

# Библиотека RealView Real-Time Library

Сергей КОПЫТИН  
ols@microcontroller.ru  
Кирилл ДОРОФЕЕВ  
info@microcontroller.ru

**RealView Real-Time Library — библиотека стандартных компонентов, включающая в себя в исходном коде полнофункциональную операционную систему реального времени RTX Kernel, коммуникационный модуль TCP/IP стек, файловую систему Flash File System, драйверы USB и CAN. В статье рассмотрены основные возможности и преимущества, которые предоставляет разработчику встраиваемых систем использование RealView Real-Time Library: обеспечение работоспособности системы, реализация сложных сетевых решений и совместимость различных компонентов системы.**

## Сложный мир встраиваемых систем

Во встраиваемых системах одновременно выполняется большое количество задач различных по характеру и быстродействию. При проектировании встраиваемых систем разработчик часто сталкивается со следующими проблемами:

- Задача должна быть выполнена на коротком интервале времени и через гарантированное время от нее должен быть получен ответ.
- Различные по времени и логике работы задачи подчинены друг другу и должны выполняться одновременно.
- Задачи должны реагировать на прерывания, отслеживать их приоритет и иметь возможность работать с многочисленными интерфейсами.

Решить перечисленные проблемы можно с помощью операционной системы реального времени Real-Time Kernel, при использовании которой разработчик получает следующие преимущества:

- разбиение программы на простые задачи;
- модульный принцип построения многозадачных приложений;
- сокращение затрат времени на разработку и тестирование приложения;
- простое конфигурирование для любого микроконтроллера;
- широкий набор функций для отслеживания событий и пересылки сообщений;
- быстрое переключение при прерываниях;
- минимальные затраты системных ресурсов.

## Библиотека RL-ARM — RealView Real-Time Library

Для ARM-микроконтроллеров компания Keil предлагает библиотеку стандартных компонентов RealView Real-Time Library (RL-ARM)

(рис. 1), необходимых разработчику при создании современных встраиваемых систем:

- **RTX Real Time Kernel** — полнофункциональная операционная система реального времени RTX Keil, которая осуществляет управление процессорным временем, семафорами (для распределения ресурсов), событиями (для синхронизации задач) и почтовыми ящиками (для связи между задачами). RTX легко настраивается при помощи файла конфигурации.
- **TCP/IP Networking Suite** — коммуникационный модуль на базе стека TCP/IP, специально адаптированный для встраиваемых приложений. Он включает в себя сокет TCP и UDP, интерфейсы PPP и SLIP, DNS, Telnet, TFTP, SMTP, встроенный веб-сервер и CGI с защитой паролем. Стек рабо-

тает как с Ethernet, так и с последовательным интерфейсом (модемом). Сконфигурированные примеры предназначены для работы на нескольких стандартных демонстрационных платах.

- **Flash File System** — система, которая позволяет сохранять файлы на Flash, ROM или RAM. Это полезно в решениях, где требуется большое количество памяти для хранения данных или для систем с интернет-интерфейсами HTTP или FTP.
- **USB Device Interface** — драйверы USB, которые позволяют подключать стандартные устройства, такие как HID или Mass Storage.
- **CAN Interface**: в RTX Keil входят библиотеки CAN, которые предоставляют мощный и простой интерфейс для различных контроллеров CAN.

RL-ARM и все его компоненты поставляются как royalty-free, то есть без дополнительных лицензионных отчислений на каждый создаваемый с их помощью проект. Каждая приобретаемая лицензия позволяет одному инженеру использовать RL-ARM для неограниченного числа проектов.

## Операционная система реального времени

Операционная система реального времени позволяет сократить время разработки, создать надежную, расширяемую систему. Поскольку операционная система берет на себя выполнение части функций, упрощается логическая структура приложения. Использование Real-Time Kernel позволяет разработчику сконцентрироваться на создании приложения, а не на управлении ресурсами системы, составляя расписание переключения задач и расставляя приоритеты. Распределением ресурсов занимается сама операционная система. Real-Time Kernel представляет собой разделительный слой между программным обеспечением и аппаратной частью, что поз-



Рис. 1. Структура библиотеки RealView Real-Time Library

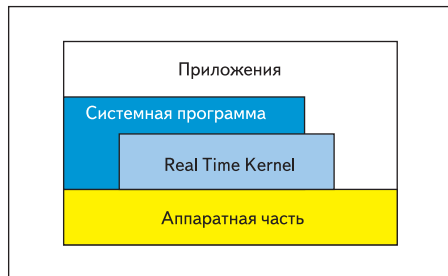


Рис. 2. Real-Time Kernel — разделительный слой между программным обеспечением и аппаратной частью

воляет осуществлять модернизацию аппаратуры без существенной переработки программы (рис. 2.).

