

УДК 004.031.6, 004.031.43

**В.А. Святный** (д-р техн. наук, проф.), **Ю.А. Иванов** (ассист.)  
Донецкий национальный технический университет  
[svjatnyj@cs.dgtu.donetsk.ua](mailto:svjatnyj@cs.dgtu.donetsk.ua), [yuriy.o.ivanov@gmail.com](mailto:yuriy.o.ivanov@gmail.com)

## **УПРАВЛЕНИЕ ЦИКЛОМ РЕАЛЬНОГО ВРЕМЕНИ ВСТРОЕННЫХ СИСТЕМ ПРИ МОДЕЛИРОВАНИИ**

Исследованы вопросы реализации алгоритмов планирования и управления систем моделирования на базе микроконтроллерных систем. Получены задержки при переключении и запуске процессов для NEEK. Проведено сравнение характера реактивности данной системы со стандартным цифровым процессором.

**моделирование, микроконтроллер, алгоритм, измерение, цикл реального времени**

### ***Введение***

Средства модельной поддержки и сопровождения разработок систем управления критичных по безопасности функционирования сложных динамических объектов во всех сферах человеческой деятельности становятся решающим фактором качества и конкурентоспособности проектов в аэрокосмическом комплексе, химических технологиях, спутниковой, инерционной, радионавигации и др.

Одним из перспективных направлений создания систем моделирования реального времени, в том числе и полунатурных, является разработка их на основе современных микроконтроллерных систем, поскольку их архитектура имеет ряд достоинств: высокая производительность, высокий уровень надежности, портативность, относительная дешевизна.

### ***Организация цикла реального времени в системе моделирования***

Микроконтроллерная система должна принимать информацию от объекта, преобразовывать ее в форму, удобную для хранения и обработки, обрабатывать по заданным алгоритмам и посылать результаты обратно по информационным каналам к объекту. Этот набор действий выполняется циклически на протяжении всего времени моделирования и является циклом реального времени.

Особенностью цикла есть безусловное требование к реактивности действий, которые бы отвечали заданным критериям реального времени

исполнения. Таким образом, программно-аппаратные затраты времени, возникающие при передаче информации и работе планировщика в цикле реального времени, должны быть минимизированы.

Модель вычислительного процесса предполагает выполнение следующих действий: посредством датчиков в систему моделирования вводятся вычисляемые параметры  $Z = [z_1, z_2, \dots, z_n]$ ; их количество строго определено и не меняется на протяжении цикла моделирования; вычисление  $i$ -параметра должно быть обязательно выполнено за период  $T_i$ ; задачи выполняются за время  $\tau_i$ , которое определено на этапе построения модели.

Большинство существующих алгоритмов управления циклом реального времени не учитывают многочастотность моделируемого объекта. В работах [1] и [2] предложены алгоритмы планирования вычислений для исследования многочастотных задач. Теоретические вопросы существования циклического расписания были рассмотрены ранее в [3], где сформулированы критерии существования цикла реального времени. Рассмотрим применение алгоритма равномерного планирования запуска задач согласно требуемых частот выполнения [4] для разработки диспетчера запуска задача микроконтроллерной системы. Особенностью предложенного алгоритма управления является одинаковое по времени вычисление каждой переменной моделирования на каждом из циклов реального времени. Циклограмма реального времени представляет размещение времен и порядок выполнения задач ( $\tau_i$ ), задержек на их запуск и переключение между задачами ( $d_i$ ) (рис. 1). Каждая вычисляемая переменная занимает  $\frac{L\tau_i}{T_i}$  времени цикла.

Для предложенного алгоритма условие существования цикла реального времени будет описываться условием

$$\frac{T_e}{L} \sum_{i=1}^n \frac{L\tau_i}{T_i} \leq T_e, \quad (1)$$

где  $L = \text{НОД} \{T_i\}$  – участок, на котором будет выполнена часть каждой задачи.

