

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ.

**«Национальный исследовательский университет
ИТМО»**

Факультет программной инженерии и компьютерной техники

Дисциплина «Информационно-управляющие системы»

Отчет

«Программный комплекс RTOS-Trace»

Выполнили:

Студенты

Строзенко Константин Викторович

Берман Денис Константинович

Гуменник Пётр Олегович

Гайдеров Ярослав Игоревич

Григорьев Александр Алексеевич

Кенжаев Рахиджон Ермахмадович

Преподаватель:

Ключев Аркадий Олегович

Санкт-Петербург

2025 г.

Техническое задание на разработку программного комплекса «RTOS-Trace»

Техническое задание
Листов «5»

Оглавление

1 Введение	2
1.1 Наименование программы	2
1.2 Краткая характеристика области применения	2
2 Основания для разработки	2
3 Назначение разработки	2
3.1 Функциональное назначение	2
3.2 Эксплуатационное назначение	3
4 Требования к программе	3
4.1 Функциональные характеристики	3
4.1.1 Состав выполняемых функций	3
4.1.2 Организация входных данных	3
4.1.3 Организация выходных данных	3
4.1.4 Временные характеристики	3
4.2 Требования к надежности	3
4.2.1 Обеспечение устойчивого функционирования	4
4.2.2 Время восстановления после отказа	4
4.2.3 Отказы из-за некорректных действий пользователя	4
4.3 Условия эксплуатации	4
4.4 Требования к составу и параметрам технических средств	4
4.5 Требования к информационной и программной совместимости	4
4.6 Требования к маркировке и упаковке	4
4.7 Требования к транспортированию и хранению	4
4.8 Специальные требования	4
5 Требования к программной документации	4
6 Технико-экономические показатели	5
7 Стадии и этапы разработки	5
8 Порядок контроля и приемки	5
9 Примечания	5
Список используемой литературы	5

1 Введение

1.1 Наименование программы

Программный комплекс «RTOS-Trace».

1.2 Краткая характеристика области применения

RTOS-Trace — программный инструмент для регистрации и визуализации переключений задач в среде FreeRTOS. В отличие от традиционных отладчиков, он не требует остановки выполнения и обеспечивает непрерывный сбор данных о работе задач в формате CSV с последующим построением временной диаграммы. Это позволяет разработчику быстро выявлять логические ошибки, конфликты и аномалии в планировании задач.

Комплекс полностью программный, не требует дополнительного оборудования и легко встраивается в существующий код с минимальными изменениями. Его применение снижает затраты на отладку и сокращает сроки разработки, что делает решение экономически эффективным уже на этапе первого внедрения.

2 Основания для разработки

Параметр	Значение
Заказчик	НИУ «ИТМО»
Представитель заказчика	Ключев Аркадий Олегович, преподаватель дисциплины «Информационно-управляющие системы»
Исполнитель	Команда студентов 3 курса факультета ПИиКТ
Название темы	«Разработка средства трассировки переключений задач в FreeRTOS»
Шифр темы	RTOS-T03
Сроки выполнения	01.04.2025 – 20.06.2025

3 Назначение разработки

3.1 Функциональное назначение

Комплекс обеспечивает разработчиков систем реального времени следующими возможностями:

№	Функция	Описание и проверяемый результат
F1	Регистрация событий	Встраиваемая библиотека фиксирует каждый факт переключения задач FreeRTOS; событие содержит: <code>time_tick</code> , <code>process_id</code> , <code>status</code> (0 – остановлена, 1 – работает).

№	Функция	Описание и проверяемый результат
F2	Сохранение журнала	После остановки приложения библиотека создаёт <code>trace_log.csv</code> в рабочем каталоге эмулятора. Проверка: файл существует, структура и порядок столбцов соответствуют F1.
F3	Построение диаграммы	Скрипт <code>visualize_trace.py</code> принимает путь к CSV и формирует PNG-изображение с Gantt-диаграммой. Проверка: запуск скрипта без параметров выдаёт справку; с параметром – создаёт файл <code>trace.png</code> в текущем каталоге.
F4	Пакет минимальных тестов	Набор unit-тестов PyTest подтверждает корректность парсинга CSV и построения диаграммы. Все тесты проходят (<code>pytest</code> возвращает <code>code 0</code>).

