

2025

ПРОФЕССИОНАЛЬНАЯ КВАЛИФИКАЦИЯ

3513-0-002 ПРОГРАММИСТ
МИКРОКОНТРОЛЛЕРОВ (УРОВЕНЬ 5)

1 Профессиональные квалификации

3513-0-002 Программист микроконтроллеров (Уровень 5)

№	Тип	Свойство
1	Код профессии	3513-0-002
2	Название профессии	Программист микроконтроллеров
3	Вид трудовой деятельности	Разработка микропрограммного обеспечения для микроконтроллеров, автоматизированных систем управления различной электронной техники
4	Уровень НРК/ОРК	5
5	Трудовая функция	1. Программирование цифровых устройств на базе микроконтроллеров Функционирование схемы, рефакторинг программного кода ПО
6	Профессиональная задача	1. Разработка программного кода ПО для микроконтроллеров 2. Отладка программного кода ПО для микроконтроллеров 3. Проверка работоспособности схемы контроллера и программного кода ПО 4. Коррекция схемы МК и рефакторинг программного кода ПО
7	Умения	1. Определить требования к микроконтроллеру 2. Разработка алгоритма управления 3. Выбирать тип микроконтроллера 4. Разработка структуры аппаратных и программных средств 5. Применять технологию программирования 6. Осуществлять анализ и проверку исходного программного кода ПО 7. Осуществлять совместную отладку аппаратных средств и программного кода ПО в режиме реального времени 8. Воспроизведение дефектов программного кода ПО 9. Установка причин возникновения дефектов программного кода ПО 10. Коррекция схем и топологии микроконтроллеров 11. Устранение выявленных дефектов в программном коде ПО 12. Формирование отчетной документации по результатам проведенных работ 13. Проводить внешний осмотр микроконтроллера 14. Проверять работоспособность микроконтроллер специальными инструментами 15. Применять методы и средства проверки работоспособности программного кода ПО 16. Разрабатывать и оформлять контрольные примеры для проверки работоспособности программного кода ПО 17. Интерпретировать и диагностировать данные программного кода ПО 18. Анализировать значения, полученных характеристик программного кода ПО 19. Документировать результаты проверки работоспособности программного кода ПО 20. Проводить внешний осмотр и диагностику микроконтроллера 21. Выявлять и заменять неисправные части микроконтроллера 22. Применять методы, средств для рефакторинга и оптимизации программного кода ПО 23. Применять методов и приемов отладки дефектов программного кода ПО 24. Использовать эксплуатационную документацию по разработке ПО

8	Знания	<ol style="list-style-type: none"> 1. Основные характеристики современных микроконтроллеров 2. Алгоритмизацию и структуры данных 3. Языки программирования и базы данных 4. Требования к разработке ПО 5. Жизненный цикл ПО 6. Современные средства разработки ПО для микроконтроллеров 7. Методы и средства совместной отладки аппаратных и программных средств 8. Знание современных языков программирования 9. Библиотек для использования в той или иной задаче 10. Процессы отладки 11. Иметь представление о ИБ 12. Основные характеристики современных микроконтроллеров 13. Средства и способы проверки работоспособности микроконтроллеров 14. ПО и его функциональные возможности 16. Методы и средства сборки модулей и компонентов ПО 17. Методы создания и документирования контрольных примеров и тестовых наборов данных ПО 18. Правила, алгоритмы и технологии создания тестовых наборов данных ПО 19. Требования к структуре и форматам хранения тестовых наборов данных ПО 20. Основные характеристики современных микроконтроллеров 21. Методы и средства рефакторинга и оптимизации программного кода ПО 22. Методы и приемы отладки программного кода ПО 23. Инструменты отладки программного кода ПО
---	--------	--

№	Тип	Свойство
1	Теоретический этап	<ol style="list-style-type: none"> 1. Тестирование (закрытые вопросы с множественным выбором ответа) 2. Собеседование (открытые вопросы и дискуссия на профессиональные вопросы) 3. Решение расчетных задач (проверка способности выполнять инженерные расчеты) 4. Анализ документации (проверка умения интерпретирования технической документации)
2	Практический этап	<ol style="list-style-type: none"> 1. Выполнение практического задания (написание программного кода для встраиваемой системы с использованием технического задания) 2. Выполнение проектного задания (написание программного кода для встраиваемой системы) 3. Анализ и доработка существующего проекта