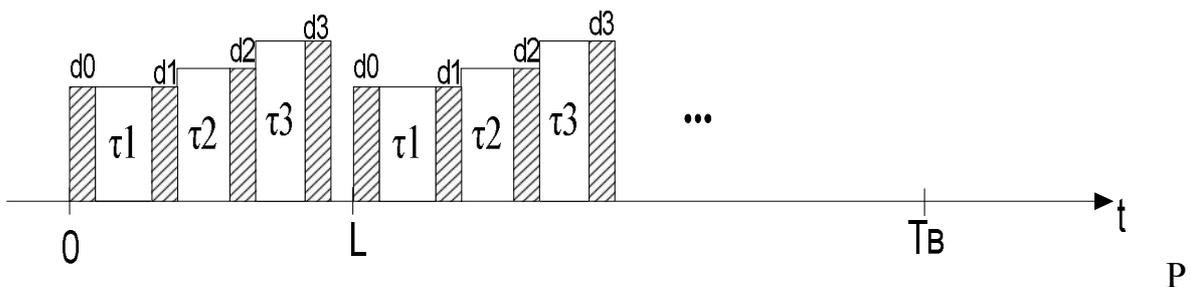


Рисунок 1 – Циклограмма запуска задач

Таким образом, согласно размещению циклограммы на оси времени затраты диспетчеризации задач  $t_d$  при данной организации вычислительного процесса в каждом периоде  $L$  составят

$$t_d = d_0 + \sum_1^n d_i, \quad (2)$$

где  $d_0$  – время, необходимое для запуска первой задачи цикла,  $d_i$  – время переключения между задачами. Тогда за все время выполнения  $T_e$  на диспетчеризацию будет затрачено:

$$T_d = \frac{T_e}{L} t_d. \quad (3)$$

Если считать время переключения с одной задачи на другую одинаковым, поскольку используется единый механизм переключения, то суммарное время на диспетчеризацию будет составлять

$$T_d = \frac{T_e}{L} (d_0 + n \cdot d). \quad (4)$$

Отсюда следует, что важно исследовать параметр  $d$  для микроконтроллерной системы.

Одновременно с этим из (1) и (4) может быть определено полное значение нагрузки процессора  $\rho$  на периоде выполнения как сумма затрат времени на диспетчеризацию и вычисления:

$$\rho = \rho_d + \rho_z = T_e \left( \frac{d_0 + n \cdot d_i}{L} + \sum_{i=1}^n \frac{\tau_i}{T_i'} \right), \quad (5)$$

где  $T_i'$  – приведенный период вычисления к условию  $L = \text{НОД} \{T_i\}$ .

Оценка эффективности данной организации цикла реального времени может быть определена как максимально допустимое значение функции (5), представляющей потенциальную нагрузочную характеристику (ПНХ) [1]. Рабочему режиму системы моделирования на плоскости характеристик нагрузки можно поставить в соответствие точку с координатами:

$$C_{PT} = M[C_T], \quad (6)$$

$$\rho_{PT} = M[\rho_T], \quad (7)$$

где  $C_{PT}$  – среднее значение интенсивности запуска задач;  $\rho_{PT}$  – средняя загрузка процессора задачами. Эта точка называется рабочей точкой (РТ). Величины  $C_T$  и  $\rho_T$ , которые стоят под знаком математического ожидания в (6) и (7), характеризуют параметры рабочей нагрузки процессора. Исходя из этого, ПНХ можно определить как траекторию гранично допустимых положений РТ (рис.2) [1]. Таким образом, для каждого значения интенсивности  $C$  справедливо уравнение, которое характеризует отношение затрат процессорного времени

$$\rho_{PT}(C) + \rho_d(C, R_{PT}) + R_{PT}(C) = M, \quad (8)$$

где  $C$  – текущая абсцисса РТ;  $\rho_{PT}(C)$  – траектория РТ;  $\rho_d(C, R_{PT})$  – затраты на работу диспетчера цикла реального времени системы моделирования.

