

УДК 004.451

DOI: 10.46548/21vek-2021-1056-0015

АППАРАТНАЯ РЕАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ НЕКОТОРЫХ СОВРЕМЕННЫХ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

© 2021

Синев Михаил Петрович, кандидат технических наук, доцент кафедры «Вычислительная техника»

Пензенский государственный университет

(440026, Россия, Пенза, улица Красная, 40, e-mail: mix.sinev@gmail.com)

Мартышкин Алексей Иванович, кандидат технических наук, доцент,

заведующий кафедрой «Программирование»

Пензенский государственный технологический университет

(440039, Россия, Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11, e-mail: alexey314@yandex.ru)

Аннотация. В статье проведено исследование трех современных алгоритмов планирования операционных систем реального времени. Экспериментальное исследование алгоритмов планирования, выполнялось на базе аппаратной платформы *STM32F407G-DISC1*. Программная основа для исследования алгоритмов – операционная система *FreeRTOS*, которая имеет возможность работать в двух режимах – с вытесняющей и кооперативной многозадачностью. Экспериментально реализованы алгоритмы планирования: *Rate Monotonic*, *Earliest Deadline First* и *Background*. Для каждого из них синтезировались функции создания задач, которые планируются в системе. На базе планировщика *FreeRTOS* реализован принцип работы каждого из перечисленных алгоритмов. После подготовки к реализации алгоритмов планирования проводились непосредственно эксперименты с последующей проверкой их работоспособности. Исследования алгоритмов планирования операционных систем реального времени производились на основе показателей – длительность переключения контекста и эффективность планировщика. Проведены эксперименты с разным количеством задач для определения тенденции изменения каждого из параметров. Основные полученные результаты наглядно показаны на графиках. В результате исследования представлены рекомендации для выбора определенного алгоритма планирования при конкретных типах задач и значениях загрузки. В заключении сформулированы основные выводы по проделанной работе.

Ключевые слова: аппаратная реализация, операционная система, система реального времени, встраиваемая система, алгоритмы планирования, планировщик, эффективность.

HARDWARE IMPLEMENTATION AND RESEARCH OF SOME MODERN ALGORITHMS FOR PLANNING REAL-TIME OPERATING SYSTEMS

© 2021

Sinev Mihail Petrovich, candidate of technical sciences,
associate professor of sub-department «Computer engineering»

Penza State University

(440026, Russia, Penza, Krasnaya Street, 40, e-mail: mix.sinev@gmail.com)

Martyshev Alexey Ivanovich, candidate of technical sciences, docent, head of sub-department «Programming»

Penza state technological University

(440039, Russia, Penza, BaydukovProyezd / Gagarin Street, 1a/11, e-mail: alexey314@yandex.ru)

Abstract. The article studies three modern algorithms for planning real-time operating systems. An experimental study of scheduling algorithms was performed on the basis of the *STM32F407G-DISC1* hardware platform. The software basis for the study of algorithms is the *FreeRTOS* operating system, which has the ability to work in two modes – with preemptive and cooperative multitasking. Planning algorithms have been experimentally implemented: *Rate Monotonic*, *Earliest Deadline First* and *Background*. For each of them, the functions of creating tasks that are planned in the system were synthesized. The principle of operation of each of the listed algorithms is implemented on the basis of the *FreeRTOS* scheduler. After preparing for the implementation of the planning algorithms, experiments were carried out directly with subsequent verification of their operability. Studies of algorithms for planning real-time operating systems were carried out on the basis of indicators - the duration of context switching and the effectiveness of the scheduler. Experiments were conducted with a different number of tasks to determine the trend of change in each of the parameters. The main results obtained are clearly shown in the graphs. As a result of the study, recommendations are presented for choosing a specific scheduling algorithm for specific types of tasks and load values. In conclusion, the main conclusions on the work done are formulated.

Keywords: hardware implementation, operating system, real-time system, embedded system, scheduling algorithms, scheduler, efficiency.

Введение. Алгоритм планирования – неотъемлемая часть любой операционной системы, во основном определяющая эффективность использования аппа-

ратных ресурсов 1 встраиваемой вычислительной системы (многопроцессорной, многоядерной, распределенной) [1]. Главным образом от алгоритма пла-

нирования зависит, какая задача будет выполняться в системе реального времени в каждый момент [2-3]. Время срабатывания системы или реакции на внешнее воздействие имеет большую роль, и медленная реакция может привести к непредсказуемым последствиям.

