

ОБЗОР МОДЕЛЕЙ ОРГАНИЗАЦИИ ВСТРАИВАЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МЕДИЦИНСКОГО ОБОРУДОВАНИЯ С АВТОНОМНЫМ ПИТАНИЕМ

В.В. Кириенко, Е.С. Семенистая

В данной статье рассмотрены основные модели организации встраиваемого программного обеспечения микроконтроллера, которые используются для создания современной цифровой медицинской техники с автономным питанием. Произведен анализ эффективности каждой модели организации встраиваемого программного обеспечения.

Ключевые слова: система с суперциклом, машина состояний, операционная система реального времени, микроконтроллер, медицинское оборудование, автономное питание.

Актуальность

Для современных медицинских устройств с автономным питанием потребление управляющего микроконтроллера становится все более важным параметром. Применение приборов такого класса требует увеличения срока службы элементов питания в этих устройствах. Это необходимо и для обеспечения требуемого времени диагностического наблюдения, а также с точки зрения удобства их использования. Кроме того уменьшение энергопотребления снижает расходы на эксплуатацию и важно с точки зрения экологии.

В аппаратно-программных комплексах функциональной диагностики, имеющих в своем составе несколько модулей (модуль съема ЭКГ, модуль съема ФПГ, миографический модуль и т.п.), проблема организации питания весьма актуальна.

Общая концепция энергосберегающих технологий в микроконтроллерах заключается в обеспечении уменьшения потребляемой мощности. Мощность, потребляемая микроконтроллером, зависит от режима его работы – в активном режиме она намного больше, чем в спящем режиме [1].

Суммарное же энергопотребление зависит от уровней мощности в каждом режиме и соотношения длительностей этих режимов. Поэтому, чтобы минимизировать энергопотребление микроконтроллера, нужно, с одной стороны, снизить уровни мощности, как в активном, так и спящем режимах, а с другой – уменьшить время работы в активном режиме. Исходя из этого, важным параметром оценки эффективности микроконтроллера с точки зрения расхода энергии является потребляемая мощность в активном режиме. Чем она меньше, тем лучше. Также не менее важным

параметрам является обеспечение малой потребляемой мощности в спящем режиме [1]. Т.к. мощность есть произведение тока на напряжение, то при постоянном напряжении оценку можно производить исключительно по току. Среднее значение потребляемого тока определяет срок службы аккумулятора [2]. На рисунке 1 показано, что для того чтобы обеспечить определенное среднее значение потребляемого тока, микроконтроллер может расходовать большую величину тока в течение коротких интервалов (активный режим), а в остальное время должно иметь меньшее потребление (дежурный режим).

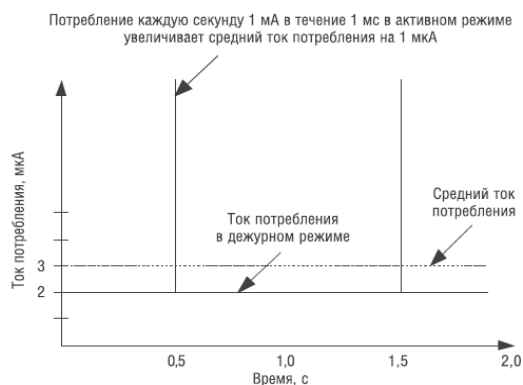


Рисунок 1 – Среднее значение потребляемого тока

Основная часть

Медицинские приборы помимо низкого потребления должны обеспечивать решение следующих задач: цифровой опрос биологических сигналов (ЭКГ, ЭЭГ, ЭМГ, ФПГ), цифровая обработка (цифровая фильтрация), математическая обработка (выделение QRS, подсчет ЧСС, выделение ST-сегмента, спектральный анализ), высокоскоростная запись на твердотельные носители

ОБЗОР МОДЕЛЕЙ ОРГАНИЗАЦИИ ВСТРАИВАЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МЕДИЦИНСКОГО ОБОРУДОВАНИЯ С АВТОНОМНЫМ ПИТАНИЕМ

(SD-карта), беспроводная передача данных (ZigBee, Bluetooth, Wi-Fi), обеспечение интерфейса взаимодействия с пользователем (клавиатура, дисплей).

Рассмотрим варианты построения встраиваемого ПО для микроконтроллера с учетом требований по минимальному энергопотреблению, в процессе работы системы. В практике программирования микроконтроллеров (МК) наиболее часто используются следующие модели организации управляющих программ [3]:

- система с суперциклом (superloop system);
- машина состояний (state machine);
- операционная система реального времени (ОСРВ англ. Real-Time Operating System, RTOS).

