

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
«ИНЖЕНЕР УМНЫХ СИСТЕМ»**

УТВЕРЖДАЮ

Генеральный директор
ООО «ИНУМСИС»
Семенов Д.А.
«15» августа 2024 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«ОПЕРАЦИОННАЯ СИСТЕМА FREERTOS»
Нормативный срок обучения: 16 часов
Категория обучающихся: программисты
Разработчик программы: Мединцев В.В.**

Рабочая программа учебного модуля 1. «Операционная система FreeRTOS»

**Учебно-тематические план рабочей программы учебного модуля 1.
«Операционная система FreeRTOS»**

№ пп	Наименование темы	Всего часов	В том числе		
			ТЗ	ПЗ	СРС
1.	Введение во FreeRTOS. Операционная система реального времени FreeRTOS. Лицензирование, состав, интеграция в проект.	2	1	0,5	0,5
2.	Управление задачами. Состояния задач. Приоритеты. Алгоритмы планирования.	3	2	0,5	0,5
3.	Управление очередями. Программные таймеры.	1,5	1	0,5	-
4.	Таймеры и управление кучей. Управление памятью. Программные таймеры.	1,5	1	0,5	-
5.	Ресурсы и прерывания. Двоичные и счетные семафоры. События. Группы событий. Нотификация.	2	1	0,5	0,5
6.	Оценка производительности. Отложенная обработка прерываний.	2	1	0,5	0,5
7.	Снижение энергопотребления. Управление питанием.	1,5	1	0,5	-
8.	Сложные случаи. Макросы. Стиль кода.	1	1	-	-

9.	Промежуточная аттестация	0,5	-	0,5	
	Итого	15	9	4	2

3. Содержание рабочей программы учебного модуля 1. «Операционная система FreeRTOS»

Тема 1.1. Введение во FreeRTOS. Операционная система реального времени FreeRTOS. Лицензирование, состав, интеграция в проект.

Теоретические занятия:

Операционная система FreeRTOS.

Многозадачность. Планирование. Переключение контекста. Приложения в реальном времени. Планирование в реальном времени. Структура кода. Создание собственного приложения. Конфигурационный файл.

Почему мы выбрали FreeRTOS.

Создание проекта. Создание задач.

Выводы.

Типы данных и именованые функций. Типы данных. Имена переменных. Имена функций.

Практические занятия:

Домашнее задание. В среде разработки STM32CubeIDE создайте проект для имеющейся у вас в наличии платы Nucleo. Определите порты ввода-вывода к которым подключены светодиод и кнопка. Если светодиод или кнопка отсутствуют на имеющейся у вас в наличии плате, подключите внешние.

Повторите пример, разобранный в процессе занятия, организовав управление частотой переключения светодиода. Для этого создайте 2 независимые задачи.

В качестве результата выполнения домашнего задания представьте код файла main.c или freertos.c (в зависимости от компоновки созданного проекта).

Готовое задание загрузить и отправить на проверку.

Самостоятельная работа слушателя по теме:

1. FreeRTOS_Real_Tutorial_Guide.pdf
2. FreeRTOS_Reference_Manual_V10.0.0.pdf
3. RTOS.xls
4. Опрос катушек.drawio

Тема 1.2. Управление задачами. Состояния задач. Приоритеты. Алгоритмы планирования.

Теоретические занятия:

Задачи.

Состояния задач. Приоритеты задач. Реализация задачи. Задача простоя. Перехватчик простоя.

Сопрограммы.

Состояния сопрограмм. Приоритет сопрограмм. Реализация сопрограмм. Планировщик и сопрограммы. Совместное использование задач и сопрограмм.

Ограничения.

Совместное использование стека. Использование операторов switch. Пример использования сопрограмм. Мигание светодиодом.

Планировщик сопрограмм. Создание сопрограммы и запуск диспетчера.

Расширенный пример: Использование параметра индекс.

Выводы раздела.

Характеристики «задачи». Характеристики «сопрограммы».