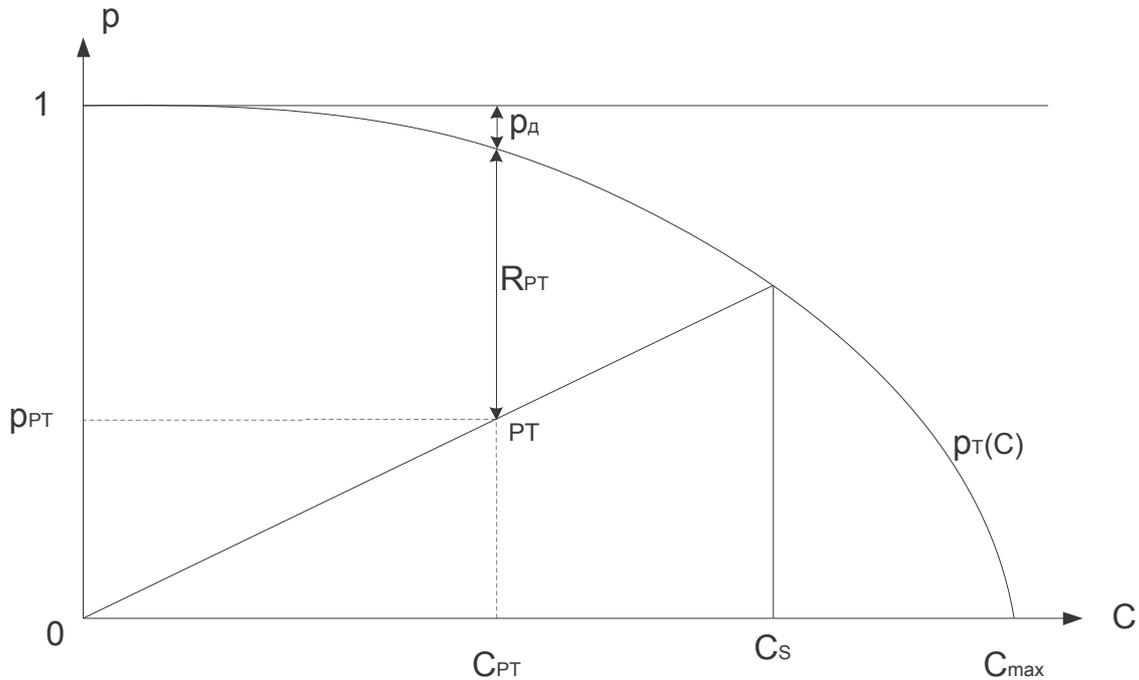


Рисунок 2 – Потенциальная нагрузочная характеристика

Для каждого значения  $C$  РТ достигает своего граничного значения при  $R_{PT}(C) = 0$  и тогда уравнение (8) примет вид (5).

Из уравнения ПНХ (8) следует, что одним из направлений повышения эффективности вычислительного процесса является оптимизации диспетчеризации, т.е. уменьшения использования на нее процессорных ресурсов. Применимо к данной задаче оптимизация предполагает уменьшение задержек  $d_0$  и  $d_i$  в (5).

### **Исследование динамических характеристик встроенных систем**

Для исследования управления цилом реального времени был выбран отладочный FPGA-комплекс «NIOS II Embedded Evaluation Kit» (NEEK), так как он удовлетворяет основным требованиям к проектированию подсистем нижнего уровня [2].

Было проведено исследования NEEK и были разработаны тестовые программы для получения экспериментальных значений величин  $d_0$  и  $d_i$ . Разработка была проведена на базе встроенной операционной системы

реального времени MicroC/OS-II, многозадачность ядра которой оптимизирована для микроконтроллерных систем.

Эксперименты были выполнены с двумя типами программ. Во-первых, разработана тестовая программа, состоящая из двух задач, которые имеют идентичные настройки и запускаются синхронно. Блок-схема программы приведена на рис. 3. Функции задач представляют собой бесконечный цикл с небольшой задержкой и выводом текущего значения счетчика тактов.