Сначала остановимся более подробно на системе с суперциклом [4,5]. Приложение, при такой модели организации, состоит из одного бесконечного цикла и процедур-обработчиков прерываний. В бесконечном цикле в определенной последовательности, в фоновом режиме по отношению к более высокоприоритетным процедурам-обработчикам прерываний, выполняются задачи/действия (модули), как логически выделенные совокупности операторов или как функции, а также обрабатываются данные/события, полученные от обработчиков прерываний. Обработчики прерываний представляют собой специальные процедуры, вызываемые по прерываниям для выполнения их обработки, являясь по сути, низкоуровневыми обработчиками аппаратных событий микроконтроллера. При использовании системы с суперциклом для реализации фрагментов алгоритма управляющей программы, критичных ко времени выполнения, используется система прерываний микроконтроллера. Так как выполнение программного модуля в основном бесконечном цикле не может быть прервано другим модулем иначе, как только через механизм прерываний, время реакции или обращения к конкретному модулю зависит от времени выполнения остальных модулей системы [6]. На рисунке 2 приведен пример обобщенной схемы, иллюстрирующей ход выполнения модулей (задачи/действия) в основном цикле с использованием подпрограмм-обработчиков прерываний, где М1-М5 это программные модули, ОП1-ОП3 обработчики прерываний.

Применительно к специфике снижения энергопотребления применение системы с суперциклом может быть реализовано следующим образом. Т.к. низкое энергопотреб-

ление сводится к задаче снижения среднего значения потребляемого тока, то во все остальное время, пока не происходит опрос данных, их передача или математическая обработка, микроконтроллер должен находиться в спящем режиме, в котором потребление тока максимально снижено. В основной режим переход происходит только по какому-либо прерыванию, такому как нажатие на кнопку или периодическое срабатывание таймера, как показано на рисунке 3. Прерывание обрабатывается, после чего система переходит в спящий режим. В свою очередь переход в основной бесконечный цикл происходит, только если данные или событие от прерывания необходимо обработать в основном цикле. Возможен вариант работы, когда переход в основной цикл вообще не осуществляется, и все получение обработки и передача информации происходит только в обработчиках прерываний [2].

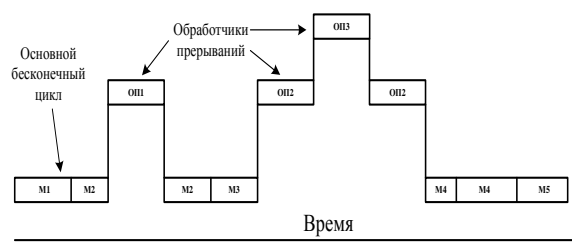


Рисунок 2 - Система с суперциклом

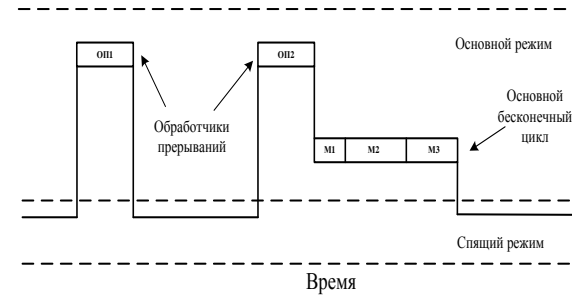


Рисунок 3 - Низкопотребляющий вариант системы с суперциклом

Перейдем к рассмотрению следующей модели реализации встраиваемого ПО, который называется машина состояний. Рассмотрим, как она выглядит на абстрактном уровне. Машина состояний подразумевает простое преобразование входящих событий в соответствующие действия. В любой момент времени такая система находится в одном из нескольких возможных состояний. На рисунке 4 приведен пример обобщенной схемы машины состояний на абстрактном уровне. Пе-

РАЗДЕЛ 5. ПРОГРАММНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И КОМПОНЕНТЫ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Переход из одного состояния в другое происходит в зависимости от внешних условий.