1. Знание базовых законов электрических цепей (Ом, Кирхгофа)
2. Простейший анализ аналоговых и цифровых схем
3. Знание электрических компонентов - резисторы, конденсаторы, диоды, транзисторы, операционные усилители.
4. Умение читать принципиальные схемы и чертежи.
5. Использование САПР для проектирования печатных плат
6. Технологии монтажа компонентов

7. Базовые навыки пайки: ручная пайка, использование паяльных станций.
8. Использование мультиметров, осциллографов, источников питания
9. Проверка работоспособности схем
10. Основы программирования и алгоритмизации (C/C++)
11. Алгоритмическая работа с популярными платформами (Arduino, STM32, ESP32, PIC)
12. Разработка прошивок для управления электронными устройствами на базе Arduino
13. Знание стандартов IPC для производства и проверки плат
14. Работа с симуляторами для тестирования схем (LTspice)
15. Опыт работы с CAD/CAM

Экзамен на подтверждение квалификации состоит из теоретического и практического этапов общей продолжительностью 6 часов.

Теоретический этап (2 часа) включает тестирование (30 минут), собеседование (30 минут), решение расчетных задач (30 минут) и анализ документации (30 минут). Тестирование проверяет фундаментальные знания, собеседование оценивает способность аргументированно отвечать на профессиональные вопросы, расчетные задачи требуют инженерных вычислений, а анализ документации проверяет умение работать с техническими спецификациями.

Практический этап (4 часа) включает выполнение практического задания (2 часа), где необходимо написать программный код для встраиваемой системы по техническому заданию, и анализ с доработкой существующего проекта (2 часа), предполагающий исправление ошибок, оптимизацию кода и улучшение структуры проекта. Минимальный проходной балл – 70.

№	Тип	Свойство
1	Тестирование	<p>1. Какой набор команд используется в микроконтроллерах STM32? (x86/ARM Cortex-M/RISC-V/AVR)</p> <p>2. Какой интерфейс чаще всего используется для программирования STM32? (JTAG/SWD/SPI/UART/I2C)</p> <p>3. Как называется официальное программное обеспечение для разработки под STM32? (Keil uVision/STM32CubeIDE/Code Composer Studio/MPLAB X)</p> <p>4. Какой минимальный тактовый генератор можно использовать для запуска STM32? (16 МГц кварц/Встроенный HSI-генератор/Внешний RC-генератор/Без генератора не запустится)</p> <p>5. Какой разрядности ядро у микроконтроллеров серии STM32F4? (8 бит/16 бит/32 бит/64 бит)</p> <p>6. Какой интерфейс НЕ поддерживается в микроконтроллерах STM32? (CAN/USB/PCIe/I2C)</p> <p>7. Какая библиотека официально предоставляется для работы с периферией STM32? (CMSIS/HAL/Arduino Core/MicroPython)</p> <p>8. Какой тип памяти используется для хранения прошивки в STM32? (RAM/EEPROM/Flash/SD-карта)</p> <p>9. Как называется механизм управления тактированием периферии в STM32? (PLL/Clock Tree/RCC/Prescaler)</p> <p>10. Какое напряжение питания типично для большинства микроконтроллеров STM32? (1.8 В/3.3 В/5 В/12 В)</p> <p>11. Какой интерфейс можно использовать для обновления прошивки STM32 «по воздуху»? (USART/I2C/DFU/SWD)</p> <p>12. Какое назначение у вывода NRST? (Сброс питания/Сброс микроконтроллера/Задание тактовой частоты/Запуск режима программирования)</p> <p>13. Какой режим энергосбережения позволяет микроконтроллеру STM32 потреблять минимум энергии? (Sleep/Stop/Standby/Run)</p> <p>14. Какой из STM32 имеет встроенный Wi-Fi? (STM32F103/STM32H750/STM32WL/STM32WB)</p> <p>15. Какой интерфейс предпочтителен для связи STM32 с датчиками, работающими на длинных линиях? (SPI/I2C/UART/CAN)</p> <p>16. Какой модуль STM32 используется для генерации прерываний по таймеру? (ADC/DMA/NVIC/TIM)</p> <p>17. Какой интерфейс чаще всего используется для подключения внешней памяти к STM32? (SDIO/QSPI/I2C/SPI)</p> <p>18. Какой сигнал в UART STM32 указывает на готовность к передаче данных? (RX/TX/CTS/RTS)</p>

