

Операционная система реального времени FX-RTOS

Операционная система реального времени FX-RTOS Nanokernel предназначена для устройств на микроконтроллерах. Ядро системы отвечает требованиям детерминизма по времени реакции на события. FX-RTOS обладает механизмами защиты доверенной части от небезопасных приложений. Компонентная архитектура позволяет оптимизировать конфигурацию OSCPВ и исключить неиспользуемый код из сборки.

Вытесняющая многозадачность

Ядро FX-RTOS Nanokernel представляет собой программную библиотеку с полным набором сервисов OSCPВ, поддержкой до 1023 уровней приоритета, планированием MPQ с группировкой потоков одного приоритета по очередям, управляемым алгоритмом Round-robin. Традиционные объекты синхронизации (опционально доступны POSIX-совместимые реализации): мьютекс, семафор, очередь сообщений, пул блоков памяти, барьер, условная переменная.

Работа в реальном времени

Ядро ОС спроектировано таким образом, что позволяет гарантировать время реакции на прерывания. Предельное время задержки перед обработкой прерывания из потока максимального приоритета фиксировано независимо от количества потоков, таймеров и других системных объектов. Ядро полностью вытесняемо во всех конфигурациях FX-RTOS, что позволяет использовать его в критичных ко времени реакции приложениях. Например, позволяет добиться времени реакции менее 5 мкс на микроконтроллере ARM Cortex-M3 с частотой 100 МГц.

Безопасность и защита памяти

В конфигурации Secure ядро FX-RTOS реализовано в виде отдельного исполнительного модуля, который реализует API через интерфейс системных вызовов. Приложение разделяется на доверенную часть, выполняющуюся в режиме ядра, и ненадежную, работающую в непривилегированном режиме. Доступ к устройствам и системным ресурсам контролируется привилегированным модулем. Если обычный поток пытается использовать запрещенный ресурс или область памяти, привилегированное приложение способно предпринять защитные действия, например прекратить выполнение потоков ненадежного модуля. Режим безопасности позволяет разрабатывать защищенные от взлома узлы, подключенные к публичным компьютерным сетям. Например, коммуникационный стек и сетевой драйвер могут быть выделены в непривилегированный модуль. Даже если атакующий, эксплуатируя уязвимости прикладного ПО, запустит вредоносный код на устройстве, привилегированное приложение способно обнаружить несанкционированный доступ и вернуть контроль над устройством.

Гибкая конфигурация

Сложно найти единое решение, подходящее для всех типов задач в условиях жестких требований и наложенных ограничений. Конфигурация FX-RTOS масштабируется от минимальной OSCPВ, управляемой событиями, без полноценной поддержки потоков до системы, обеспечивающей поддержку многоядерных процессоров с SMP. Управляя набором компонентов, программист задает конфигурацию, после чего специальная утилита анализирует зависимости и, используя технику внедрения зависимостей, компилирует OSCPВ, игнорируя избыточный функционал. Выбор одной реализации из нескольких предложенных вариантов, например схемы синхронизации, позволяет оптимизировать производительность и время реакции на прерывания.

Ключевые возможности:

- Вытесняющая многозадачность
- Неограниченное количество потоков и системных объектов
- Широкий выбор примитивов синхронизации
- Время реакции на прерывание детерминировано
- Работа в условиях ограниченности ресурсов
- Наличие режима защиты и разграничения доступа
- Выделенный стек прерываний
- Настраиваемая система обработки прерываний
- Поддержка популярных инструментов разработки и типов процессоров
- Масштабируемость и поддержка многопроцессорности
- Возможность переключения между конфигурациями без изменения API
- Использование динамических прерываний системного таймера (Tickless)

Низкие требования к ресурсам

Оптимальный объем полученного машинного кода достигается за счет конфигурирования ядра на этапе сборки. Таким образом, пользователь получает только требуемое подмножество ОС. Минимальные требования – от 2 Кбайт ОЗУ и 8 Кбайт ПЗУ – могут меняться в зависимости от целевой платформы. Благодаря выделенному стеку прерываний нет необходимости резервировать стек для вложенных обработчиков во всех потоках. Базовые функции ОС позволяют использовать преимущества многозадачности: переносимость и сокращение времени разработки на микроконтроллерах начального уровня.

Масштабируемость

Все компоненты ядра FX-RTOS, кроме самых низкоуровневых, являются кроссплатформенными и написаны на языке Си. Список поддерживаемых архитектур включает ARM RISC-V, MIPS, AVR32, MSP430. ОС может быть перенесена на нестандартные процессоры (например, реализованные в ПЛИС) с разной шириной шины и порядком байтов. Ядро спроектировано с расчетом на симметричные многопроцессорные системы: исключены глобальные общесистемные блокировки, общее количество глобальных переменных между процессорами сведено к минимуму.

Многоядерность

FX-RTOS изначально была разработана с расчетом на симметричные многопроцессорные системы. Исключены глобальные общесистемные блокировки, общее количество глобальных переменных между процессорами сведено к минимуму. Следовательно, производительность масштабируется с ростом количества процессоров. Реализация AMP позволяет использовать общие примитивы синхронизации между ядрами, более эффективно используя преимущества параллелизма в гетерогенных системах на кристалле.

Экосистема

Пользователям предлагается набор программных сервисов, реализующих стандартные протоколы передачи данных и файловые системы: USB device, CAN/CanOpen, TCP/IP, журналируемая FAT, отказоустойчивый NAND FTL.

Open-Source

Версия Lite с открытыми исходными кодами под лицензией BSD и примерами использования доступна на Github.

Коммерческая лицензия

Клиенты получают доступ к полному набору модулей системы и инструментарию конфигурирования. В рамках поддержки производится портирование на целевую аппаратную платформу, предоставление консультаций и примеров проектирования.

Варианты конфигураций

Профиль	Tiny	Lite	Standard	Low latency	Secure
Доступен в Open Source	+		-		
Макс. количество потоков	16 или 32	Неограниченно			
Количество потоков с одинаковым приоритетом	1	Неограниченно			
Архитектура обработчика прерываний	Унифицированная			Сегментированная с отложенным вызовом процедуры обработчика	Унифицированная
Защита	-				Разделение режимов kernel/user
Обработка событий таймера реализована	В прерывании		Отдельный поток с настраиваемым приоритетом		
Целевые системы	Низкобюджетные 16- и 32-битные МК с частотой 20-30 МГц	32-битные МК общего назначения 70+ МГц		32-битные МК общего назначения 100+ МГц	
Минимальный объем ПЗУ	8 Кбайт (размер ядра < 2 Кбайт)	8 Кбайт (размер ядра ~ 2 Кбайт)		32 Кбайт	128 Кбайт
Минимальный объем ОЗУ	2 Кбайт	16 Кбайт			

Веб-сайт: www.eremex.ru

Тел./факс: +7 (495) 232-1864

E-mail: info@eremex.ru

GitHub: www.github.com/Eremex