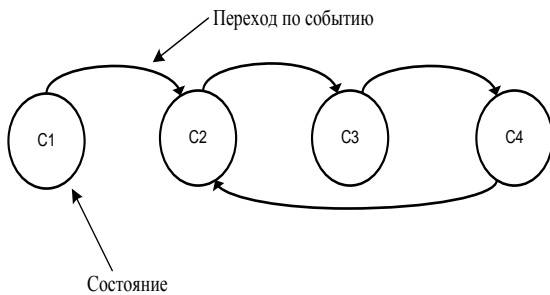


Рисунок 4 - Машина состояний на абстрактном уровне

На прикладном уровне, при использовании такой модели построения встраиваемого ПО, приложение так же, как при использовании системы с суперциклом, состоит из основного бесконечного цикла и процедур обработчиков прерываний. Разница состоит в том, что в основном цикле происходит переключение между несколькими основными ветками, которые соответствуют разным состояниям. На рисунке 5 приведен пример обобщенной схемы на абстрактном уровне, где СхМу - модуль у состояния х. Одно состояние может выполняться циклически сколь угодно долго, пока не произойдет переход в следующее. Переключение может происходить, как по событию или данным из прерывания, так и по какому-либо логическому результату в ветке.

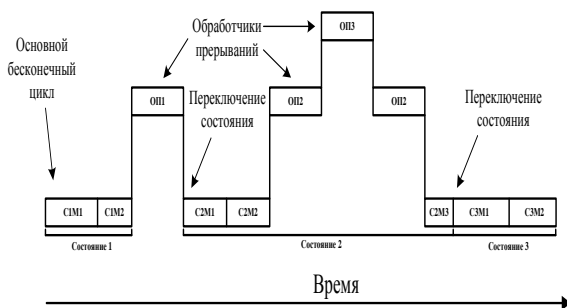


Рисунок 5 - Машина состояний на прикладном уровне

Снижение среднего значения потребляемого тока, при реализации программной модели машины состояний, достигается тем, что одно из состояний такого автомата переводит систему в спящий режим. По какому-либо прерыванию (возникло событие, приняты данные), такая система пробуждается (переходит в основной режим) и последовательно отработывает одно или несколько своих состояний в основном цикле, после чего вызывается состояние погружения в режим по-

ниженного энергопотребления, как показано на рисунке 6.

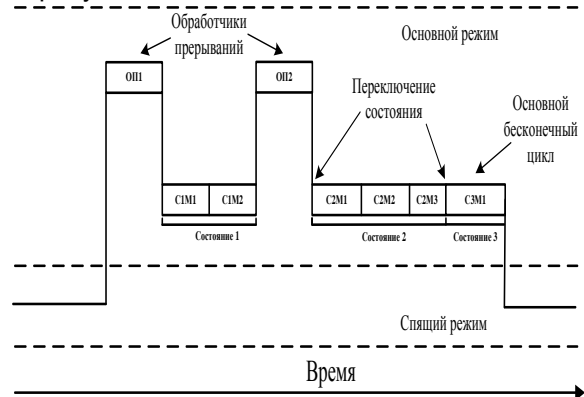


Рисунок 6 - Низкопотребляющий вариант машины состояний

Для задачи создания медицинского оборудования с автономным питанием применение такого подхода связано с определенными ограничениями. Разработка программ, основанных на модели с суперциклом, усложняется по мере увеличения размера управляющего кода. Система с суперциклом, при достаточной сложности возлагаемой на встраиваемое ПО задачи, является неудобным и громоздким инструментом создания больших многозадачных решений, решаемых в реальном времени и требующих последующей доработки, но более удобна для реализации относительно простых алгоритмов, не требовательных ко времени. При этом, снижение потребления энергии микроконтроллером, при данном типе управляющей программы, может быть легко достигнуто, при достаточных вычислительных способностях микроконтроллера.

Модель машины состояний является более удобным инструментом создания встраиваемых программ для медицинского оборудования с автономным питанием чем система с суперциклом. Несмотря на прежнее отсутствие полноценной многозадачности и обеспечения реального времени выполнения действий, поведение такой системы более управляемо, предсказуемо и надежно [3]. При использовании такой системы, проще реализовать проекты средней величины, делать отладку и производить последующую программную доработку. Применение машины состояний может обеспечивать необходимое снижение потребления энергии, как описывалось выше.

Рассмотрим последнюю модель реализации встраиваемого ПО при помощи ОСРВ, которая в полной мере позволяет реализо-

ОБЗОР МОДЕЛЕЙ ОРГАНИЗАЦИИ ВСТРАИВАЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МЕДИЦИНСКОГО ОБОРУДОВАНИЯ С АВТОНОМНЫМ ПИТАНИЕМ

вать многозадачность. ОСРВ – операционная система, в которой успешность работы любой программы зависит не только от её логической правильности, но и от времени, за которое она получила этот результат. Стандарт POSIX 1003.1 даёт такое определение: "Реальное время в операционных системах - это способность операционной системы обеспечить требуемый уровень сервиса в определённый промежуток времени" [4].