Цель работы – исследование и сравнение некоторых востребованных алгоритмов планирования операционных систем реального времени.

Объект исследования – операционная система реального времени для встраиваемых систем.

Предмет исследования – аппаратно реализованные алгоритмы планирования.

Материалы и результаты исследования. В статье для достижения поставленной цели решаются задачи: аппаратная реализация некоторых известных алгоритмов планирования на базе платформы *STM32F407G-DISCI*; экспериментальные исследования с аппаратно реализованными алгоритмами.

В работах [4-8], [9-11] и [12-14] приведены известные алгоритмы планирования операционных систем реального времени (ОСРВ). Проанализированы наиболее популярные из них. Для оценки производительности функций операционных систем применяют программное обеспечение, аналогичное представленному здесь [15, 16]. В этой статье попытаемся подойти к вопросу исследования современных алгоритмов планирования операционных систем реального времени для встраиваемых систем со стороны их аппаратной реализации и дальнейшего анализа. Представим результаты оценки таких характеристик алгоритмов планирования, как переключение контекста и коэффициент эффективности планировщика.

Реализация алгоритмов планирования, выполня-

лась на аппаратной платформе *STM32F407G-DISCI* [17]. Программной основой для исследования алгоритмов является операционная система *FreeRTOS*, которая может работать в двух режимах – с вытесняющей и кооперативной многозадачностью [18]. Во *FreeRTOS*, задачи, готовые к выполнению, представляют собой структуры и хранятся в виде упорядоченных закольцованных двусвязных списков [19, 20]. Для реализации алгоритмов подготовлены специальные списки задач для хранения готовых к исполнению задач. На платформе *STM32F407G-DISCI* с ОСРВ *FreeRTOS* реализованы алгоритмы планирования: *Rate Monotonic*, *Earliest Deadline First* и *Background*. Для каждого из перечисленных алгоритмов синтезировались функции создания задач, которые планируются в системе. На базе планировщика *FreeRTOS* реализован принцип работы каждого из перечисленных алгоритмов. После подготовки к реализации алгоритмов планирования проводились непосредственно эксперименты с последующей проверкой их работоспособности.

Для проверки работоспособности алгоритма *Rate Monotonic* создано 3 периодических задачи (с одинаковым временем выполнения, равным 9 нс) с периодами 35 нс, 40 нс и 50 нс соответственно. Значение коэффициента загрузки для такого набора задач составило 68%, что является допустимой и говорит о том, что данные задачи могут планироваться с использованием алгоритма *Rate Monotonic*.

Получены трассировочные данные, для наглядного их отображения построим схематический график, представленный на рисунке 1, где зафиксированы время выполнения и тип задачи, выполняемой в определенный момент времени.

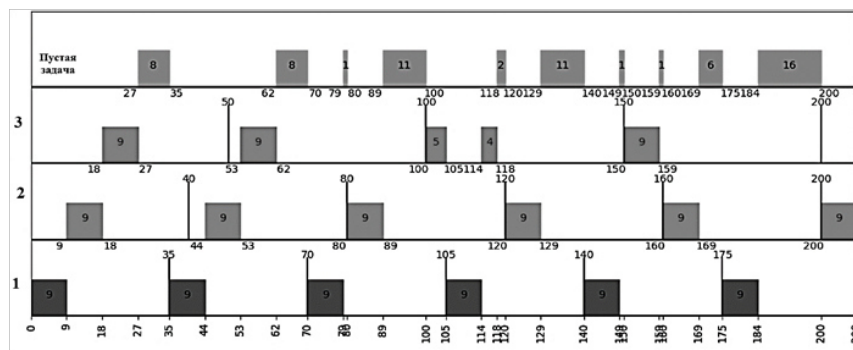


Рисунок 1 – Результат работы алгоритма *Rate Monotonic*

Проведя анализ полученных трассировочных данных, можно подтвердить, что алгоритм работает верно, т.к. в каждый момент планирования происходит выбор задачи с наименьшим периодом. Наиболее отчетливо это можно видеть во временном интервале с 100 по 118 миллисекунду, где первоначально управление получает 3-я задача (100 мс), но после появления первой задачи в списке готовых, 3-я задача (105 мс) вытаскивается и только после выполнения первой задачи (114 мс), продолжает выполняться (114-118 мс).

