

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«САХАЛИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для студентов по проведению
самостоятельных внеаудиторных работ
по дисциплине

ПМ.03 Участие в разработке приложений взаимодействия с интеллектуальными интегрированными системами

укрупненная группа:

09.00.00 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

**специальность: 09.02.08 Интеллектуальные
интегрированные системы**

базовый уровень подготовки

Квалификация: Техник по интеллектуальным интегрированным системам

Форма обучения: очная

УТВЕРЖДАЮ

Зам. директора по НМР



Е.Н. Ермолаева

«24» марта 2025г.

Разработчик: Перов И.А., преподаватель информационных дисциплин

Одобрено на заседании ПЦК
информационных дисциплин
Протокол № 8 от «21» марта 2025г.
Председатель ПЦК



О.Б. Куприяшина

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	2
2. САМОСТОЯТЕЛЬНЫЕ РАБОТЫ (РАЗДЕЛ 1. СЕТЕВЫЕ И ОБЛАЧНЫЕ ТЕХНОЛОГИИ).....	2
Самостоятельная работа №1.....	2
Самостоятельная работа №2.....	3
Самостоятельная работа №3.....	3
Самостоятельная работа №4.....	4
Самостоятельная работа №5.....	5
Самостоятельная работа №6.....	5
Самостоятельная работа №7.....	6
Самостоятельная работа №8.....	7
Самостоятельная работа №9.....	7
Самостоятельная работа №10.....	8
Самостоятельная работа №11.....	9
3. САМОСТОЯТЕЛЬНЫЕ РАБОТЫ (РАЗДЕЛ 2. РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ МИКРОКОНТРОЛЛЕРОВ)	10
Самостоятельная работа №12.....	10
Самостоятельная работа №13.....	10
Самостоятельная работа №14.....	11
Самостоятельная работа №15.....	12
Самостоятельная работа №16.....	12
Самостоятельная работа №17.....	13
4. КРИТЕРИИ ОЦЕНКИ САМОСТОЯТЕЛЬНЫХ РАБОТ	15
5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА И ИСТОЧНИКИ	16

1. ОБЩИЕ ПОЛОЖЕНИЯ

Самостоятельные работы по ПМ.03 направлены на углублённое изучение сетевых технологий, облачных сервисов, а также инструментов разработки приложений для микроконтроллеров. Каждая работа рассчитана на **2 академических часа** (кроме отдельных, где указано иначе). Форма отчёта – **презентация (PowerPoint/Google Slides) или письменный конспект** в зависимости от задания. Работа должна быть сдана преподавателю в электронном виде или представлена на занятии.

2. САМОСТОЯТЕЛЬНЫЕ РАБОТЫ (РАЗДЕЛ 1. СЕТЕВЫЕ И ОБЛАЧНЫЕ ТЕХНОЛОГИИ)

Самостоятельная работа №1

«Основные технологии сетей передачи данных» (2 часа)

(Тема 1.1)

Цель: изучить и систематизировать современные технологии передачи данных (Ethernet, Wi-Fi, GPON, LTE, 5G).

Задание:

1. Подготовить презентацию на 8–10 слайдов.
2. Включить разделы:
 - Проводные технологии: Ethernet (Fast, Gigabit), xDSL, GPON.
 - Беспроводные технологии: Wi-Fi (802.11a/b/g/n/ac/ax), Bluetooth, ZigBee, LoRa.
 - Сотовые сети: 2G–5G (скорости, задержки, применение).
 - Сравнительная таблица: скорость, дальность, область применения.
3. Привести примеры использования каждой технологии в интеллектуальных интегрированных системах.

Методические указания:

- Использовать учебник «Сети и телекоммуникации» под ред. Самуйлова К.Е.
- На слайдах размещать схемы, логотипы стандартов, минимум текста.
- В конце презентации – список источников (не менее 3).

Форма отчёта:

Файл презентации .pptx или PDF. На первом слайде – тема, ФИО, группа.

Контрольные вопросы:

1. Чем отличается GPON от Ethernet PON?
2. Какая технология обеспечивает наименьшую задержку для промышленного IoT?
3. Что такое MIMO в Wi-Fi?

Самостоятельная работа №2

«Стандартизирующие организации в области телекоммуникаций» (2 часа)
(Тема 1.1)

Цель: ознакомиться с деятельностью международных организаций, разрабатывающих стандарты связи (ITU, IEEE, IETF, ETSI, 3GPP).