API в примерах.

xTaskAbortDelay(). xTaskCreate(). vTaskDelay(). vTaskDelayUntil(). vTaskDelete(). taskENTER_CRITICAL(). xTaskGetCurrentTaskHandle(). uxTaskGetNumberOfTasks(). vTaskGetRunTimeStats(). uxTaskGetStackHighWaterMark(). xTaskGetTickCount(). vTaskStartScheduler().

Практические занятия:

Домашнее задание. Создайте проект в среде STM32CubeIDE. Создайте несколько задач с разным приоритетом. В теле задач реализуйте “нагрузочный код” (например, создайте нагрузку за счет длительных циклов операции __NOP() или проведения затратных по времени математических вычислений).

Соберите статистику о времени, выделенном каждой из задач за счет использования внутренних счётчиков.

В качестве результата выполнения домашнего задания представьте исходный код (файлы main.c или freertos.c в зависимости от компоновки проекта) и ваши выводы о причинах замеченных явлений.

Готовое задание загрузить и отправить на проверку.

Самостоятельная работа слушателя по теме:

1. FreeRTOS functions.docx.

Тема 1.3. Управление очередями. Программные таймеры.

Теоретические занятия:

Очереди. Характеристики очереди. Доступ. Блокировка. Использование очередей. Блокировка очередей.

Функции API. xQueueCreate(). vQueueDelete(). uxQueueMessagesWaiting(). xQueuePeek(). xQueueReceive(). xQueueReset(). xQueueSend(). xQueueSendToFront(). xQueueSendToBack().

Практические занятия:

Домашнее задание. Создайте проект в среде STM32CubeIDE. Создайте две задачи:

Задача управления светодиодом;

Задача, которая опрашивает кнопку.

Организируйте отправку данных из задачи обслуживания кнопки в задачу отображения при помощи очереди.

В качестве результата выполнения домашнего задания представьте исходный код (файлы main.c или freertos.c в зависимости от компоновки проекта).

Готовое задание загрузить и отправить на проверку.

Тема 1.4. Таймеры и управление кучей. Управление памятью. Программные таймеры.

Теоретические занятия:

Управление кучей.

Распределение динамической памяти и его значение для FreeRTOS.

Heap_1. Heap_2. Heap_3. Heap_4. Heap_5.

Отслеживание размера кучи. Программные таймеры. Функция обратного вызова.

Период программного таймера. Состояния таймера.

API функции. xTimerChangePeriod(). xTimerCreate(). xTimerDelete(). xTimerReset(). xTimerStart(). xTimerStop().

Практические занятия:

Домашнее задание.

Создайте проект в среде STM32CubeIDE, в проекте реализуйте программный таймер для управления светодиодом. Опробуйте использование программного таймера для обновления сторожевого таймера как признак работоспособности операционной системы.

В качестве результата выполнения домашнего задания представьте исходный код (файлы main.c или freertos.c в зависимости от компоновки проекта).

Готовое задание загрузить и отправить на проверку.

Тема 1.5. Ресурсы и прерывания. Двоичные и счетные семафоры. События. Группы событий. Нотификация.

Теоретические занятия:

Ресурсы и прерывания.

Прерывания. Приоритет задачи и приоритет прерывания. Функции API для использования в обработчике прерывания. Преимущества.

Недостатки. Параметр `xHigherPriorityTaskWoken`. Переключение контекста.

Макросы `portYIELD_FROM_ISR ()` и `portEND_SWITCHING_ISR ()`. Отложенная обработка прерывания. Совместный доступ к ресурсам.

Взаимное исключение. Критические разделы и приостановка планировщика. Приостановка (или блокировка) планировщика. Совместное использование ресурсов. Мэдрик код.

Практические занятия:

Домашнее задание.

В дополнительных материалах к данному уроку имеется руководство пользователя и инструкция к прибору «Модули мониторинга и информирования серии GТех». Также, вместо приложенных документов вы можете использовать формальное описание любого другого прибора из сферы автоматизации.

В качестве домашнего задания к данному уроку предложите свою реализацию разделения кода проекта на задачи, очереди, прерывания. Кратко опишите функции каждой из задач и взаимодействие между ними.

Результатом выполнения домашнего задания является PDF документ с описанием функционала предлагаемых вами задач, очередей, обработчиков прерываний для реализации требуемого функционала.