Для алгоритма *Rate Monotonic* произведены вре-

менные замеры длительности переключения контекста на входе и выходе из функции *vTaskSwitchContext*. Более того, проводилось фиксирование длительности трассировочных конструкций на входе и выходе функции, значения которых составили по 7,1 мкс.

Кроме показателя времени переключения контекста, для оценки алгоритма планирования известно понятие коэффициент эффективности планировщика – отношение среднего времени переключения контекста к минимальному времени выполнения задач. Чтобы оценить длительность времени переключения

контекста проведены тестовые запуски системы с 1, 3, 5, 7 и 9 задачами.

Результаты оценки этих параметров можно увидеть на рисунке 2.

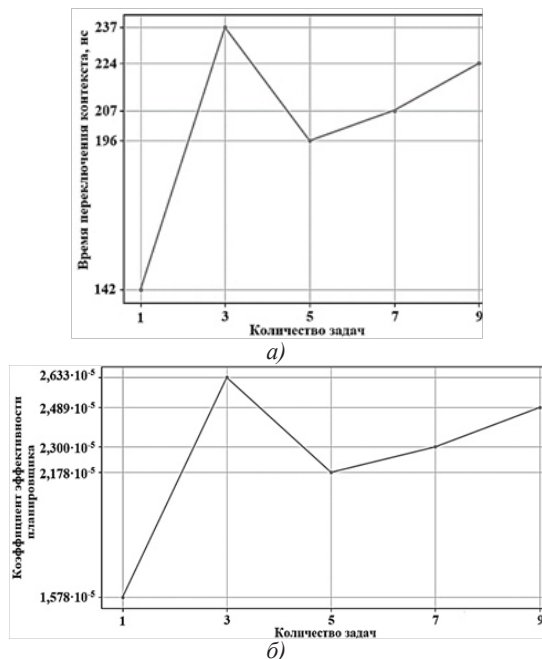


Рисунок 2 – Зависимость времени переключения контекста от количества задач (а); коэффициента эффективности планировщика от количества задач (б) алгоритма Rate Monotonic

Из рисунка 2,а видно, что время переключения контекста алгоритма Rate Monotonic находится в интервале от 142 нс до 237 нс и среднее время составляет 201 нс. По рисунку 2,б можно определить среднее значение коэффициента эффективности планировщика, которое составило $2,2236 \cdot 10^{-5}$.

Для проверки работы алгоритма Earliest Deadline

First (EDF) создано 3 периодических задачи (время выполнения, равное 9 нс на задачу) с периодами 20 нс, 30 нс и 40 нс соответственно. Значение коэффициента загрузки для такого набора задач составило 97,5%. Получены представленные ниже трассировочные данные (рис. 3).

После анализа полученных данных подчеркнем, что алгоритм работает верно, т.к. в каждый момент планирования происходит выбор задачи с ранним крайним сроком.

Отличительные черты исследуемого алгоритма можно проследить с 27 нс по 45 нс. В начале управление получает первая задача (27 нс). В 30 наносекунду в список готовых задач возвращается 2-я задача, поскольку наступил ее новый период. Однако 2-я задача не вытесняет первую, поскольку первая имеет более ранний крайний срок выполнения (40 нс), чем 2-я (60 нс).

Для оценки длительности переключения контекста проведены эксперименты с 1, 3, 5, 7 и 9 задачами. Результаты оценки этих параметров можно увидеть на рисунке 4.

По рисунку 4,а определим, что время переключения контекста для алгоритма EDF находится в промежутке от 214 нс до 228 нс и среднее время составляет 221 нс. По рисунку 4,б определим среднее значение коэффициента эффективности планировщика, которое составило $2,46 \cdot 10^{-5}$.

Далее покажем результат исследования алгоритма Background. Были созданы 2 периодические задачи со следующими характеристиками: время выполнения – 9 нс на задачу и периодами 20 нс и 40 нс соответственно, и 2 аperiodических запроса с характеристиками: время выполнения – 9 нс на задачу. Система с указанными характеристиками была протестирована, получены трассировочные данные (рис. 5).