Многозадачность – это процесс планирования и переключения процессора между задачами, которые он выполняет. Порядок, в котором выполняются задачи, определяется планировщиком [5-7]. Существует два типа планировщиков: кооперативный (non preemptive) и вытесняющий (preemptive). Большинство систем реального времени используют вытесняющие планировщики, потому что они реагируют быстрее, нежели кооперативные ядра.

При такой модели организации встраиваемого ПО, каждая из выполняемых планировщиком задач имеет свой собственный бесконечный цикл. В RTOS с вытесняющей многозадачностью планировщиком используется периодическое прерывание по таймеру, в обработке которого происходит прерывание выполнения текущей задачи и, при необходимости, переключение задач [5-7]. При этом будет выполняться задача, имеющая наиболее высокий приоритет и находящаяся в состоянии готовности. Если в обработке прерывания другой задаче будет установлен более высокий приоритет, произойдет переключение задач – текущая задача будет остановлена, задача с более высоким приоритетом – запущена [3]. На рисунке 7 приведен пример обобщенной схемы выполнения задач ОСРВ с вытесняющей многозадачностью, где ЗхМу - модуль у задачи х.

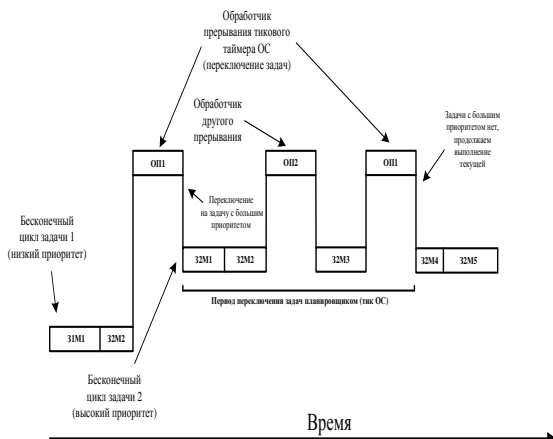


Рисунок 7 - ОСРВ с вытесняющей многозадачностью

Для обеспечения снижения среднего значения потребляемого тока, при реализации программной модели ОСРВ с вытесняющей многозадачностью, в задаче бездействия (Idle task) осуществляется переход системы в спящий режим [6,8]. По прерыванию системного таймера каждый системный тик происходит пробуждение (переходит в основной режим) микроконтроллера, проверяется наличие готовых к выполнению задач. Если есть готовые к выполнению задачи, то они выполняются согласно их приоритету, как показано на рисунке 8. Системный таймер периодически пробуждает систему, хоть и на краткий промежуток времени. Для длительного интервала времени, когда основные задачи не загружены или ожидают внешнего события, целесообразно отключение системного таймера для еще большего снижения среднего потребления. Включение данного таймера происходит в другом обработчике прерывания (часы реального времени, клавиатура), который срабатывает через необходимый интервал времени после его выключения или по событию.

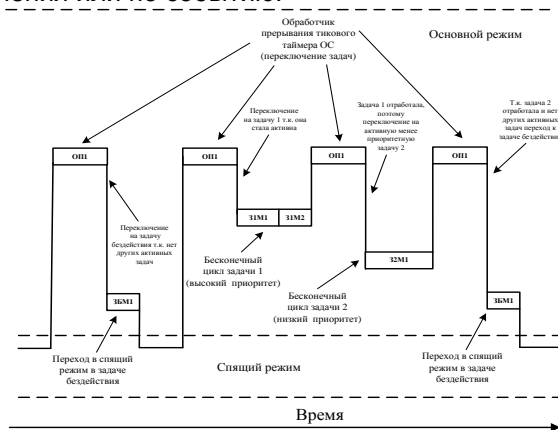


Рисунок 8 - Низкопотребляющий вариант ОСРВ с вытесняющей с многозадачностью

Выводы

При помощи ОСРВ с вытесняющей многозадачностью можно создать встраиваемое программное обеспечение для медицинского оборудования, которое будет обладать достаточной надежностью, эффективно использовать все ресурсы микроконтроллера, параллельно выполнять все необходимые задачи и реагировать на события с точными требуемыми интервалами времени. Использование ОСРВ обеспечивает легкое и удобное создание проектов средней и крупной величины, которые легко масштабировать, а также производить их отладку и трассировку. Недостатком применения ОСРВ является дополнительный расход ресурсов микро-

РАЗДЕЛ 5. ПРОГРАММНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И КОМПОНЕНТЫ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

контроллера (процессорное время, RAM, Flash, таймер). Поэтому использование ОСРВ для простых проектов на слабых микроконтроллерных ядрах с малым количеством памяти не оправдано.