Вопросы к практическому заданию:

1. Сколько будет задач?
2. Какие функции вы возложите на эти задачи?
3. GSM-модем подключён по интерфейсу UART. Как мы будем взаимодействовать с модемом?
4. Потребуется ли использование очередей?

Готовое задание загрузить и отправить на проверку.

Самостоятельная работа слушателя по теме:

1. GТех 3.pdf

2. Table of config.pdf

Тема 1.6. События, нотификация, семафоры.

Теоретические занятия:

Семафоры. Двоичные семафоры, используемые для синхронизации.

`xSemaphoreCreateBinary()`. `xSemaphoreTake()`. `xSemaphoreGiveFromISR()`.

Счётные семафоры. `xSemaphoreCreateCounting()`.

Пример синхронизации задачи и обработчика прерывания.

События. Характеристики группы событий. Группы событий, флаги событий и биты событий.

`EventBits_t`. Мультизадачный доступ.

Практический пример использования группы событий.

Управление событиями с использованием групп событий.

`xEventGroupSetBits ()`. `xEventGroupSetBitsFromISR()`. `xEventGroupWaitBits()`.

Синхронизация задач с помощью группы событий. `xEventGroupSync()`.

Нотификация. Общение через посреднические объекты.

Уведомления. Преимущества и ограничения. Ограничения уведомлений. Использование уведомлений.

xTaskNotifyGive (). vTaskNotifyGiveFromISR (). ulTaskNotifyTake (). xTaskNotify () и xTaskNotifyFromISR (). xTaskNotifyWait().

Практические занятия:

Домашнее задание.

Достаточно часто в промышленной автоматике требуется реализация различной логики управления.

К примеру, выход активируется, если состояние входов соответствует той или иной маске событий.

Также часто требуется формировать управляющий сигнал с задержкой.

Используйте 4 кнопки, подключённые к плате Nucleo в качестве входов, а 4 светодиода — в качестве выходов. Используя механизмы операционной системы FreeRTOS, реализуйте программу, устанавливающую состояние выходов в зависимости от того, соответствует ли состояние входов запрограммированной «маске».

Пример логики:

Выход 1 должен срабатывать при событиях 1 и 4 входов;

Выход 2 при событии 1 входа;

Выход 3 при событиях 3 и 4 входов;

Выход 4 при событии любого входа.

Сделайте маску событий изменяемой.

В качестве результата выполнения домашней работы представьте исходный код (файлы main.c или freertos.c в зависимости от компоновки проекта).

Готовое задание загрузить и отправить на проверку.

Самостоятельная работа слушателя по теме:

1. scmRTOS.ru.pdf.

Тема 1.7. Снижение энергопотребления. Управление питанием.

Теоретические занятия:

Снижение энергопотребления.

Режим Sleep. Режим Stop. Режим Standby.

FreeRTOS и режимы энергопотребления.

Макрос portSUPPRESS_TICKS_AND_SLEEP (). TICKLESS_IDLE в генераторе кода.

Реализация portSUPPRESS_TICKS_AND_SLEEP (). vTaskStepTick. xTaskCatchUpTicks. eTaskConfirmSleepModeStatus.

SysTick и частота ядра.

Генерация тактов от источника, отличного от SysTick.

Использование специфических функций с низким энергопотреблением.

Демонстрационный пример.

Выводы.

Практические занятия:

Домашнее задание.

Дополните ранее выполненный проект следующими элементами:

Подключите 4 кнопки в качестве имитации входных сигналов;

Подключите 4 светодиодных индикатора (или дисплей, на ваш выбор) в качестве индикации дискретных выходных сигналов;

Перепишите код проекта, обеспечивающий функционирование платы Nucleo с использованием операционной системы FreeRTOS;

Дополните страницу, отображаемую созданным WEB-сервером, механизмом ввода маски, логически соединяющей входы и выходы платы.

Например:

Выход 1 должен срабатывать при событиях 1 и 4 входов;

Выход 2 при событии 1 входа;
Выход 3 при событии 3 и 4 входов;
Выход 4 при событии любого входа.

Критерии оценки:

В ходе проверки работы будет оцениваться код, разработанный для отладочной платы Nucleo.