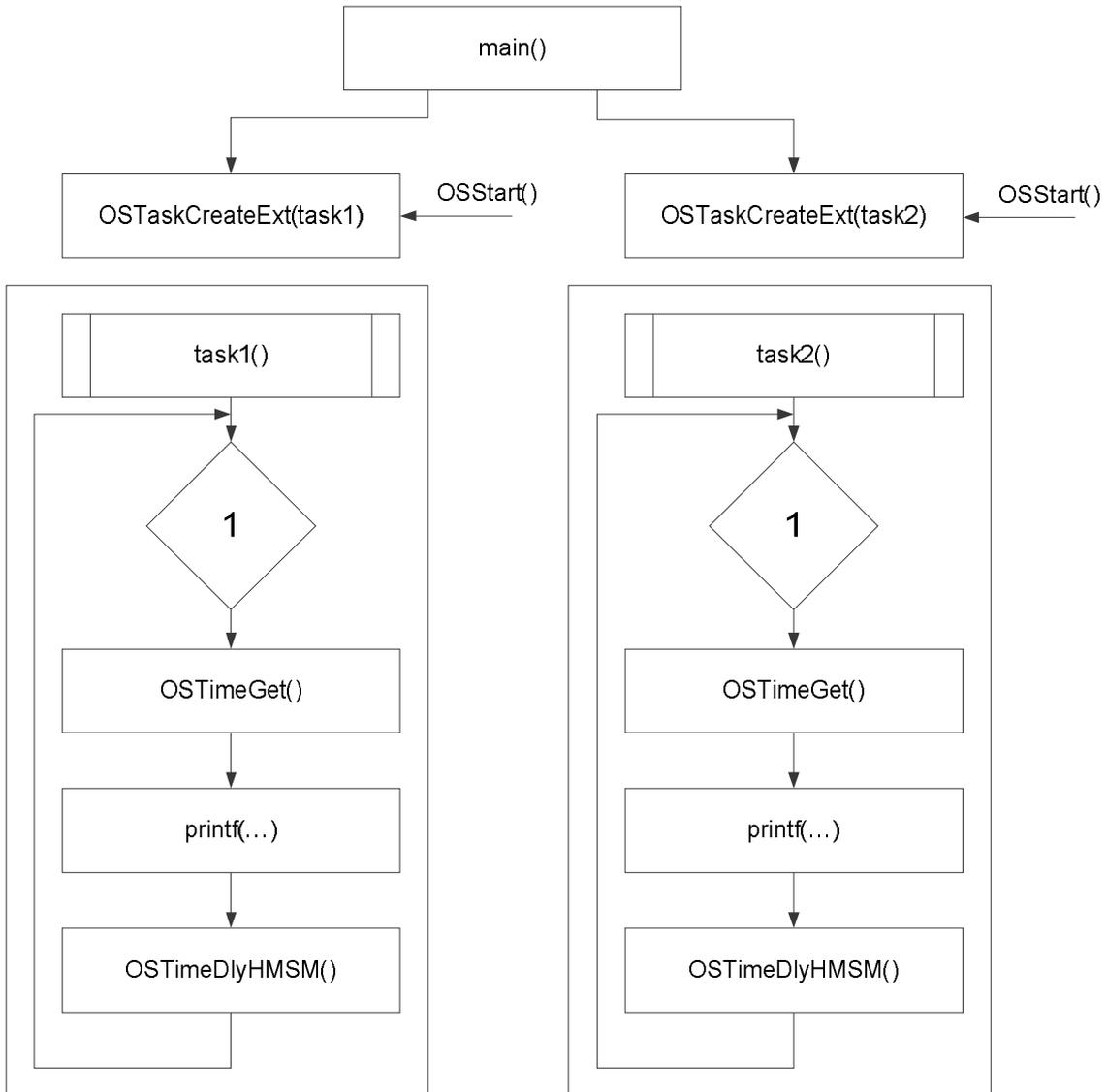