3.2 Эксплуатационное назначение

Программа запускается на ПК разработчика для анализа поведения задач FreeRTOS в эмуляторе.

4 Требования к программе

4.1 Функциональные характеристики

4.1.1 Состав выполняемых функций

Перечень функций приведён в табл. F1 - F4 (см. п. 3.1).

4.1.2 Организация входных данных

Источник – встраиваемая библиотека FreeRTOS (код в репозитории проекта).

Формат – три столбца CSV: `time_tick:uint64, process_id:uint16, status:uint8`.

Ввод со стороны пользователя – отсутствует.

4.1.3 Организация выходных данных

`trace_log.csv` – журнал событий;

`trace.png` – диаграмма переключений.

4.1.4 Временные характеристики

Время построения диаграммы на тестовом стенде не превышает 2 секунд при размере лога до 10 000 записей.

4.2 Требования к надежности

Комплекс должен обеспечивать безотказную работу с вероятностью не ниже 0,98 в течение непрерывной 8-часовой сессии.

4.2.1 Обеспечение устойчивого функционирования

При отсутствии прав записи библиотека завершает приложение с кодом 1 и сообщением об ошибке.

Исходный код снабжён unit-тестами; цель – покрытие ≥ 50 % основного кода.

4.2.2 Время восстановления после отказа

Повторный запуск после нефатального сбоя ОС или приложения не должен превышать 5 минут (время на перезапуск эмулятора FreeRTOS и повторную генерацию диаграммы).

4.2.3 Отказы из-за некорректных действий пользователя

Отказ возможен, если пользователь удалит CSV до визуализации. Скрипт обязан вывести понятное предупреждение и завершиться без исключения.

4.3 Условия эксплуатации

Эксплуатация на офисном ноутбуке при +18...+25 °C, влажность до 75%.

Операционная система – Windows 10/11 x64.

Python ≥ 3.10 (официальный дистрибутив).

4.4 Требования к составу и параметрам технических средств

Ноутбук с CPU x86-64 1 ГГц, 4 ГБ RAM, 200 МБ свободного места на диске.

4.5 Требования к информационной и программной совместимости

Python-библиотеки matplotlib и pandas.

FreeRTOS v10.5.1

4.6 Требования к маркировке и упаковке

Не предъявляются.

4.7 Требования к транспортированию и хранению

Не предъявляются.

4.8 Специальные требования

Исходный код хранится в Git, лицензия MIT.

5 Требования к программной документации

Включает ТЗ, описание архитектуры, реализации, тестов и отчёт об испытаниях.

6 Техно-экономические показатели

Использование «RTOS-Trace» позволяет снизить трудозатраты на отладку задач FreeRTOS на 5-10 %; лицензия распространяется бесплатно.

7 Стадии и этапы разработки

Стадия	Этап	Срок завершения	Выходной документ
Техническое задание	Разработка, согласование, утверждение	16.05.2025	утверждённое ТЗ
Тех./раб. проект	Разработка кода, unit-тестов, документации	17.05.2025	исходный код, README
Испытания	Приемо-сдаточные испытания	18.05.2025	отчет об испытаниях
Внедрение	Передача в эксплуатацию	19.05.2025	

8 Порядок контроля и приемки

Приемо-сдаточные испытания выполняются по «Программе и методике испытаний». Условием приёмки является успешное прохождение всех тестов и подписание акта Заказчиком и Исполнителем.

9 Примечания

При необходимости уточнения требований изменения вносятся в ТЗ по согласованию сторон и оформляются протоколом изменений.