Ожидается что в процессе выполнения итоговой работы вы продемонстрируете знания, полученные на курсе по операционной системе FreeRTOS, а именно:

Адекватно задаче спланируете задачи.

Используете средства обмена информацией между задачами (очереди).

Готовое задание загрузить и отправить на проверку.

Тема 1.8. Сложные случаи. Макросы. Стил ь кода.

Теоретические занятия:

Примеры. Получение данных из нескольких источников. Разные типы и длины данных в очереди.

Перехватчик запуска задачи демона. Откладывание работы на демона. xTimerPendFunctionCallFromISR().

Функционал для разработчиков. configASSERT().

Статистика времени выполнения. uxTaskGetSystemState(). vTaskList(). vTaskGetRunTimeStats(). xPortGetFreeHeapSize(). xPortGetMinimumEverFreeHeapSize().

Макросы-ловушки. Определение макросов трассировки.

Использование макросов.

4. Организационно-педагогические условия реализации рабочей программы учебного модуля 1. «Операционная система FreeRTOS»

Реализация рабочей программы учебного модуля должна обеспечить приобретение слушателями знаний и умений, необходимых для работы в операционной системе FreeRTOS». Выбор методов обучения для каждого занятия определяется преподавателем в соответствии с составом и уровнем подготовленности обучающихся, степенью сложности излагаемого материала, наличием и состоянием технических возможностей обучающихся.

Теоретические занятия проводятся с целью изучения нового учебного материала. Изложение материала необходимо вести в форме, доступной для понимания обучающихся.

Практическое занятие предусматривает выполнение практического задания, которое включает в себя отработку умений и навыков по всем темам модуля. Практическое задание выполняется после изучения каждой темы модуля (полностью).

Цель практических заданий – дать возможность обучающемуся проверить понимание теоретических положений изучаемого курса, знать основные понятия, классификации и т.п.

В процессе организации работы большое значение имеют самостоятельные работы слушателей. Самостоятельная работа предназначена не только для овладения конкретной темой программы, но и для формирования навыков самостоятельной работы вообще (способности самостоятельно решать проблемы, находить конструктивные решения).

Материально-техническое обеспечение рабочей программы учебного модуля:

Дополнительная профессиональная программа повышения квалификации реализуется полностью с применением дистанционных образовательных технологий (ДОТ). Для эффективного внедрения ДОТ и использования электронных образовательных ресурсов имеется качественный доступ педагогических работников и обучающихся к информационно-телекоммуникационной сети Интернет. Услуга подключения к сети Интернет предоставляется в режиме 24 часа в сутки 7 дней в неделю без учета объемов потребляемого трафика. Для использования ДОТ педагогическому работнику

предоставляется свободный доступ к средствам информационных и коммуникационных технологий.

Рабочее место педагогического работника:

№ п/п	Наименование	Кол-во(шт.)
1.	Системный блок Intel Core i3, 2, 3400 МГц	1
2.	Монитор Philips	1
3.	Клавиатура	1
4.	Мышь проводная Genius	1
5.	Наушники	1
6.	Веб-камера	1
7.	Ноутбук	1
8.	Стол преподавателя	1
9.	Стул преподавателя	1

В состав программно-аппаратных комплексов установлено программное обеспечение, необходимое для осуществления учебного процесса: общего назначения (операционная система (операционные системы), офисные приложения, средства обеспечения информационной безопасности, архиваторы, графический, видео- и аудио-редакторы).

Формирование информационной среды осуществляется с помощью программной системы дистанционного обучения, расположенном на сайте, открытом для свободного ознакомления, публично доступном для физических и юридических лиц.

С помощью системы дистанционного обучения (далее - СДО):

- преподаватели разрабатывают и размещают содержательный контент; планирует свою педагогическую деятельность: выбирает из имеющихся или создает нужные для обучающихся ресурсы и задания;

- педагогические работники и обучающиеся обеспечиваются доступом к полной и достоверной информации о ходе учебного процесса, промежуточных и итоговых результатах благодаря автоматическому фиксированию указанных позиций в информационной среде;

- обучающиеся выполняют задания, предусмотренные образовательной программой, при необходимости имеют возможность обратиться к педагогическим работникам за помощью;

- все результаты обучения сохраняются в информационной среде, на их основании формируются портфолио обучающихся и педагогических работников.

