить DMA для копирования массива `src[100]` в `dst[100]` с генерацией прерывания по завершению. При срабатывании прерывания зажигать светодиод (PC13).

2) Настроить DMA для АЦП с двойным буфером (`buf1[50]` и `buf2[50]`). При заполнении каждого буфера выводить в UART сообщение: "Buffer 1 Ready" или "Buffer 2 Ready".

3) Настроить DMA для приёма данных через USART1 в буфер `rx_buf[64]`. При половинном заполнении (32 байта) отправлять "HALF", при полном — "FULL".

4) Настроить передачу данных из `array[200]` в USART2 через DMA. При возникновении ошибки (например, переполнение) мигать светодиодом (PA5) и перезапускать DMA.

Лабораторная работа 9. Режимы работы АЦП

Перечень заданий и вопросов для защиты лабораторной работы:

1) Какие основные режимы работы АЦП существуют? Кратко опишите каждый.

- 2) Как настроить АЦП в STM32 для одиночного преобразования на канале ADC1_IN0 (PA0)? Приведите код инициализации (HAL).
- 3) Как настроить АЦП в непрерывном режиме с передачей данных через DMA в массив adc_data[100]?
- 4) Как настроить АЦП для последовательного опроса 3 каналов (PA0, PA1, PA2) в режиме сканирования?
- 5) Какие источники триггеров могут запускать АЦП (например, от таймера)? Как это настроить?
- 6) Как использовать Analog Watchdog для отслеживания выхода напряжения за пределы диапазона (например, 1.0–2.0 В)?
- 7) Какие параметры влияют на скорость и точность АЦП? Как найти баланс между ними?

Лабораторная работа 10. Исследование основ RTC.

Перечень заданий и вопросов для защиты лабораторной работы:

- 1) Какие основные функции выполняет модуль RTC в микроконтроллерах? Приведите примеры использования.
- 2) Как настроить RTC в STM32 для работы с часами и календарем? Приведите пример кода (HAL).
- 3) Как настроить будильник RTC для срабатывания каждые 30 секунд? Как обработать прерывание?
- 4) Как обеспечить работу RTC при отключении основного питания? Какие регистры защищены от сброса?
- 5) Как выполнить калибровку RTC для компенсации неточности кварцевого резонатора?
- 6) Как прочитать текущее время и дату из RTC? Какие структуры данных используются?
- 7) Как настроить RTC для генерации периодических прерываний (например, каждые 5 секунд) для выхода из режима сна?

Лабораторная работа 11. Исследование режима работы генерации сигналов.

Перечень заданий и вопросов для защиты лабораторной работы:

- 1) Как настроить таймер (например, TIM1 в STM32) для генерации ШИМ-сигнала с частотой 1 кГц и коэффициентом заполнения 50% на канале CH1?
- 2) Как динамически изменить частоту ШИМ-сигнала (например, с 1 кГц на 2 кГц) без переинициализации таймера?
- 3) Как сгенерировать синусоидальный сигнал через ЦАП (DAC), используя DMA для передачи данных из массива?
- 4) Как сгенерировать одиночный импульс длительностью 100 мкс с помощью таймера?
- 5) Как использовать таймер в режиме Capture/Compare для генерации сигнала с изменяемой скважностью?
- 6) Как сгенерировать периодический сигнал (например, 1 Гц) с помощью RTC?

7) Как реализовать прямой цифровой синтез (DDS) для генерации сигналов произвольной формы?

Лабораторная работа 12. Исследование режима энергосбережения RTC
Перечень заданий и вопросов для защиты лабораторной работы:

- 1) Какие основные механизмы позволяют RTC работать в режиме энергосбережения? Как организовано резервное питание?
- 2) Как настроить RTC в STM32 для работы только от батарейки при отключении основного питания? Какие регистры нужно защитить?
- 3) Какие режимы энергосбережения микроконтроллера (Sleep, Stop, Standby) совместимы с работой RTC? Чем они отличаются?
- 4) Как настроить будильник RTC для вывода микроконтроллера из режима Stop/Standby?
- 5) Как калибровка RTC влияет на энергопотребление? Какие методы калибровки существуют?
- 6) Как измерить ток потребления RTC в режиме Standby? Какие факторы влияют на энергопотребление?
- 7) Какие оптимизации можно применить для снижения энергопотребления RTC?

Вопросы к комплексному заданию ТК4

1. Распознавание и обработка прерываний *RISC*-контроллера .
2. Подсистема ввода/вывода цифровых систем.
3. Организация ввода/вывода *RISC*-контроллера.
4. АЦП и динамическая индикация.
5. Структура, назначение таймерного модуля *RISC*-контроллера
6. Режим широтно-импульсной модуляции.
7. Программирование встроенного модуля таймера/счетчика.
8. Сторожевой таймер *RISC*-контроллера.
9. Особенности, преимущества языка C++. Директивы C++.
10. Интегрированные среды разработки микропроцессорных устройств. Этапы разработки и отладки программного обеспечения.

Для промежуточной аттестации в 8 –м семестре:

Вопросы к экзамену:

1. Способы адресации команд МП. Индексная адресация. Привести примеры.
2. Способы адресации команд МП. Относительная адресация. Привести примеры.
3. Системы тактирования микроконтроллера.
4. Стековая память микроконтроллера.(Структура, принцип работы стека. Указатель стека).
5. Подсистема памяти микропроцессорных систем.
6. Организация памяти микроконтроллера.
7. Разновидности памяти микроконтроллера.
8. Память *RISC*-контроллера.

9. Организация обработки прерываний в микропроцессорных системах.
10. Прерывания в микроконтроллерах. (Маскирование прерываний. Распознавание и обработка прерываний).
11. Распознавание и обработка прерываний *RISC*-контроллера .
12. Подсистема ввода/вывода цифровых систем.
13. Организация ввода/вывода *RISC*-контроллера.
14. АЦП и динамическая индикация.
15. Структура, назначение таймерного модуля *RISC*-контроллера
16. Режим широтно-импульсной модуляции.
17. Программирование встроенного модуля таймера/счетчика.
18. Сторожевой таймер *RISC*-контроллера.
19. Особенности, преимущества языка C++. Директивы C++.
20. Интегрированные среды разработки микропроцессорных устройств. Этапы разработки и отладки программного обеспечения.
21. Системы команд процессорного ядра *ARM Cortex*.
22. Функции и режимы работы портов ввода/вывода *RISC*-контроллера.
23. Программирование прерываний и событий в цифровых системах на *RISC*-контроллерах.
24. Программирование режимов работы АЦП на *RISC*-контроллерах.
25. Реализация АЦП в микроконтроллерах.
26. Исследование основ *RTC RISC*-контроллеров.
27. Формирование сигналов в цифровых системах на *RISC*-контроллерах.
28. Исследование режима энергопотребления *RTC*.
29. Программирование асинхронной последовательной передачи данных в цифровых системах.
30. Программирование синхронной последовательной передачи данных в цифровых системах.