Описание архитектуры

Основные компоненты

MyTasks.c – содержит реализацию пользовательских задач. В этом модуле определяются функции задач, имитирующие полезную нагрузку (workload). Каждая задача может, например, выполнять вычисления или задержки с помощью вызовов FreeRTOS (таких как vTaskDelay), чтобы создать характерную загрузку. Задачи из MyTasks.c регистрируются (создаются) при инициализации системы и затем передаются планировщику FreeRTOS для конкурентного выполнения.

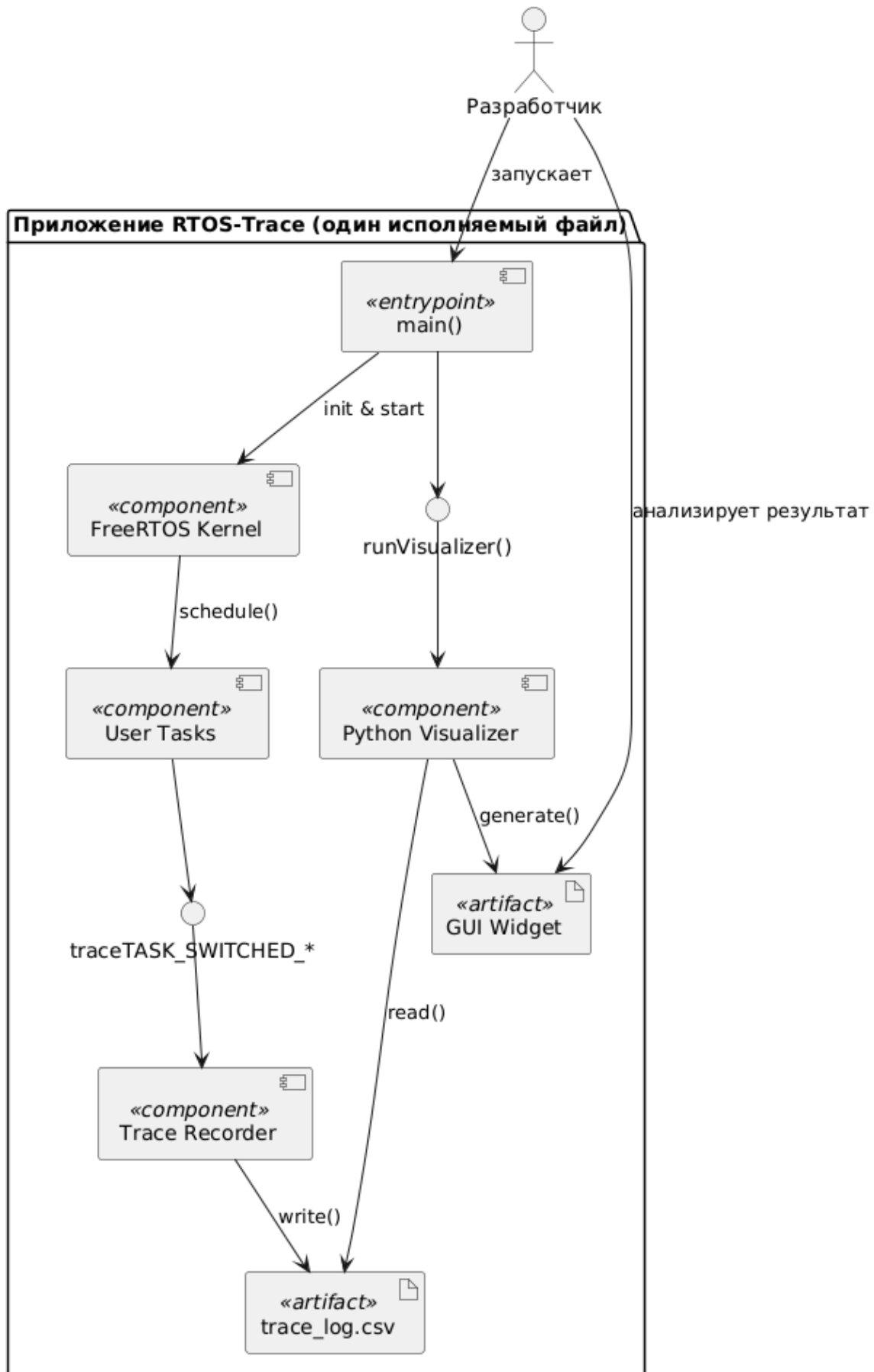
tracehooks.c – реализует хуки планировщика FreeRTOS для отслеживания переключений задач. Этот модуль перехватывает события смены выполняемой задачи (вход задачи на