Проблемы реального времени и многозадачности в Real-Time Kernel решаются с помощью механизмов синхронизации. Операционная система работает под управлением одного из таймеров микроконтроллера, который измеряет относительное время с момента старта. В качестве временных отсчетов служат прерывания от таймера. Интервал между прерываниями называется квантом времени. Механизм синхронизации задач базируется на трех методах синхронизации:

- Сигналы (*Signals*) — самый простой и быстрый способ синхронизации, когда одна задача для запуска на выполнение другой задачи посылает в нужный момент сигнал.
- Сообщения (*Messages*) — это данные, пересылаемые через почтовые ящики. Почтовый ящик хранит множество сообщений по принципу FIFO, то есть поступившее первым сообщением первым и выдается на обработку. Если несколько задач ожидает сообщения, его забирает задача с наивысшим приоритетом.
- Семафоры (*Semaphores*) — механизм для разделения общих ресурсов, таких как последовательный порт. При потребности в ресурсе задача выставляет требование к соответствующему семафору, и семафор его предоставляет, если нет других задач, нуждающихся в ресурсе.

Задача может ожидать наступления различных событий, не затрачивая при этом ресурсов микроконтроллера. К событиям относятся: получение сообщения, выставление сигнала, прерывание, разрешение от семафора, истечение кванта времени, любая комбинация этих событий. Пока одна задача ожидает появления события, другие задачи выполняются. Задача с наивысшим приоритетом поступает на выполнение вне очереди, прерывая текущую. Если инициация такой задачи не произошло, то по истечению кванта времени происходит переключение на другую задачу с тем же приоритетом.

Для хранения промежуточных результатов может потребоваться область динамической памяти. В многозадачных системах

время загрузки и выгрузки памяти не должно превосходить заданный лимит. По этой причине блоки переменного размера не подходят к приложениям реального времени. Real-Time Kernel работает с пулом памяти, выстроенным из фиксированных по размеру блоков. Можно создавать до 256 различных пулов памяти (каждый использует блоки различного, но фиксированного размера). Каждый пул состоит из 255 блоков.

### Что означает «реальное время»?

Иногда понятие реального времени напрямую связывают с высокой скоростью исполнения. Приложения, работающие в реальном времени, конечно, могут быть быстрыми, но работа в режиме реального времени лишь означает, что задание должно выполняться за какой-то определенный отрезок времени или в каком-либо заранее определенном порядке наряду с другими заданиями. Системы, работающие в реальном времени, могут быть относительно медленными. Хорошим примером может служить система выдачи багажа в аэропорту. Данная система не является быстрой, но должна обеспечить доставку багажа во вполне определенное место, что требует контроля в режиме реального времени, иначе багаж попадет не по адресу. Все задачи можно разделить на 2 класса:

- Быстрые задачи, у которых время ответа и время переключения при прерываниях значительно меньше, чем у стандартных задач. Быстрой задаче выделяется специальный банк регистров и специальный программный стек, которые не могут быть использованы другими задачами.
- Стандартные задачи используют меньше памяти, чем быстрые, но тратят больше времени при переключении задач. Стандартные задачи используют общий банк регистров и область стека. Когда задача прекращает свое выполнение, содержимое регистров и стека копируются во внешнее ОЗУ. При возвращении к обработке задачи, состояние ее регистров и область стека восстанавливаются. Прерывания обрабатываются как быстрыми, так и стандартными задачами.

### Операционная система Keil RTX Real-Time Kernel

Хорошая операционная система реального времени способна создавать сложные, быстрые, гибкие, но в то же время простые с точки зрения реализации и дальнейшей поддержки встраиваемые системы. Предсказуемость работы и возможность расширения — также необходимые условия для хорошей операционной системы. Real-Time Kernel должна обеспечивать выполнение задач в течение определенного периода времени, и это время должно сводиться к минимуму. Значительное число уровней прерывания

(более чем 128), большое количество задач и гибкость диспетчеризации — вот требования, которым должна отвечать операционная система реального времени для встраиваемых систем.

