

# ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ И РЕШЕНИЙ В ОБЛАСТИ ПРОЕКТИРОВАНИЯ МНОГОПОТОЧНОЙ АРХИТЕКТУРЫ МИКРОКОНТРОЛЛЕРНЫХ СИСТЕМ

Корниенко В.Р.

Научный руководитель – к.т.н., доц. Филиппенко И.В.

Харьковский национальный университет радиоэлектроники  
(61166, Харьков, пр. Науки, 14, каф. Автоматизации и проектирования  
вычислительной техники, тел (057)70-21-326)  
e-mail:valentyn.korniienko1@nure.ua

One of the microcontroller`s world problem now – problem of choice the concept of middleware software with simple bare-metal interaction. Modern embedded systems usually based on ARM microchips, that provide all possibilities for multithreaded architecture of application. This article is brief overview of concepts and modern techniques for microsystem designs.

Тенденция увеличения производительности микроконтроллерной системы за счет усовершенствования архитектуры и периферийных возможностей существенно влияет на подходы к архитектуре встраиваемого программного обеспечения. На текущий момент традиционные известные подходы не являются подходящими для организации архитектуры достаточно сложной системы. Известными на сегодня типовыми решениями являются: State-based программирование - типичная модель конечного автомата, примененная в рамках системы управления устройством. Данный подход к архитектуре позволяет проектировать решение в кратчайшие сроки, вводя в проектируемое приложение проблему связности архитектурных компонентов. Тесная связность компонентов ограничивает возможности тестирования приложения и возможности создания кроссплатформенного решения. Также в многопоточной среде модель состояния вводит в систему проблемы, связанные с обеспечением уникального доступа к периферийным ресурсам и данным, что ведет за собой ограничения, как по максимальной производительности решения, так и по надежности системы в целом, т.к. реализация уникального доступа к ресурсам является нетривиальной задачей в встраиваемых решениях. Следующим подходом, позволяющим решить архитектурные проблемы в приложении на МК, является построение архитектуры на базе операционной системы реального времени (ОСРВ), что позволяет добавить в приложение архитектурную модель диспетчера задач - множества задач. Данная модель позволяет проектировать систему как множество отдельных задач, не имеющих зависимостей между собой.

ОСРВ для встраиваемых систем в большинстве своем поддерживают режим вытесняющей многозадачности, предоставляют гибкий менеджмент памяти и ресурсов внутри системы. Некоторые ОСРВ поддерживают

тесное взаимодействие с графическими библиотеками для встраиваемых систем, одной из таких графических библиотек является свободно-распространяемая TouchGFX. На данный момент есть множество операционных систем, которые можно подразделить на три группы: универсальные, которые обеспечивают максимальную переносимость, специализированные, заточенные под конкретное семейство микроконтроллеров и сертифицированные, отвечающие определенным отраслевым требованиям и стандартам.

Наиболее применимой ОСРВ для подобных решений является FreeRTOS с дополнительным C++ API в виде CMSIS-OS. Недостатком данного подхода является привязка к конкретному API выбранной операционной системы. На данный момент набирает популярность построение систем на основе модели Акторов – концепции распределенных вычислений, позволяющей строить отказоустойчивые приложения с возможностью самовосстановления состояния Актора. Актор представляет собой примитивную единицу, получающую сообщения из диспетчера сообщений и выполняющую действия на основе полученных сообщений. Данная модель является синтезом модели многозадачной системы и системы на базе автоматных состояний. Каждый из акторов получает сообщения асинхронно, но обрабатывает по-очереди. Данная модель позволяет создавать систему из простых взаимодействующих компонентов с возможностью независимого тестирования каждого из акторов и обеспечивает масштабируемость, достаточную для переноса приложения на другую архитектуру или семейство процессоров.

Таким образом, из приведенных в статье подходов к проектированию архитектуры микроконтроллерного приложения необходимо отметить набирающую популярность модель акторов как основу приложения и проектирование архитектуры приложения на базе операционной системы реального времени. Данные подходы обеспечивают необходимую гибкость и масштабируемость приложения, достаточную для портирования и тестирования.

Список источников:

1. Сорокин С. Как много ОСРВ хороших. Современные технологии автоматизации. 1997. No 2.
2. Борисов-Смирнов А. Операционные системы реального времени для микроконтроллеров. Chip news. 2008. No 5.
3. Сорокин С. Системы реального времени. Современные технологии автоматизации. 1997. No 2.
4. Татарчевский В. Применение SWITCH- технологии при разработке прикладного программного обеспечения для микроконтроллеров. Компоненты и технологии. 2006. No 11.