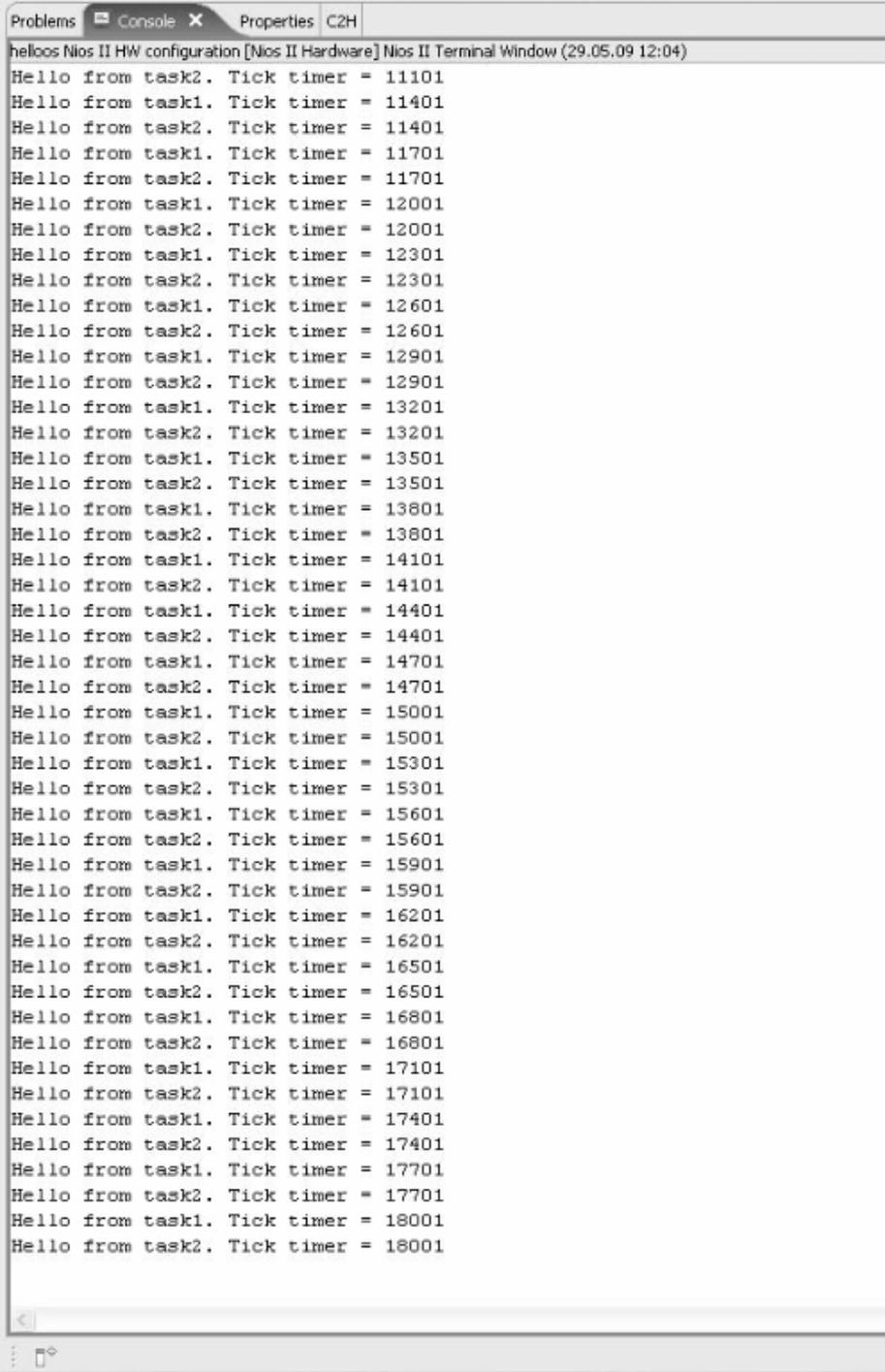