		<p>19. Какой протокол передачи данных обеспечивает максимальную скорость? (UART/I2C/SPI/USB)</p> <p>20. Какой режим DMA позволяет передавать данные из периферии в память без участия процессора? (Memory-to-Memory/Peripheral-to-Memory/Memory-to-Peripheral/Loopback)</p> <p>21. Какой из STM32 имеет встроенный Bluetooth? (STM32F4/STM32G0/STM32WB/STM32L0)</p> <p>22. Какое максимальное напряжение можно подавать на аналоговые входы ADC в STM32 при питании 3.3 В? (1.8 В/2.5 В/3.3 В/5 В)</p> <p>23. Какой модуль STM32 используется для работы с сенсорными кнопками? (TIM/DMA/ADC/TSI)</p> <p>24. Какой модуль STM32 отвечает за защиту от некорректных тактовых сигналов? (RCC/WWDG/IWDG/NVIC)</p> <p>25. Какой режим работы GPIO в STM32 используется для работы с внешним прерыванием? (Analog/Input/Output/Alternate Function)</p>
2	Собеседование	<p>1. Какие основные периферийные интерфейсы поддерживают микроконтроллеры STM32, и в каких случаях лучше использовать каждый из них?</p> <p>2. Как работает механизм прерываний в STM32, и как можно задать приоритеты для разных источников прерываний?</p> <p>3. Объясните, как настроить и использовать таймеры STM32 для генерации ШИМ-сигналов. Какие параметры важны?</p> <p>4. Как работает система тактирования в STM32, и какие источники тактовой частоты можно использовать? Как выбрать оптимальную конфигурацию?</p> <p>5. Какие методы энергосбережения поддерживают микроконтроллеры STM32, и как можно снизить энергопотребление устройства без потери функциональности?</p>
3	Решение расчетных задач	<p>1. В STM32 внешний кварцевый генератор работает на частоте 8 МГц. Если PLL умножает частоту в 4 раза, какая будет итоговая частота процессора?</p> <p>2. Нужно настроить таймер STM32 на генерацию сигнала с частотой 1 кГц. Если системная частота 16 МГц, какое значение нужно установить в регистр счетчика, если предделитель равен 16?</p> <p>3. Если скорость передачи данных по UART установлена на 115200 бод, сколько битов будет передаваться за 1 секунду?</p> <p>4. В спящем режиме микроконтроллер потребляет 5 мА. Сколько часов он проработает от батареи 500 мАч?</p> <p>5. Если к пину STM32, настроенному как выход, подключена светодиодная лампа, и в коде написано <code>GPIO_SetHigh(GPIOA, PIN5)</code>, что произойдет со светодиодом?</p>
4	Анализ документации	<p>1. В документации на STM32F103 указано несколько источников тактирования. Какие основные источники тактовой частоты поддерживает этот микроконтроллер, и какова максимальная частота процессора?</p>

	<p>2. В даташите на STM32L052 указаны разные режимы энергосбережения. Найдите в документации, какой режим обеспечивает наименьшее энергопотребление и сколько потребляет микроконтроллер в этом режиме.</p> <p>3. В референсном руководстве на STM32F103 найдите, какой регистр отвечает за настройку режима работы вывода (input, output, alternate function, analog) и как установить порт в режим альтернативной функции.</p> <p>4. В даташите на STM32L052 найдите, какое максимальное входное напряжение можно подавать на аналоговый вход АЦП при питании 3.3 В.</p> <p>5. В документации на STM32L052 найдите, поддерживает ли этот микроконтроллер встроенную EEPROM, и если да, какой объем памяти доступен для хранения данных?</p>
--	---

