

## Impact Factor:

ISRA (India) = 6.317  
ISI (Dubai, UAE) = 1.582  
GIF (Australia) = 0.564  
JIF = 1.500

SIS (USA) = 0.912  
ПИИИ (Russia) = 0.126  
ESJI (KZ) = 9.035  
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630  
PIF (India) = 1.940  
IBI (India) = 4.260  
OAJI (USA) = 0.350

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

### International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2021 Issue: 08 Volume: 100

Published: 13.08.2021 <http://T-Science.org>

QR – Issue



QR – Article



**Tatyana Viktorovna Pestrikova**  
Pacific National University  
lecturer of Computer Engineering  
(PNU)

**Ekaterina Dmitrievna Birukova**  
Pacific National University  
student,  
KST(ab)-91

## ReTi - REAL-TIME OPERATING SYSTEM FOR MICROCONTROLLERS

**Abstract:** The aim of the work is the development and implementation of a real-time operating system for microcontrollers, the main difference of which is the presence of only basic functions, small, in comparison with existing operating systems, volume, combined with much higher performance. To achieve this goal, an automaton paradigm for writing programs was chosen with the imposition of a number of restrictions.

**Key words:** FABRIK-algorithm, linear motor, Denavit-Hartenberg representation, kinematic chain, distal phalanx, direct and inverse problems of kinematics.

**Language:** Russian

**Citation:** Pestrikova, T. V., & Birukova, E. D. (2021). ReTi - real-time operating system for microcontrollers. *ISJ Theoretical & Applied Science*, 08 (100), 165-168.

**Soi:** <http://s-o-i.org/1.1/TAS-08-100-31> **Doi:**  <https://dx.doi.org/10.15863/TAS.2021.08.100.31>

**Scopus ASCC:** 2204.

### ReTi – ОПЕРАЦИОННАЯ СИСТЕМА РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ МИКРОКОНТРОЛЛЕРОВ

**Аннотация:** Целью работы является разработка и реализация операционной системы реального времени для микроконтроллеров, главным отличием которой является наличие только основных функций, малый, в сравнении с существующими операционными системами, объем, в сочетании с гораздо большим быстродействием. Для достижения поставленной цели была выбрана автоматная парадигма написания программ с наложением ряда ограничений.

**Ключевые слова:** FABRIK-algorithm, linear motor, Denavit-Hartenberg representation, kinematic chain, distal phalanx, direct and inverse problems of kinematics.

#### Введение

ОСРВ – это операционная система для устройств и систем, которым необходимо быстро реагировать на триггер. В случае отказоустойчивого программного обеспечения, например, ОСРВ будет откладывать процессы с более низким приоритетом, чтобы позаботиться о задаче с более высоким. При наличии нескольких процессов и устройств, когда отдаётся предпочтение времени выполнения работы, нежели производительности, ОСРВ будет

необходима. Другими словами, ОСРВ используется при необходимости в запуске нескольких процессов в определенное время. Она гарантирует выполнение поставленных задач с меньшей задержкой и точно определяет ее выполнение. ОСРВ позволяет выполнять как срочные по времени задачи, так и некритичные.

Использование ОСРВ приводит к определенным расходам, что может включать, как денежные траты на смену микроконтроллера по причине малой мощности, так и из-за недостатка

## Impact Factor:

ISRA (India) = 6.317  
ISI (Dubai, UAE) = 1.582  
GIF (Australia) = 0.564  
JIF = 1.500

SIS (USA) = 0.912  
РИИЦ (Russia) = 0.126  
ESJI (KZ) = 9.035  
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630  
PIF (India) = 1.940  
IBI (India) = 4.260  
OAJI (USA) = 0.350

памяти программ для хранения ОСРВ ядра, расходов памяти данных на хранение стеков для каждой задачи, очередей, семафоров и иных объектов ядра, временные затраты на переключение между задачами процессора.

Таким образом, целью нашей работы является разработка и реализация операционной системы реального времени, главным отличием которой является наличие только основных функций, необходимых для определенной аппаратуры, и имеющая малый, в сравнении с существующими операционными системами, объем.

Актуальность разработанного решения проблемы заключается в следующем: память, занимаемая ОСРВ, может уменьшиться за счет сокращения библиотеки функций для микроконтроллеров, а значит память для выполнения основных функций может увеличиться. Таким образом, разработанная операционная система позволит выделить больше памяти под выполнение различных задач, а значит будет более востребована в промышленном оборудовании и различной бытовой технике, то есть, в создаваемой библиотеке будет только основной и необходимый набор функций для работы устройств.