Задание:

1. Подготовить презентацию (8–10 слайдов), в которой для каждой организации указать:
 - полное название, год основания, штаб-квартира;
 - основные области стандартизации;
 - примеры известных стандартов (например, IEEE 802.3, IETF RFC 791).
2. Показать взаимосвязь: какая организация отвечает за физический уровень, какая – за протоколы интернета.
3. Добавить слайд «Российские аналоги» (Росстандарт, ТКС).

Методические указания:

- Использовать официальные сайты (itu.int, ieee.org, ietf.org).
- На слайдах разместить логотипы.
- Упомянуть процесс утверждения стандарта (RFC, Recommendation).

Форма отчёта:

Презентация. Обязательно указать даты последних версий стандартов.

Контрольные вопросы:

1. Чем отличается стандарт de facto от de jure?
2. Какую роль играет IETF в развитии интернета?
3. Какой стандарт определяет работу Ethernet-коммутаторов?

Самостоятельная работа №3

«Топология сетей, маршрутизация и коммутация» (1 час)
(Тема 1.2)

Цель: изучить основные топологии (звезда, кольцо, шина, ячеистая), принципы коммутации и маршрутизации.

Задание:

1. Подготовить презентацию (6–8 слайдов):
 - Топологии: схемы, преимущества, недостатки, примеры.
 - Коммутация: таблица MAC-адресов, широковещательный домен.
 - Маршрутизация: таблица маршрутизации, статическая vs динамическая (RIP, OSPF).

2. Привести пример сети здания с указанием топологии и используемого оборудования.

Методические указания:

- Использовать Packet Tracer для создания иллюстраций (скриншоты).
- Объяснить, как коммутатор узнаёт MAC-адреса (flooding, learning).

Форма отчёта:

Презентация + краткое описание примера.

Контрольные вопросы:

1. Почему в современных сетях используется топология «звезда»?
2. Что такое «петля» (loop) и как STP её предотвращает?
3. Как маршрутизатор выбирает путь при наличии нескольких маршрутов?

Самостоятельная работа №4

«Стек протоколов ISO/OSI, TCP/IP, IEEE 802» (2 часа)

(Тема 1.2)

Цель: сравнить эталонную модель OSI, стек TCP/IP и модель IEEE 802, понять соответствие уровней.

Задание:

1. Подготовить презентацию (10 слайдов):
 - Уровни OSI (7 уровней) с примерами протоколов и устройств.
 - Уровни TCP/IP (4 уровня) – соответствие уровням OSI.
 - Семейство IEEE 802 (802.3, 802.11, 802.1Q) и их место в моделях.
2. Создать сравнительную таблицу «Уровень – протокол – устройство – PDU».
3. Показать инкапсуляцию данных на примере HTTP-запроса.

Методические указания:

- Использовать рисунки инкапсуляции (данные → сегмент → пакет → кадр → биты).
- Взять пример из учебника Самуйлова.

Форма отчёта:

Презентация. На последнем слайде – ответ на вопрос: «Почему модель OSI осталась теоретической, а TCP/IP стал практическим стандартом?».

Контрольные вопросы:

1. Какой уровень OSI отвечает за сквозную доставку (end-to-end)?
2. Что такое PDU для канального уровня?
3. Какой протокол работает на сеансовом уровне?

Самостоятельная работа №5

«История развития облачных технологий» (2 часа)
(Тема 1.5)

Цель: изучить эволюцию облачных вычислений от концепции utility computing до современных сервисов.

Задание:

1. Подготовить презентацию (8–10 слайдов) с хронологией:
 - 1960-е: идея «компьютер как услуга» (Дж. Маккарти).
 - 1990-е: виртуализация, [Salesforce.com](https://www.salesforce.com) (SaaS).
 - 2006: появление Amazon EC2 и S3 (IaaS).
 - 2008: Google App Engine (PaaS).
 - 2010-е: контейнеризация (Docker), serverless (AWS Lambda).
2. Упомянуть ключевых игроков (AWS, Azure, Google Cloud, Yandex Cloud).
3. Добавить временную ленту (график).

Методические указания:

- Использовать статьи на Habr, официальные блоги провайдеров.
- На слайдах использовать иконки и фотографии основателей.

Форма отчёта:

Презентация. Указать не менее 5 ключевых событий с датами.

Контрольные вопросы:

1. Почему именно Amazon, а не IBM, запустила первый массовый IaaS?
2. Что такое «гипервизор» и как он связан с облаками?
3. Какую роль сыграла контейнеризация в развитии облаков?