выполнение и ее вытеснение) с помощью специальных макросов FreeRTOS trace hooks. При каждом переключении контекста функция-хук из tracehooks.c автоматически вызывается: она регистрирует событие в лог-файл (в формате CSV), фиксируя имя задачи, тип события (например, переключение на задачу или с задачи) и метку времени (текущий системный тик). Модуль отвечает за открытие файла трассировки при старте, построчную запись событий во время работы системы и корректное закрытие файла по завершении сбора данных.

main.c – отвечает за инициализацию и запуск системы. В функции main() создаются задачи (определенные в MyTasks.c), настраиваются приоритеты и необходимые ресурсы, после чего вызывается запуск планировщика через функцию FreeRTOS vTaskStartScheduler(). Кроме того, main.c содержит реализацию хука тика ОС – функцию vApplicationTickHook. Этот тик-хук настроен так, чтобы отслеживать продолжительность работы системы по количеству тиков: по достижении заданного порога (предельного числа тиков) он инициирует завершение работы задачи-трассировщика. После остановки планировщика (то есть после завершения заданного временного интервала трассировки) main.c выполняет системный вызов (system("python <path>")) для запуска внешнего Python-скрипта, который займется обработкой собранных данных и их визуализацией.

Python-скрипт визуализации – внешняя компонента, реализующая анализ и отображение результатов трассировки. Данный скрипт запускается автоматизированно из main.c по завершении работы RTOS. Он читает сгенерированный CSV-файл с логом событий из tracehooks.c и строит диаграмму Ганта – графическое представление исполнения задач во времени. В диаграмме каждая задача отображается на отдельной горизонтальной линии, а временные интервалы, когда та или иная задача была активна (исполнялась на процессоре), показаны в виде отрезков (цветных блоков), расположенных вдоль оси времени.

Компонентная UML-диаграмма комплекса «RTOS-Trace»



Список используемой литературы

1. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. – Введ. 1979-01-01. – М.: Изд-во стандартов, 1978. – 11 с.
2. FreeRTOS Plus Trace – Trace Recorder Library Documentation [Электронный ресурс]. – URL: https://freertos.org/Documentation/03-Libraries/02-FreeRTOS-plus/05-FreeRTOS_plus_Trace/00-FreeRTOS_Plus_Trace (дата обращения: 18.05.2025).
3. Hook Functions in FreeRTOS [Электронный ресурс]. – URL: <https://www.freertos.org/Documentation/02-Kernel/02-Kernel-features/12-Hook-functions> (дата обращения: 18.05.2025).
4. Tasks and Co-routines in FreeRTOS [Электронный ресурс]. – URL: <https://www.freertos.org/Documentation/02-Kernel/02-Kernel-features/01-Tasks-and-co-routines/00-Tasks-and-co-routines> (дата обращения: 18.05.2025).
5. Tracealyzer for FreeRTOS – User Event and Signal Plot View [Электронный ресурс]. – URL: <https://percepio.com/getstarted/latest/html/freertos.html> (дата обращения: 18.05.2025).
6. STMicroelectronics. X-CUBE-FREERTOS. FreeRTOS™ Software Expansion for STM32Cube: Data Brief [Электронный ресурс]. – URL: https://www.st.com/resource/en/data_brief/x-cube-freertos.pdf (дата обращения: 18.05.2025).
7. FreeRTOS – ESP-IDF Programming Guide [Электронный ресурс]. – URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/freertos.html> (дата обращения: 18.05.2025).