### Обзор операционной системы

ReTi является кооперативной неприоритетной системой реального времени. Чтобы управление от одного процесса перешло к другому, нужно, чтобы текущий процесс сам отдал управление системе. Кооперативные планировщики также могут быть приоритетными

и неприоритетными. Примером кооперативной ОС с приоритетным планированием является Salvo ([www.pumpkininc.com](http://www.pumpkininc.com)).

Такт системы – минимальный временной интервал определяемый системой (тактовая частота ОС)

Цикл системы - время выполнения всего цикла из последовательного вызова всех автоматов

### Структура ОС

Система состоит из двух основных частей:

- службы таймеров;
- службы передачи сообщений.

ReTi по сути определяет набор правил, как организовывать задачу, по каким правилам ее строить. Это по факту, модифицированный стиль программирования с одним бесконечным циклом while и прерываниями. ReTi предоставляет удобную службу программных таймеров и механизм передачи сообщений.

Каждая задача представляет собой конечный автомат. Каждый автомат содержит определяемое пользователем количество программных таймеров, а также, определяемое пользователем количество входных и выходных сообщений.

Каждый автомат содержит переменную состояния и предыдущее состояние.

Автомат представляет собой минимальную строительную единицу пользовательского уровня. Его можно представить, как черный ящик с входными и выходными сигналами, как на рисунке 1.

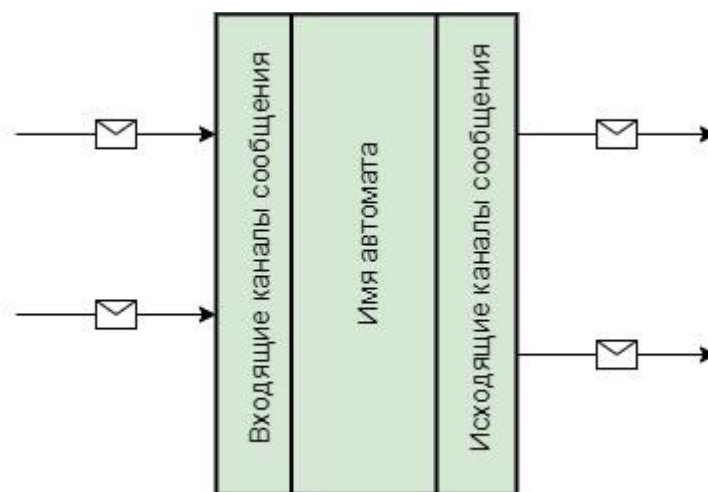


Рисунок 1 – обобщенное представление автомата

Входами и выходами автомата являются сообщения. Каждый автомат может иметь разное количество входных выходных сообщений. Сообщения могут иметь параметры, а могут быть без параметров.

Сообщения без параметров – это аналог дискретного сигнала между микросхемами. Или сигнал есть или его нет. Этот тип сообщений аналогичен понятию «Семафор» в других ОС.

## Impact Factor:

ISRA (India) = 6.317  
ISI (Dubai, UAE) = 1.582  
GIF (Australia) = 0.564  
JIF = 1.500

SIS (USA) = 0.912  
ПИИЦ (Russia) = 0.126  
ESJI (KZ) = 9.035  
SJIF (Morocco) = 7.184

ICV (Poland) = 6.630  
PIF (India) = 1.940  
IBI (India) = 4.260  
OAJI (USA) = 0.350

Сообщения с параметрами – аналог цифрового интерфейса или аналогового сигнала. Кроме факта отправки сообщения у этого сообщения есть блок данных (данные в интерфейсе, уровень аналогового сигнала).

Вся программа на уровне пользователя представляет собой набор таких блоков, входы и выходы которых соединены согласно логике работы системы.

### Служба таймеров

Для работы программных таймеров (ПТ) выделен один аппаратный таймер МК.

Для работы этой службы в системе:

- настроены прерывания аппаратного таймера (АТ) на период 1мС;
- Создана глобальная переменная SoftTimerTick;

- создан массив ПТ, в котором каждый ПТ имеет счетный регистр.

Алгоритм работы прост.

- Каждую 1мС по прерыванию от АТ вызывается функция в которой значение SoftTimerTick увеличивается на 1;

- в начале каждого цикла системы, то есть самой первой функцией в основном цикле while(1), вызывается функция, которая считывает значение SoftTimerTick и прибавляет его к счетным регистрам каждого ПТ.

Для отсчета времени используются 2 функции:

- Функция сброса таймера - void ResetTimer(sys\_timer \* Timer); Эта функция сбрасывает значение счетного регистра в 0. Как будто обнуляется значение секундомера;

- Функция проверки текущего значения счетного регистра unsigned int GetTimer(sys\_timer \* Timer). Эта функция считывает значение мС которое прошло с момента сброса таймера в 0.