Самостоятельная работа №6

«Классификация облачных систем: частные, публичные, гибридные облака» (2 часа)
(Тема 1.5)

Цель: изучить модели развертывания облаков, их преимущества и недостатки.

Задание:

1. Подготовить презентацию (8–10 слайдов):
 - Публичное облако (public): примеры, кому подходит, риски.
 - Частное облако (private): on-premises, управляемый хостинг.
 - Гибридное облако (hybrid): сочетание, сценарии (bursting, отказоустойчивость).
 - Мультиоблако (multi-cloud) – отличие от гибридного.

2. Сравнительная таблица по 5 критериям (стоимость, безопасность, контроль, масштабируемость, сложность).
3. Пример: государственная организация выбирает частное облако – почему?

Методические указания:

- Использовать материалы NIST SP 800-145 (определение облачных вычислений).
- Привести реальные кейсы: Netflix (публичное), банк (частное), Siemens (гибридное).

Форма отчёта:

Презентация. В конце – вывод, какая модель подходит для стартапа, а какая – для госучреждения.

Контрольные вопросы:

1. Может ли частное облако быть размещено вне дата-центра компании?
2. Что такое «облачный мост» (cloud bridge)?
3. В чем проблема vendor lock-in при мультиоблаке?

Самостоятельная работа №7

«Модели развертывания облачных систем» (2 часа)
(Тема 1.5)

Цель: разобрать различные способы развертывания облачной инфраструктуры: on-premises, колокация, managed hosting, hybrid.

Задание:

1. Презентация (6–8 слайдов) с описанием каждой модели:
 - On-premises: полный контроль, капитальные затраты.
 - Колокация: аренда стойки, своё оборудование.
 - Managed hosting: аренда сервера с обслуживанием.
 - Облачные модели (IaaS/PaaS/SaaS) как разновидности развертывания.
2. Сравнить по CAPEX/OPEX, времени развертывания, ответственности за обновления.
3. Привести диаграмму уровней ответственности (пользователь vs провайдер).

Методические указания:

- Использовать модель разделения ответственности (shared responsibility model) AWS/Azure.
- Пример: компания выбирает managed hosting для ERP – аргументы.

Форма отчёта:

Презентация. Добавить слайд-шпаргалку для выбора модели в зависимости от бизнес-задачи.

Контрольные вопросы:

1. Чем отличается колокация от managed hosting?
2. Кто отвечает за безопасность ОС в IaaS?
3. В какой модели развертывания максимальные CAPEX?

Самостоятельная работа №8

«SaaS – программное обеспечение как услуга» (2 часа)

(Тема 1.5)

Цель: изучить особенности SaaS-модели, примеры, преимущества и недостатки.

Задание:

1. Презентация (8–10 слайдов):
 - Определение, характерные черты (многопользовательская аренда, подписка, доступ через браузер).
 - Примеры: Google Workspace, Salesforce, 1С:Облако, Zoom.
 - Архитектура: общая база данных, изоляция данных на уровне приложения.
 - Преимущества: автоматическое обновление, доступ с любого устройства.
 - Недостатки: привязка к провайдеру, кастомизация ограничена.
2. Рассмотреть метрики: ARPU, Churn rate, MRR.
3. Привести пример расчета TCO для перехода с on-premises на SaaS.

Методические указания:

- Использовать данные исследования Gartner по рынку SaaS.
- Упомянуть API-интеграции (REST, webhooks).

Форма отчёта:

Презентация. Добавить график роста рынка SaaS.

Контрольные вопросы:

1. Как в SaaS обеспечивается изоляция данных разных клиентов?
2. Что такое SSO (Single Sign-On) и почему важно для SaaS?
3. Чем отличается SaaS от настольного ПО с подпиской?

Самостоятельная работа №9

«PaaS – платформа как услуга» (2 часа)

(Тема 1.5)

Цель: изучить платформенные сервисы для разработки и развертывания приложений (Heroku, Google App Engine, Yandex Cloud Platform).

Задание:

1. Презентация (8–10 слайдов):
 - Определение PaaS, что входит в платформу (среда исполнения, базы данных, инструменты разработки).

- Примеры провайдеров и их особенностей.
 - Поддержка языков (Java, Python, Node.js, .NET).
 - Автомасштабирование, балансировка нагрузки.
 - Сравнение с IaaS: уровень абстракции, скорость развертывания.
2. Описать процесс развертывания приложения на Heroku (git push).
 3. Упомянуть контейнерные PaaS (Red Hat OpenShift, Google Kubernetes Engine).