№	Тип	Свойство
1	Практическое задание	<p>Разработать прошивку для STM32F103C8T6, которая выполняет следующие функции:</p> <ol style="list-style-type: none"> 1. Отображение данных на I2C-семисегментном дисплее – микроконтроллер должен выводить текущее значение одного из потенциометров (0–100%). Переключение между потенциометрами осуществляется нажатием одной из кнопок. 2. Управление DC-моторами – два потенциометра регулируют скорость вращения двух двигателей. Один потенциометр управляет первым двигателем, второй – вторым. ШИМ используется для регулирования скорости. 3. Обработка кнопок – шесть кнопок выполняют различные функции: включение/выключение каждого двигателя (две кнопки), изменение направления вращения (еще две кнопки) и переключение отображаемых данных на дисплее (две кнопки). 4. Управление светодиодами – шесть светодиодов индицируют состояние системы: два показывают направление вращения двигателей, два – их активность, еще два используются для других функций (например, подтверждение нажатия кнопок). 5. Если кнопки не нажимались в течение 30 секунд, дисплей отключается для экономии энергии. <p>Использовать стандартную библиотеку HAL. Реализовать обмен данными с семисегментным дисплеем через I2C. Обработать нажатия кнопок с помощью внешних прерываний. Настроить АЦП в режиме периодического измерения с таймером. Использовать ШИМ для управления скоростью двигателей и яркостью светодиодов.</p>
2	Проектное задание	<p>Разработать прошивку для STM32L052, которая выполняет следующие функции:</p> <ol style="list-style-type: none"> 1. Отображение данных на I2C-дисплее – микроконтроллер должен выводить текущее значение потенциометра (0–100%) на OLED-дисплей, подключенный по I2C. 2. Работа с кнопками – две кнопки должны управлять яркостью RGB-светодиода: одна увеличивает, другая уменьшает (яркость регулируется в 10 шагов). 3. Измерение аналогового сигнала – потенциометр подключен к АЦП, и его значение считывается каждые 100 мс. 4. Управление RGB-светодиодом – каждый цвет (R, G, B) управляется через ШИМ, и интенсивность каждого канала регулируется в зависимости от положения потенциометра. 5. Если кнопки не нажимались в течение 30 секунд, экран должен выключаться для экономии энергии. <p>Использовать стандартную библиотеку HAL. Реализовать обмен данными с дисплеем через I2C. Обработать нажатия кнопок с помощью внешних прерываний. Настроить АЦП в режиме периодического измерения с таймером. Использовать ШИМ для управления яркостью светодиода.</p>
3	Анализ и доработка существующего проекта	<p>Вам предоставлен проект для STM32F103C8T6, который реализует управление системой с использованием двух DC-моторов, семисегментного дисплея I2C, шести светодиодов, шести кнопок и двух потенциометров. Проект выполняет следующие функции:</p> <ol style="list-style-type: none"> 1. Отображает данные на семисегментном дисплее I2C – текущее значение одного из потенциометров (0–100%). Переключение между потенциометрами

		<p>осуществляется нажатием одной из кнопок.</p> <ol style="list-style-type: none">2. Управляет двумя DC-моторами – скорость вращения каждого из моторов регулируется одним из потенциометров с использованием ШИМ.3. Обрабатывает нажатия кнопок – шесть кнопок выполняют следующие функции: включение/выключение каждого двигателя, изменение направления вращения двигателей, переключение между потенциометрами для отображения данных на дисплее.4. Управляет шестью светодиодами – два светодиода отображают направление вращения двигателей, два — их активность, два другие используются для индикации других состояний.5. Дисплей отключается через 30 секунд бездействия. <p>Задача:</p> <ol style="list-style-type: none">1. Оптимизировать текущий код, улучшив производительность системы.2. Добавить защиту от случайных нажатий кнопок с дебаунсом.3. Реализовать корректное переключение между моторами и потенциометрами с добавлением визуальной индикации на дисплее.4. Усовершенствовать управление светодиодами, чтобы они четко индицировали не только включение моторов, но и их направление.5. Настроить систему так, чтобы дисплей автоматически выключался в случае длительного отсутствия взаимодействия, а также добавьте режим энергосбережения для моторов, когда они не используются. <p>Используйте стандартную библиотеку HAL. Внесите необходимые изменения для улучшения стабильности работы системы и правильной работы всех функциональных блоков.</p>
--	--	---