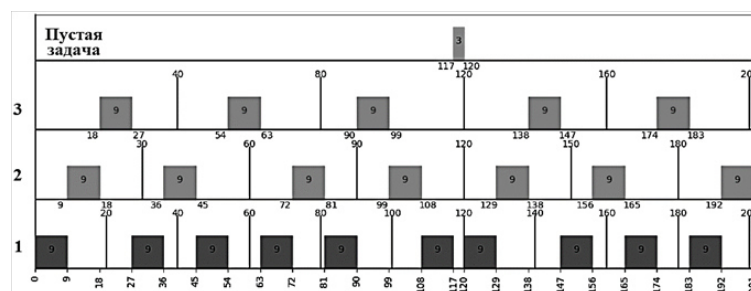


Рисунок 3 – Результат работы алгоритма Earliest Deadline First

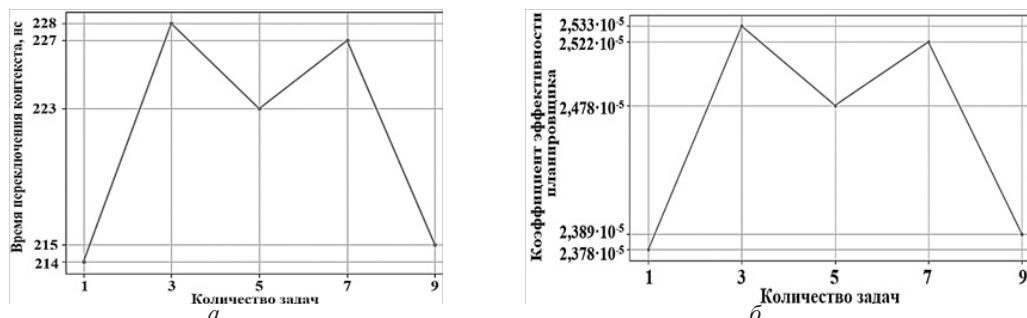


Рисунок 4 – Зависимость времени переключения контекста от количества задач (а); коэффициента эффективности планировщика от количества задач (б) алгоритма Earliest Deadline First

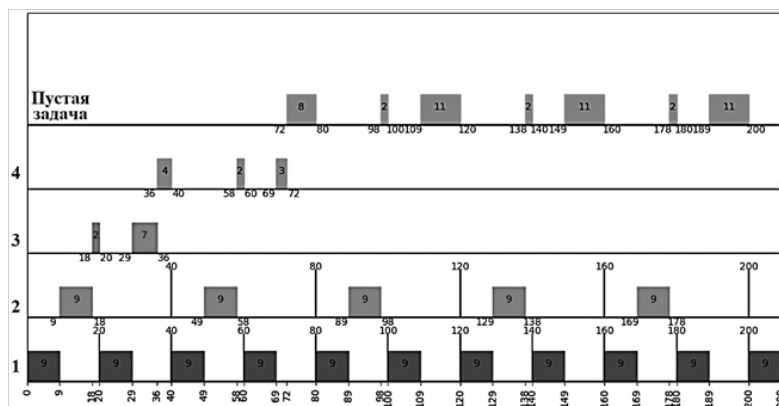


Рисунок 5 – Результат работы алгоритма Background

После анализа полученных данных резюмируем, что алгоритм функционирует корректно, т.к. аperiodические запросы выполняются в моменты времени, когда отсутствуют периодические задачи.

Отличительные черты, присущие этому алгоритму, прослеживаются с 0 по 72 нс. В начале управление получают первая и вторая периодические задачи до 18 нс. Затем начинают выполняться аperiodические задачи в оставшихся временных «окнах». С 18 нс по 20 нс и с 29 нс по 36 нс выполняется аperiodический запрос (3-я задача), после выполнения которого управление получает вторая аperiodическая задача (4-я задача) во временных промежутках с 36 нс по 40 нс, с 58 нс по 60 нс и с 69 нс по 72 нс, когда отсутствуют периодические задачи.

Для оценки длительности переключения контекста проведены эксперименты с 1, 3, 5, 7 и 9 задачами. Результаты оценки этих параметров можно увидеть на рисунке 6.