Система дистанционного обучения содержит образовательный контент доступный для Пользователей по интернет-адресу СДО с использованием настольных компьютеров или ноутбуков. Образовательная организация осуществляет электронную систему учета контингента обучающихся, путем регистрации обучающихся в системе дистанционного обучения, учет по прохождению обучающимися процесса обучения, формирует статистический отчет и данные о прохождении промежуточной и итоговой аттестации.

Требования к рабочим местам обучающихся:

Наличие ПК с Операционной системой Windows 7, Windows 8, Windows 8.1, Windows 10 и более поздней версии, а также MacOS 9 и выше или Ubuntu 18+;

Оперативная память не менее 4 Гб;

Монитор с разрешением от 1366x768 и выше, 256 цветов;

Установленный интернет-браузер на базе Chromium (Chrome 64.0 или выше, Яндекс.Браузер 17.6.1 или выше; Opera версии 60.x и выше);

Обеспечение устойчивого и качественного доступа обучаемых в информационно-коммуникационную сеть Интернет на период обучения;

Наличие личного E-mail для каждого пользователя или доступ к мессенджеру Telegram со своих рабочих компьютеров или личных мобильных устройств.

5. Требования к кадровому обеспечению образовательного процесса рабочей программы учебного модуля 1. «Операционная система FreeRTOS»

Реализация рабочей программы учебного модуля обеспечивается научно-педагогическими работниками организации, а также лицами, привлекаемыми к реализации программы на иных законных основаниях.

Квалификация руководящих и научно-педагогических работников организации соответствует квалификационным характеристикам, установленным в Едином квалификационном справочнике должностей руководителей, специалистов и служащих, раздел «Квалификационные характеристики должностей руководителей и специалистов высшего профессионального и дополнительного профессионального образования», утвержденном приказом Министерства здравоохранения и социального развития Российской Федерации от 11 января 2011 г. №1н.

6. Учебно-методическое и информационное обеспечение учебного модуля 1. «Операционная система FreeRTOS»

Основная литература, использованная при разработке образовательной программы:

1. Мединцев Владимир М42 Операционные системы микроконтроллеров: На примере операционной системы реального времени FreeRTOS / Владимир Мединцев. 16. м.]: Издательские решения, 2023. - 280 с. ISBN 978-5-0060-5990-0

2. The FreeRTOS Reference manual / 2017 г. – URL: https://www.freertos.org/Documentation/FreeRTOS_Reference_Manual_V10.0.0.pdf – Текст: электронный.

3. Гордеев, А.В. Операционные системы / А.В. Гордеев – 2-е изд. – СПб.: Питер, 2004.

4. Курниц, А. FreeRTOS – операционная система для микроконтроллеров / А. Курниц // Компоненты и технологии – 2011 г.

5. FreeRTOS. Краткое описание / URL: http://easyelectronics.ru/freertos_manual.html – Текст: электронный.

Дополнительная литература:

1. Операционные системы реального времени / И.Б. Бурдонов, А.С. Косачев, В.Н. Пономаренко – Москва: ИСП РАН, 2006 г. – URL: http://citforum.ru/operating_systems/rtos/1.shtml – Текст: электронный.

2. Операционные системы реального времени и Windows NT / Е. Хухлаев – 1997 г. – URL: <https://www.osp.ru/os/1997/05/179265> – Текст: электронный.

3. Most Popular Open-source RTOS Comparison for Embedded Systems / URL: <https://scienceprog.com/most-popular-open-source-rtos-comparison-for-embedded-systems/> – Текст: электронный.

4. Обзор ОСРВ FX-RTOS / URL: <https://www.aremex.ru/products/fx-rtos/> – Текст: электронный.

5. Операционная система Eremex FX-RTOS Руководство пользователя / Версия 3.0 – 2018г. – URL: https://www.aremex.ru/upload/iblock/1ee/FXRTOS_GUIDE.pdf – Текст: электронный.