Методические указания:

- Зарегистрироваться в бесплатном tier Heroku (или аналог) и сделать скриншот успешного деплоя.
- Показать файл Procfile и requirements.txt.

Форма отчёта:

Презентация + скриншоты работающего приложения в PaaS.

Контрольные вопросы:

1. Почему PaaS не подходит для приложений с жесткими требованиями к аппаратному обеспечению?
2. Как PaaS решает проблему «у меня работает, а на сервере нет»?
3. Что такое «buildpack» в Heroku?

Самостоятельная работа №10

«IaaS – инфраструктура как услуга» (2 часа)

(Тема 1.5)

Цель: изучить базовые компоненты IaaS (виртуальные машины, сети, хранилища), сравнить крупнейших провайдеров.

Задание:

1. Презентация (8–10 слайдов):
 - Определение IaaS, основные ресурсы: compute, storage, networking.
 - Примеры: AWS EC2, Google Compute Engine, Yandex Compute Cloud.
 - Типы виртуальных машин (общего назначения, с ускорением GPU, высокопроизводительные).
 - Блочные хранилища (EBS, persistent disks) и объектные (S3).
 - Сетевые компоненты: VPC, подсети, маршрутизаторы, балансировщики.
2. Сравнить три провайдера по цене за vCPU/GB RAM, наличию бесплатного уровня.
3. Привести пример создания VM через консоль или CLI.

Методические указания:

- Использовать актуальные прайс-листы (апрель 2025).
- В презентации показать скриншот консоли управления.

Форма отчёта:

Презентация. Добавить таблицу сравнения с ценами (в рублях/долларах).

Контрольные вопросы:

1. Что такое «спотовые инстансы» и зачем они нужны?
2. Как обеспечить отказоустойчивость ВМ в IaaS?
3. Чем отличается объектное хранилище от блочного?

Самостоятельная работа №11**«Обзор существующих облачных систем» (2 часа)**

(Тема 1.5)

Цель: провести обзор и сравнение глобальных и российских облачных платформ.

Задание:

1. Подготовить презентацию (10–12 слайдов), в которой рассмотреть:
 - AWS (Amazon) – лидер рынка, основные сервисы (EC2, S3, Lambda, RDS).
 - Microsoft Azure – интеграция с Windows, Active Directory.
 - Google Cloud – BigQuery, AI/ML, Kubernetes.
 - Yandex Cloud – возможности, сертификация ФСТЭК.
 - VK Cloud (MCS) – особенности.
 - Selectel, RuCloud.
2. Сравнить по географическому присутствию, наличию российских дата-центров, соответствию 152-ФЗ.
3. Выбрать одну платформу для гипотетического проекта (интернет-магазин) и обосновать.

Методические указания:

- Использовать официальные сайты и сравнительные обзоры (CNews, Habr).
- Добавить карту расположения центров обработки данных.

Форма отчёта:

Презентация. В конце – рекомендация для образовательных целей (бесплатные гранты).

Контрольные вопросы:

1. Какие сервисы AWS подходят для хранения больших данных?
2. Почему российские компании предпочитают Yandex Cloud или VK Cloud?
3. Что такое «импортозамещение» в облаках?

3. САМОСТОЯТЕЛЬНЫЕ РАБОТЫ (РАЗДЕЛ 2. РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ МИКРОКОНТРОЛЛЕРОВ)

В рабочей программе для раздела 2 (МДК.03.02) самостоятельные работы детально не расписаны, но предусмотрены часы (50 часов). Ниже приведены рекомендуемые темы и задания, соответствующие содержанию модуля.

Самостоятельная работа №12

«Изучение документации и настройка среды разработки (STM32CubeIDE / Arduino IDE)» (4 часа)

Цель: самостоятельно установить и настроить среду разработки для целевого микроконтроллера, изучить структуру проекта.

Задание:

1. Скачать и установить STM32CubeIDE (или Arduino IDE, если используется Arduino).
2. Создать пустой проект для своей платы (STM32F103C8T6 или Arduino Uno).
3. Изучить сгенерированные файлы: `.ioc`, `main.c`, `stm32f1xx_hal_msp.c`, `startup_stm32f103xb.s`.
4. Настроить тактовую частоту на максимальную (72 МГц для STM32).
5. Настроить отладчик (ST-Link) и убедиться, что прошивка загружается.
6. Написать простую программу мигания светодиодом, скомпилировать и загрузить.