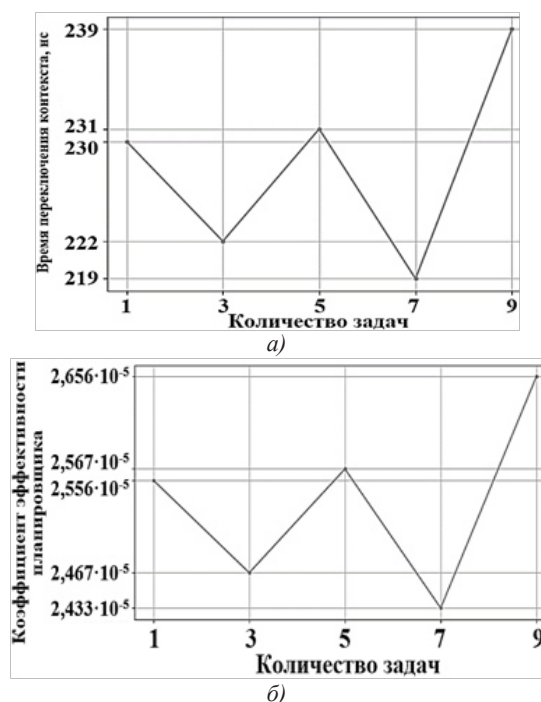


Рисунок 6 – Зависимость времени переключения контекста от количества задач (а); коэффициента эффективности планировщика от количества задач (б) алгоритма Background

По рисунку 6,а определим, что время переключения контекста для алгоритма Background находится в промежутке от 219 нс до 239 нс и среднее время составляет 228 нс. По рисунку 4,б определим среднее значение коэффициента эффективности планировщика, которое составило $2,536 \cdot 10^{-5}$.

Сравнение алгоритмов. Для сравнения приведенных выше алгоритмов использованы данные, полученные при анализе каждого из исследованных и приведенных в работе из алгоритмов. Эксперименты проводились для 1, 3, 5, 7 и 9 задач с целью определения времени переключения контекста и коэффициента эффективности планировщика.

Анализируя полученные в работе данные, можно сказать о том, что для алгоритмов EDF и Background, изменение количества задач, выполняющихся в системе, никак не влияет на время переключения контекста.

Также был проанализирован вопрос изменения коэффициента эффективности планировщика. Получено, что коэффициент эффективности для RM, EDF и Background является постоянным.

После проведения экспериментов составлена таблица 1, в которой отображаются характерные черты рассмотренных алгоритмов планирования и временные параметры, полученные в ходе исследования. В таблице приняты обозначения переменных: Sw – среднее время переключения контекста; Eff – среднее значение коэффициента эффективности планировщика.

Таблица 1 – Сравнение исследуемых алгоритмов планирования

Алгоритм	Тип	Макс. загрузка	Sw, нс	Eff
Rate Monotonic	Статический	$n \cdot \left(2^{\frac{1}{n}} - 1\right)$	201	$2,236 \cdot 10^{-5}$
Earliest Deadline First	Динамический	<100%	221	$2,46 \cdot 10^{-5}$
Background	Динамический	$n \cdot \left(2^{\frac{1}{n}} - 1\right)$	228	$2,536 \cdot 10^{-5}$

Анализируя представленные в таблице 1 значения, можно сказать, что наиболее оптимальным алгоритмом для планирования периодических задач при значениях загрузки меньше чем $n \cdot \left(2^{\frac{1}{n}} - 1\right)$, где n – число задач, является алгоритм Rate Monotonic, поскольку он имеет наименьшее время переключения контекста

и наименьшее значение коэффициента эффективности планировщика. Для планирования периодических задач при значениях загрузки больше, чем $n \cdot \left(\frac{1}{2^n} - 1\right)$ оптимальным выбором является алгоритм *Earliest Deadline First*. По полученным данным видно, что для планирования аperiodических задач оптимальным выбором является алгоритм *Background*, но только в том случае, если аperiodические задачи в системе не имеют жестких сроков (системы мягкого реального времени).

Закключение. Исследования алгоритмов планирования операционных систем реального времени производились на основе двух показателей – длительность переключения контекста и коэффициент эффективности планировщика. Проведены эксперименты с разным количеством задач для определения тенденции изменения каждого из параметров. В результате получены и представлены рекомендации для выбора определенного алгоритма планирования при конкретных типах задач и значениях загрузки.