Рисунок 3 – Блок-схема программы выполнения двух потоков

По результатам выполнения (фрагмент представлен на рис. 4), можно сделать вывод о правильности функционирования мультизадачной системы. Задачи исполнялись с одинаковым периодом 3с, при этом обе задачи получали управления в один такт системного времени, что свидетельствует о времени переключения между потоками меньше такта

системного таймера. Это может быть объяснено спецификой SoPC-процессора и оптимизированностью системы реального времени.



```
Problems Console X Properties C2H
helloos Nios II HW configuration [Nios II Hardware] Nios II Terminal Window (29.05.09 12:04)
Hello from task2. Tick timer = 11101
Hello from task1. Tick timer = 11401
Hello from task2. Tick timer = 11401
Hello from task1. Tick timer = 11701
Hello from task2. Tick timer = 11701
Hello from task1. Tick timer = 12001
Hello from task2. Tick timer = 12001
Hello from task1. Tick timer = 12301
Hello from task2. Tick timer = 12301
Hello from task1. Tick timer = 12601
Hello from task2. Tick timer = 12601
Hello from task1. Tick timer = 12901
Hello from task2. Tick timer = 12901
Hello from task1. Tick timer = 13201
Hello from task2. Tick timer = 13201
Hello from task1. Tick timer = 13501
Hello from task2. Tick timer = 13501
Hello from task1. Tick timer = 13801
Hello from task2. Tick timer = 13801
Hello from task1. Tick timer = 14101
Hello from task2. Tick timer = 14101
Hello from task1. Tick timer = 14401
Hello from task2. Tick timer = 14401
Hello from task1. Tick timer = 14701
Hello from task2. Tick timer = 14701
Hello from task1. Tick timer = 15001
Hello from task2. Tick timer = 15001
Hello from task1. Tick timer = 15301
Hello from task2. Tick timer = 15301
Hello from task1. Tick timer = 15601
Hello from task2. Tick timer = 15601
Hello from task1. Tick timer = 15901
Hello from task2. Tick timer = 15901
Hello from task1. Tick timer = 16201
Hello from task2. Tick timer = 16201
Hello from task1. Tick timer = 16501
Hello from task2. Tick timer = 16501
Hello from task1. Tick timer = 16801
Hello from task2. Tick timer = 16801
Hello from task1. Tick timer = 17101
Hello from task2. Tick timer = 17101
Hello from task1. Tick timer = 17401
Hello from task2. Tick timer = 17401
Hello from task1. Tick timer = 17701
Hello from task2. Tick timer = 17701
Hello from task1. Tick timer = 18001
Hello from task2. Tick timer = 18001
```

Рисунок 4 – Результаты программы выполнения двух потоков

Для определения задержки, вносимой функциями измерения времени и подготовки задач, была разработана программа, которая аналогична первой задаче, но состоит из одной задачи и содержит минимально допустимую задержку, равную 1 мс. Были получены

результаты вида: Tick timer = 1 ... Tick coun timer = 1895 – всего 983 вычисления. Характер изменения времени для данного эксперимента иллюстрирует рис. 5.

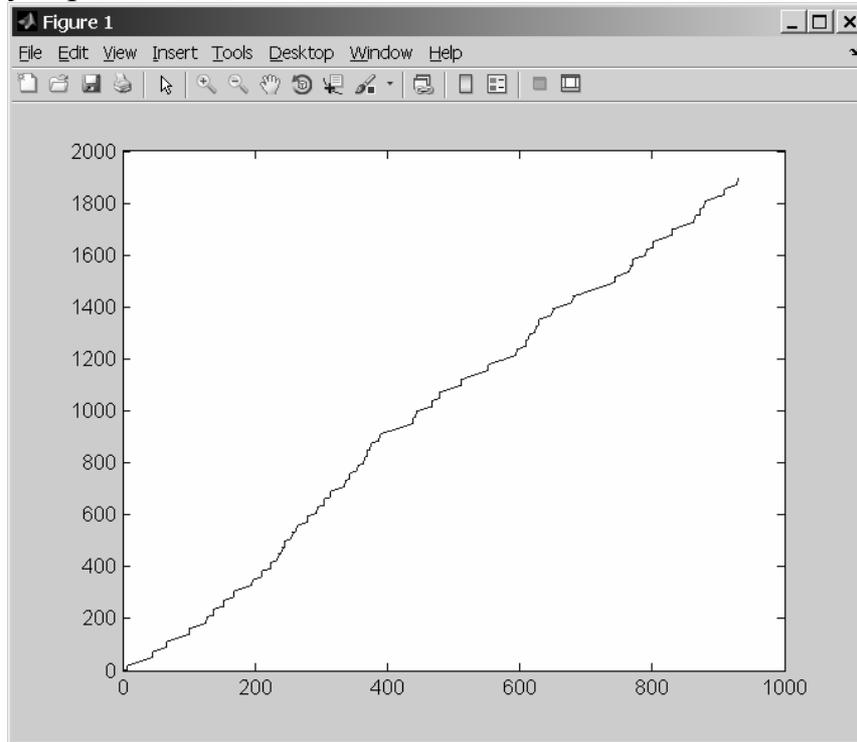


Рисунок 5 – Результаты определения задержек

На графике по оси Y отображена задержка, которая отвечает  $i$ -му вызову задачи, что отображено на оси X. Безусловно, на линии графика есть определенные перепады, которые можно объяснить влиянием контроля MicrОC/OS-II и HAL. Но этот график имеет более линейный вид, а также не содержит резких скачков, характерных для аналогичных исследований различных операционных систем персонального компьютера [5].