Методические указания:

- Использовать видеоуроки и официальную документацию STMicroelectronics.
- Для Arduino: установить поддержку платы через менеджер плат.

Форма отчёта:

Скриншоты рабочей среды, листинг кода, фотография мигающего светодиода. Ответить на вопрос: «Какие файлы нужно добавить в `.gitignore`?».

Контрольные вопросы:

1. Как изменить частоту ядра в CubeMX?
2. Для чего нужен файл `.ioc`?
3. Как подключить внешнюю библиотеку (например, FatFS)?

Самостоятельная работа №13

«Разработка блок-схемы алгоритма для опроса датчиков» (2 часа)

Цель: научиться проектировать логику программы до написания кода, используя блок-схемы или диаграммы состояний.

Задание:

1. Выбрать сценарий: опрос датчика температуры DS18B20 каждые 5 секунд, отправка данных по UART, при превышении порога – включение светодиода.
2. Разработать блок-схему алгоритма (в любом редакторе: draw.io, Visio, Lucidchart).
3. Включить блоки: инициализация, чтение датчика, проверка условия, действие, задержка.
4. Дополнительно: диаграмма состояний конечного автомата (ожидание, измерение, отправка).
5. По блок-схеме написать псевдокод.

Методические указания:

- Использовать стандартные символы блок-схем (ромб – условие, прямоугольник – действие, параллелограмм – ввод/вывод).
- Для диаграммы состояний – кружки и стрелки.

Форма отчёта:

Файл с блок-схемой (PNG/PDF) + псевдокод. Вывод о том, помогает ли визуализация избежать ошибок.

Контрольные вопросы:

1. Чем блок-схема отличается от диаграммы деятельности UML?
2. Как изобразить прерывание на блок-схеме?
3. Почему для микроконтроллеров часто используют конечные автоматы?

Самостоятельная работа №14

«Изучение основ Git для контроля версий прошивок» (4 часа)

Цель: освоить базовые команды Git для управления версиями встраиваемого ПО.

Задание:

1. Установить Git, настроить `user.name` и `user.email`.
2. Создать локальный репозиторий для папки проекта.
3. Добавить файлы, сделать коммит.
4. Внести изменения в код (добавить второй светодиод), сделать второй коммит.
5. Создать ветку `feature/pwm`, переключиться, реализовать ШИМ-управление светодиодом, закоммитить.
6. Слить ветку с `main`.
7. Создать удалённый репозиторий на GitHub, отправить туда все коммиты.
8. Написать `.gitignore` для исключения временных файлов (`.o`, `.elf`, `.d`, `Debug/`).

Методические указания:

- Использовать Git Bash или встроенный терминал VS Code.
- Для STM32CubeIDE типовой `.gitignore` можно взять с GitHub.

Форма отчёта:

Скриншоты вывода `git log --oneline --graph`, ссылка на репозиторий. Ответить: «Как откатить изменения до предыдущего коммита?».

Контрольные вопросы:

1. Что такое `git stash` и когда его применяют?
2. Как разрешить конфликт при слиянии?
3. Зачем нужны теги (`tags`) в репозитории?

Самостоятельная работа №15

«Анализ примеров кода для работы с FreeRTOS» (4 часа)

Цель: изучить основы многозадачности на примере FreeRTOS, запустить пример с двумя задачами.

Задание:

1. Скачать пример FreeRTOS для своей платы (STM32CubeIDE включает FreeRTOS в качестве middleware).
2. Изучить документацию по API: `xTaskCreate`, `vTaskDelay`, очереди, семафоры.
3. Создать два задачи:
 - Задача 1: мигает светодиодом с периодом 500 мс.
 - Задача 2: отправляет в UART строку "Hello from task2" каждые 2 секунды.
4. Собрать проект, прошить, убедиться в параллельной работе.
5. Добавить очередь: задача 2 отправляет счётчик нажатий кнопки, задача 1 выводит его в UART.

Методические указания:

- Использовать примеры из папки `Middlewares/Third_Party/FreeRTOS`.
- Настроить конфигурацию FreeRTOS (конфигурационный файл `FreeRTOSConfig.h`).

Форма отчёта:

Листинг кода с пояснениями, скриншот вывода UART, описание приоритетов задач.