СПИСОК ЛИТЕРАТУРЫ:

1. Данилов М.В. Методы планирования выполнения задач в системах реального времени // Программные продукты и системы. – 2001. – № 4. – С. 28-35
2. Методы планирования выполнения задач в системах жесткого реального времени: автореферат дис. ... кандидата технических наук: 05.13.11 / С.-Петербург. ин-т информатики и автоматизации РАН. – Санкт-Петербург, 2002. – 16 с.
3. Бурдонов, И. Б. Операционные системы реального времени / И.Б. Бурдонов, А. С. Косачев, В.Н. Пономаренко // Препринт Института системного программирования РАН. – 2006. – №14. – С. 3.
4. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. Х. Бос. 4-е изд. – Санкт-Петербург: Питер, 2015. – 602 с.
5. Audsley, N. C. Applying new scheduling theory to static priority preemptive scheduling / N.C. Audsley, A. Burns, M. Richardson, K. Tindell, A. Wellings // Software Engineering Journal. – 2002. – Vol. 8. - № 5. – P. 1.
6. Мартышкин А.И. Анализ наиболее перспективных алгоритмов планирования операционных систем реального времени // Современные инновационные технологии подготовки инженерных кадров для горной промышленности и транспорта. 2020. № 1 (7). С. 278-287.
7. Мартышкин А.И. Современные направления развития встраиваемых операционных систем // Современные инновационные технологии подготовки инженерных кадров для горной промышленности и транспорта. 2018. № 1 (5). С. 262-271.
8. Panigrahi R. L. Real Time System for Software Engineering: An Overview / R. L. Panigrahi and M.K. Senapaty // Global Journal for Research Analysis. – 2014. – Vol. 3. – Issue 1. – P. 25-27.
9. Alan Burns and Andy Wellings (2009), Real-Time Systems and Programming Languages (4th ed.), Addison-Wesley, ISBN 978-0-321-41745-9
10. Bini, E. The space of rate monotonic schedulability / E. Bini, G.C. Buttazzo // IEEE Real-Time Systems Symposium. – 2002. - №23. – P. 1.
11. Buttazzo G.C. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (Real-Time Systems Series) / G.C. Buttazzo. – 3rd ed. – London: Springer, 2011. – P. 130, P. 143.
12. Мартышкин А.И. Исследование алгоритмов планирования процессов в системах реального времени. Современные методы и средства обработки пространственно-временных сигналов: сборник статей XIII Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. 2015. С. 118-124.
13. Carraro E. Implementation and Test of EDF and LLREF Schedulers in FreeRTOS // Site Regensburg University of Applied Sciences – 2016 [Электронный ресурс]. URL: http://tesi.cab.unipd.it/51896/1/Implementation_and_Test_of_EDF_and_LLREF_Scheduler_in_FreeRTOS.pdf (дата обращения: 12.10.2016).
14. Marko Bertogna M. C. Improved schedulability analysis of edf on multiprocessor platforms / M.C. Marko Bertogna, G. Lipari // Euromicro Conference on Real-Time Systems. – 2005. – №17. – P. 2.
15. Мартышкин А.И. Программа для измерения производительности некоторых функций операционных систем. Свидетельство о регистрации программы для ЭВМ RU 2016619353, 18.08.2016. Заявка № 2016616799 от 27.06.2016.
16. Мартышкин А.И. Программа для расчета основных вероятностно-временных характеристик подсистемы планирования и диспетчеризации многопроцессорных систем. Свидетельство о регистрации программы для ЭВМ RU 2016619353, 18.08.2016. Заявка № 2016616801 от 27.06.2016.
17. GitHub - gyaesu/STM32F407G-DISC1: Learning to programme the STM32F4 microprocessor via the Embedded.fm tutorial series [Электронный ресурс]. URL: <https://github.com/gyaesu/STM32F407G-DISC1> (дата обращения: 13.11.2021).
18. Курниц, А. FreeRTOS – операционная система для микроконтроллеров / А. Курниц // Компоненты и технологии. – 2011. – №2. – С. 97.
19. Kase R. Efficient Scheduling Library for FreeRTOS // Site Digital Scientific Archive in Sweden and Countries of Northern Europe – 2016 [Электронный ресурс]. URL: <http://www.diva-portal.org/smash/get/diva2:1085303/FULLTEXT01.pdf> (дата обращения: 23.11.2016).
20. Liu C. L. Real-time systems / C. L. Liu. - Albuquerque, New Mexico: Integre Technical Publishing Co, 2000. – P. 69, P. 192.

Исследование выполнено за счет гранта Российского научного фонда № 21-71-00110, <https://rscf.ru/project/21-71-00110/>.

Статья поступила в редакцию 01.11.2021

Статья принята к публикации 07.12.2021