Проведенные эксперименты показали, что в отдельных случаях задержка на переключение процессов может составить не более 1 мс. Поскольку в задачах систем реального времени самое главное – это гарантия временных интервалов, предложено для системы NEEK рассматривать наихудшую ситуацию, тогда (5) примет вид (мс):

$$\rho = T_s \left( \frac{1+n}{L} + \sum_{i=1}^n \frac{\tau_i}{T_i} \right). \quad (9)$$

Полученная величина задержки является малой относительно тактовой частоты системного таймера микроконтроллерной системы. Таким образом, данные системы для исследуемых задач моделирования можно считать эффективными, что объясняется ПНХ для NEEK. За счет уменьшения величины  $\rho_d(C, R_{PT})$  в (8) кривая нагрузочной характеристики

будет смещена вверх и вправо, что свидетельствует о более эффективной организации вычислительного процесса при равномерном алгоритме планирования цикла реального времени.

### **Управление циклом реального времени микроконтроллерной системы**

Опираясь на полученные данные, была разработана программа диспетчеризации цикла реального времени для системы NEEK (рис. 6).

```

Nios II C/C++ - hello_ucosii.c - Nios II IDE
File Edit Refactor Navigate Search Project Tools Run Window Help
<terminated> hello_ucosii_0 Nios II HW configuration [Nios II Hardware] Nios II Terminal Window (05.01.10 12:35)
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "USB-Blaster [USB-0]", device 1, instance 0
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)

Program started...
SVZ started...
-----L-----
15
SVZ started...
Thread 0 started...
Thread 0 completed.
SVZ started...
Thread 1 started...
Thread 1 completed.
-----L-----
Iteration 1: x1=1.000000 y1=-0.125000 | x2=1.000000 y2=-0.000000
39
Thread 0 started...
Thread 0 completed.
Thread 1 started...
Thread 1 completed.

```

Рисунок 6 – Работа системы NEEK в режиме отладки

В программе использованы функции системных библиотек:

- OSTaskCreateExt. Функция создает задачу с установленными параметрами: адресом функции обработки, памятью задачи, приоритетом, порядковым номером, размером стека, указателем на блок управления задачами и дополнительными опциями.
- OSStart. Функция запускает на исполнение систему.
- OSTimeGet. Функция возвращает значения 32-битного счетчика

- тиков системного таймера.
- OSTimeDlyHMSM. Функция устанавливает задержку на необходимое количество часов, минут, секунд, миллисекунд.

Выполнены отладка и профилирование программы управления циклом реального времени.

Для исследования алгоритма планирования и управления вычислениями было использовано численное интегрирование системы дифференциальных уравнений, описывающих модель гармонического осциллятора:

$$\begin{cases} x_1' = x_2 \\ x_2' = -\omega^2 x_1 \end{cases} \quad (10)$$

Для моделирования двух гармонических осцилляторов с разными частотами был использован численный алгоритм Эйлера. Один из осцилляторов моделировался программным потоком с частотой 0,5 рад/сек, а второй – другим потоком с частотой 0,001 рад/сек (рис. 7), которые соответствуют частотам реально возможных сигналов возмущений, характерных для системы проветривания угольной шахты.