Результаты исследований, изложенные в данной статье, получены при финансовой поддержке Минобрнауки РФ в рамках реализации проекта "Создание высокотехнологичного производства по изготовлению мобильного многофункционального аппаратно-программного комплекса длительного кардиомониторирования и эргометрии" по постановлению правительства №218 от 09.04.2010 г. Исследования проводились во ФГАОУ ВПО ЮФУ.

СПИСОК ЛИТЕРАТУРЫ

1. Елисеев, Н. Микроконтроллеры EFM32 компании Energy Micro – ультранизкое потребление при высокой производительности [Электронный ресурс] / Н. Елисеев, М. Стариченков // Электроника НТБ. Выпуск №5/2011 – Режим доступа: <http://www.electronics.ru/journal/article/2922>. – Загл. с экрана. – Яз. рус.
2. Семейство микроконтроллеров MSP430. [Текст] / Рекомендации по применению: Пер. с англ. – М.: Серия «Библиотека Компэла», 2005. – 544 с.
3. Кириенко, В.В. Обзор ОСРВ, пригодных для встраиваемых систем, применяемых при разработке современного диагностического оборудования [Текст] / В.В. Кириенко // Ползуновский вестник. – 2013. - № 2 - С. 280 – 282.
4. Ванцев, Д.В. Основные модели программирования микроконтроллеров [Электронный ресурс] / Д.В. Ванцев. - Режим доступа: http://mcu.ucoz.ru/publ/programmirovanie/razrabotka_programmnogo_obespechenija/osn_ovyne_modeli_programmirovaniija_mikrokontrollerov/11-1-0-4 – Загл. с экрана. – Яз. рус.
5. Курниц, А. FreeRTOS. Операционная система реального времени для микроконтроллеров [Электронный ресурс] / А. Курниц // Компоненты и технологии, № 2/2011 - Режим доступа: <http://kit-e.ru/assets/listalka/Kurniz/Kurniz.pdf> – Загл. с экрана. – Яз. рус.
6. EmbOS [Электронный ресурс] / User & reference guide. Режим доступа: https://www.segger.com/admin/uploads/productDocs/UM01001_emb_OS_Generic.pdf. – Загл. с экрана. – Яз. англ.
7. Labrosse uC/OS-III The Real-Time Kernel [Электронный ресурс] / J Jean. - Режим доступа: <http://www.micrium.com/> – Загл. с экрана. – Яз. англ.

Начальник сектора Кириенко В.В тел.: (8634) 311-143, e-mail: snowball@nxt.ru, Научно-технический центр "Техноцентр" Южного федерального университета; начальник отдела, ассистент кафедры МПС Семенистая Е.С., тел. 8 (8634) 311-143, suncat_75@mail.ru - Научно-технический центр "Техноцентр" ФГАОУ ВПО Южный федеральный университет

УДК 615.471

ОБЗОР МЕТОДОВ ТЕСТИРОВАНИЯ ЭНЕРГОЭФФЕКТИВНОСТИ МИКРОКОНТРОЛЛЕРОВ С УЛЬТРАНИЗКИМ ПОТРЕБЛЕНИЕМ ДЛЯ МЕДИЦИНСКОГО ОБОРУДОВАНИЯ

В.В. Кириенко, Е.С. Синютин

В данной статье рассмотрены основные методы тестирования энергоэффективности микроконтроллеров, которые используются для создания современной медицинской техники с автономным питанием. Произведено тестирование энергопотребления представителей трех семейств микроконтроллеров.

Ключевые слова: тестирование энергопотребления, микроконтроллер, медицинское оборудование, автономное питание.

Актуальность

Современные медицинские устройства с автономным питанием требуют минимизации потребляемой энергии, т.к. это увеличивает время диагностирования и срок службы элементов питания, а также повышает удобство использования прибора и положительно сказывается на экологии. Понижения потребляемой энергии медицинским прибором такого

класса можно добиться применением низкопотребляющих микросхем [1]. Отсюда следует и необходимость применения низкопотребляющего микроконтроллера, который, как правило, является логическим ядром прибора и основным энергозатратным его элементом. Из существующих семейств микроконтроллеров в сегменте низкого энергопотребления были выбраны три основных ли-