### Служба сообщений

Служба сообщений работает по следующим принципам:

1. посланное сообщение активно только в течение одного следующего цикла системы. То есть, автомат посылает сообщение и на следующем цикле системы принимающий автомат может принять сообщение; В данной ОС нет очереди сообщений – все сообщения или приняты в течение следующего цикла – или не приняты вовсе.

2. Есть два типа сообщений: широковещательные и адресные. Широковещательное сообщение может быть прочитано каждым автоматом. Адресное становится неактивным после первого прочтения.

Сообщения без параметров – это аналог дискретного сигнала между микросхемами. Или

сигнал есть или его нет. Этот тип сообщений аналогичен понятию «Семафор» в других ОС.

Сообщения с параметрами – аналог цифрового интерфейса или аналогового сигнала. Кроме факта отправки сообщения у этого сообщения есть блок данных (данные в интерфейсе, уровень аналогового сигнала).

Сообщения могут содержать данные, которые можно извлечь функцией MSG\_DATA \*GetMessageParam(MsgObj \* Msg). Функция возвращает указатель на объект типа MSG\_DATA. Данные представляют собой структуру вида:

```
typedef struct MSG_DATA {  
    unsigned char CommandID;  
    void * data;  
} MSG_DATA;
```

То есть, данные сообщения содержит поле с целочисленным кодом CommandID, и поле с дополнительными данными произвольного типа. В большинстве случаев для обмена информацией между автоматами достаточно просто факта прихода сообщения и численного кода команды. Пример: автомат имеет выходное сообщение с именем FSM\_STATUS\_MSG и в этом сообщении отправляет текущий статус работы автомата, для которого как раз хватает поля CommandID. Значению 1 – соответствует нормальная работа, 2 – режим пониженного потребления, 3 – ошибка.

Использование поля data оправдано при передаче данных. Пример: автомат обрабатывает протокол передачи данных по протоколу UART. Автомат принимает сообщение с данными, обрабатывает заголовок, проверяет контрольную сумму и отправляет уже непосредственно данные другому автомату в виде указателя на массив или структуру, или другой тип данных.

### Заключение

Разработана ОСРВ с самым малым объемом ядра из существующих, при этом по времени системного такта ReTi занимает первое место среди аналогичных. Правила, которые накладываются на стиль программирования задач, при использовании ReTi, позволяют обеспечить жесткий детерминизм разрабатываемого устройства. Сама ОСРВ обладает гораздо меньшим объемом «излишних» функций, и обеспечивает только самые важные: служба таймеров и служба сообщения. Все задачи в системе ReTi являются конечными автоматами. Каждое сообщение или принято в следующем системном такте, или не принято вовсе.

В целом удалось создать кооперативную неприоритетную систему реального времени, которую можно применять для проектов без сложных мультимедийных и сетевых функций. В дальнейшем планируется развить систему до приоритетной.

<b>Impact Factor:</b>	<b>ISRA (India) = 6.317</b>	<b>SIS (USA) = 0.912</b>	<b>ICV (Poland) = 6.630</b>
	<b>ISI (Dubai, UAE) = 1.582</b>	<b>ПИИИ (Russia) = 0.126</b>	<b>PIF (India) = 1.940</b>
	<b>GIF (Australia) = 0.564</b>	<b>ESJI (KZ) = 9.035</b>	<b>IBI (India) = 4.260</b>
	<b>JIF = 1.500</b>	<b>SJIF (Morocco) = 7.184</b>	<b>OAJI (USA) = 0.350</b>

## References:

- (n.d.). *Datasheet MAXIM MAX263-MAX268*
- (n.d.). *Datasheet Analog Devices AD823*
- (n.d.). *Tiva C Series TM4C123G LaunchPad Evaluation Kit ReadMe First*
- Gurlev, D. S. (1966). "Spravochnik po jelektronnym priboram". Kiev: izdatel'stvo "Tehnika".
- Evstifeev, A.V. (2007). "Mikrokontrollery AVR semejstva Mega". Moskva: Dodjeka-XXI.
- Potehin, V.A. (2012). «*Shemotehnika cifrovih ustrojstv*». Tomsk: V-Spektr.
- Kovalenko, S. V. (2014). "Jelektronika", Habarovsk: TOGU.
- Horovic, H. (2014). "Iskusstvo shemotehniki. Binom: Moskva.
- Ch`e, E.U. (2007). "Shemotehnika. Analogovye, impul'snye, cifrovye, analogo-cifrovye ustrojstva". Habarovsk: TOGU.
- Shegal, A. A. (2014). «Primenenie programmogo kompleksa Multisim dlja proektirovanija ustrojstv na mikrokontrollerah». - Ekaterinburg: Izd-vo Ural. un-ta.
- Karter, B., & Manchini, R. (2016). *Operacionnye usiliteli dlja vseh*. per. s angl A.N Rabodzeja, Moscow: DMK Press.