Контрольные вопросы:

1. Чем отличается `vTaskDelay` от `vTaskDelayUntil`?
2. Что такое «приоритет задачи» и как он влияет на планирование?
3. Как синхронизировать задачи с помощью двоичного семафора?

Самостоятельная работа №16

«Подготовка реферата «Сравнение протоколов MQTT и CoAP для IoT» (4 часа)

Цель: сравнить два популярных протокола для передачи данных в Интернете вещей.

Задание:

1. Написать реферат объемом 5–7 страниц (шрифт 14, Times New Roman, 1.5 интервал).
2. Структура:
 - Введение (актуальность IoT-протоколов).
 - Обзор MQTT: архитектура (брокер, клиент), QoS, retain-сообщения.
 - Обзор CoAP: протокол поверх UDP, методы (GET, POST), observe.
 - Сравнение по критериям: надежность, заголовки, работа в ограниченных сетях, использование в LoRaWAN.
 - Примеры применения: MQTT – умный дом, CoAP – сенсорные сети.
3. В конце – таблица сравнения и вывод о целесообразности выбора для конкретной задачи.

Методические указания:

- Использовать спецификации MQTT 5.0 и CoAP (RFC 7252).
- Привести схемы обмена сообщениями.

Форма отчёта:

Реферат в формате .docx или .pdf. Обязательна ссылка на первоисточники.

Контрольные вопросы:

1. Как MQTT обеспечивает доставку сообщений при обрыве соединения?
2. Почему CoAP использует UDP, а не TCP?
3. Какой протокол лучше подходит для батарейных датчиков?

Самостоятельная работа №17

«Создание чек-листа для отладки микроконтроллерного приложения» (2 часа)

Цель: систематизировать шаги по поиску типовых ошибок в прошивке.

Задание:

1. Составить чек-лист из 20–30 пунктов, разделённых на категории:
 - Аппаратные проблемы (питание, тактирование, соединения).
 - Конфигурация периферии (такты, пины, альтернативные функции).
 - Логические ошибки (переполнение таймеров, неправильная обработка прерываний).
 - Работа с памятью (переполнение стека, утечки, неинициализированные переменные).
 - Взаимодействие с датчиками (тайминги, протоколы).
2. К каждому пункту добавить способ проверки (осциллограф, логический анализатор, отладчик, вывод в UART).
3. Оформить чек-лист в виде таблицы или интеллект-карты.

Методические указания:

- Взять за основу реальные проблемы, возникавшие при выполнении практических работ.
- Добавить пункт «проверить, не сбросился ли микроконтроллер из-за watchdog».

Форма отчёта:

Файл с чек-листом (Excel, Word или PDF). В конце – пример успешного использования чек-листа (кейс).

Контрольные вопросы:

1. Как проверить, работает ли внешний кварцевый резонатор?
2. Что делать, если программа зависает в обработчике прерывания?
3. Как обнаружить переполнение стека в RTOS-задаче?

4. КРИТЕРИИ ОЦЕНКИ САМОСТОЯТЕЛЬНЫХ РАБОТ

Оценка	Требования
5 (отлично)	Задание выполнено полностью, презентация/реферат/код оформлены аккуратно, содержание полное, есть иллюстрации и выводы. Студент может устно пояснить любой слайд или фрагмент.
4 (хорошо)	Задание выполнено, но есть 1-2 недочёта (отсутствует список источников, мелкие ошибки в терминологии, неполная таблица).
3 (удовлетворительно)	Задание выполнено частично (не менее 60%), презентация содержит только текст без схем, реферат поверхностный.
2 (неудовлетворительно)	Задание не выполнено или представлен материал, не соответствующий теме.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА И ИСТОЧНИКИ

Основные (из РП):

1. Зверева В.П., Назаров А.В. Сопровождение и обслуживание программного обеспечения компьютерных систем. – М.: Академия, 2020.
2. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем. – М.: Академия, 2020.
3. Гуров В.В. Микропроцессорные системы. – М.: ИНФРА-М, 2022.
4. Огнева М.В., Кудрина Е.В. Программирование на языке C++. – М.: Юрайт, 2021.
5. Самуйлов К.Е. и др. Сети и телекоммуникации. – М.: Юрайт, 2021.

Дополнительные:

- Документация STM32CubeIDE (STMicroelectronics).
- FreeRTOS Reference Manual.
- MQTT Specification (OASIS).
- CoAP RFC 7252.
- Материалы курса «Облачные технологии» на [Stepik.org](https://stepik.org).
- Cisco Packet Tracer – справочные руководства.