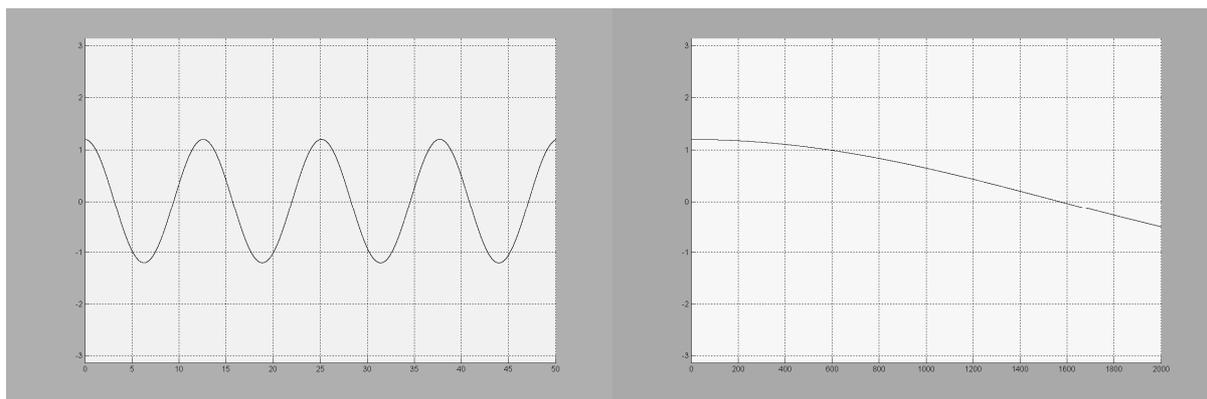


Рисунок 7 – Одновременное моделирование двух возмущений

## **Выводы**

В статье исследована возможность создания средств моделирования на базе микроконтроллерных систем. Получены временные задержки при переключении и запуске процессов для NEEK, которые составили 1 мс. Реактивность данной системы по сравнению со стандартным цифровым процессором отличается линейностью и отсутствием резких перепадов.

Для микроконтроллерной системы разработана программа управления вычислительным процессом и протестирована на примере модели гармонического осциллятора. Анализ ПНХ показывает, что

применение микроконтроллерных систем для реализации полунатурных средств моделирования является эффективным.

Одним из направлений развития результатов работы является создание на основе универсального планировщика библиотеки циклов реального времени различных динамических систем, опирающихся на технологии FPGA и SOPC-процессоров.

### **Список литературы**

1. Виттих В.А. Оптимизация бортовых систем сбора и обработки данных / В.А. Виттих, В.А. Цыбатов. – М.:Наука,1985. – 175 с.:ил., с. 20-25.
2. П'явченко О.Н. Розподілені мікрокомп'ютерні системи збору й обробки інформації. Навчальний посібник. Частина 1 / О.Н.П'явченко. – ТРТУ.: Таганрог, 2007.
3. Гильман А.Л. Расписания в задачах организации периодической обработки информации / А.Л. Гильман, Я.Г. Хаит. – Харьков, 1970.
4. Святный В.А. Алгоритм организации цифровых вычислений в цикле реального времени АЦВС // А.Ю. Иванов, В.А. Святный. - Вопросы радиоэлектроники .Сер.ЭВТ. Вып.2,1986г.
5. Иванов Ю. А. «Измерение накладных расходов в операционной системе для построения системы моделирования процессов реального времени», Проблемы моделювання та автоматизації проектування динамічних систем. / Наукові праці Донецького національного технічного університету. Серія "Проблеми моделювання та автоматизації проектування динамічних систем". Випуск 7 (150)- Донецьк: ДонНТУ – 2008, с. 158-176.

*Надійшла до редакції 02.09.2009 Рецензент: к.т.н., доц. Краснокутський В.О.*

**В.А. Святный, Ю.О. Иванов**

Донецький національний технічний університет

**Керування циклом реального часу вбудованих систем при моделюванні.** Досліджені питання розробки та реалізації циклічних алгоритмів планування та керування для систем моделювання на базі мікроконтролерів, орієнтовані на дослідження жорстких динамічних систем. Отримані затримки при перемиканні та запуску процесів для NEEK комплексу. Проведено порівняння реактивності даної системи з процесором стандартного персонального комп'ютера.

**моделювання, мікроконтролер, алгоритм, вимір, цикл реального часу**

**V.A. Svyatnyy, Y.A. Ivanov**

Donetsk National Technical University

**Real-time process management in modeling for embedded systems.** Development scheduling cyclic algorithms and management modeling systems for embedded systems were analyzed for stiff dynamic systems. Time intervals of delays occurred in processes switching and management were obtained for NEEK. Responsiveness type of this system was compared to standard digital processor.

**modeling, microcontroller, algorithm, measurement, real-time loop**