Keil RTX оптимизирована для 32-разрядной архитектуры ARM и полностью отвечает всем этим требованиям. RTX имеет следующие характеристики: малое время переключения <5 мкс, 255 уровней прерываний, до 256 выполняемых одновременно задач, неограниченное число почтовых ящиков и таймеров пользователя. Гибкость RTX достигается за счет поддержки следующих типов многозадачности:

- циклическая *Round-Robin* (все активные задачи, они не имеют приоритетов, между ними осуществляется последовательное переключение с выделением одинакового времени для решения каждой задачи);
- совместная *Cooperative* (несколько задач выполняются параллельно, остальные задачи не активны и ждут сигнала от активных задач для запуска);
- вытесняющая *Preemptive* (каждый процесс имеет свой уровень приоритета, в соответствии с которым ему выделяется процессорное время).

Структура операционной системы Keil RTX Real-Time Kernel представлена на рис. 3.

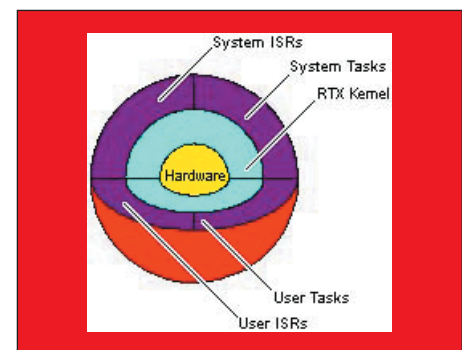


Рис. 3. Структура операционной системы Keil RTX Real-Time Kernel

Программный пакет RealView MDK-ARM включает в свой состав операционную систему RTX, но не содержит библиотеку RL-ARM, которая является самостоятельным программным пакетом. То есть MDK-ARM не содержит исходный код RTX Source Code, а также Flash File System, TCP/IP Protocol Suite и драйверы USB/CAN. Однако как RTX, так и RL-ARM полностью интегрированы в MDK-ARM. Функции RTX описаны на стандартном языке C и компилируются при помощи компилятора RealView. Описание на языке C позволяет легко определять функции задачи без конфигурирования стека и переменных.

Интегрированная среда разработки *μVision IDE* обеспечивает полную поддержку RTX при отладке и предоставляет информацию о задаче RTX, ее приоритете и состоянии

(выполнен, готов, отложен), осуществляет запись события вместе с меткой времени (активация задачи, вход или выход в/из программы прерывания). Эта информация дает понимание работы приложений и показывает очередность и продолжительность исполняемых задач, их переключений и прерываний.

Несмотря на то, что RTX является многофункциональной и мощной системой, ее системные требования невелики: по крайней мере, один таймер, минимум 500 байт ROM и менее 5 кбайт RAM. Полный исходный код RTX становится частью Real-Time Library, позволяя пользователям видеть весь набор функций, использованных при создании проекта, и обеспечивая работоспособность их приложения в будущем.

### Коммуникационные возможности библиотеки RL-ARM

Встраиваемые приложения взаимодействуют с другими системами или с периферийными устройствами, подключаемыми через последовательные интерфейсы USB или CAN. Библиотека RL-ARM содержит конфигурируемые драйверы USB и CAN, что существенно облегчает подключение большинства ARM-устройств с этими интерфейсами.

Не всегда удается объединить в сеть устройства с помощью последовательных интерфейсов. Решением задачи в таком случае является использование протокола TCP/IP, который позволяет подключить систему в сеть или к Интернету.

В то же время использование протокола TCP/IP может вызвать серьезные трудности у разработчика, который впервые столкнется с созданием сетевых приложений. Сразу

возникают вопросы: где достать подходящий TCP/IP стек, как заставить его работать, как сконфигурировать все сложные периферийные устройства, будет ли все это работать с другими компонентами системы.

К счастью, все эти вопросы решаются при использовании Real-Time Library. Библиотека TcpNet специально разработана для встраиваемых систем и характеризуется максимальным быстродействием при минимальных затратах памяти. Пользователь может выбрать любой из распространенных интерфейсов, например TCP/IP и UDP-сокеты, а также различные физические соединения: Ethernet, PPP и SLIP (dial-up).

Так же как и RTX, TcpNet полностью подержан RealView MDK. Мастер конфигурации позволяет настроить все сетевые параметры, оперативно подключить устройство к сети и проверить его работу. Полная отладочная информация доступна через последовательный интерфейс UART во время работы системы. Наглядность сетевых процессов предоставляет разработчику всю требуемую информацию для корректировки взаимодействия сетевого интерфейса с другими программными компонентами системы. Программы TcpNet написаны с использованием стандартных конструкций языка C и скомпилированы при помощи RealView компилятора MDK-ARM (так же как и для RTX).

### Примеры реальных приложений

Для пользователя наличие собственной библиотеки компонентов, способной помочь при создании сложной встраиваемой системы, библиотека RL-ARM — существенный шаг вперед. Теперь он может имплементировать RL-ARM в свою систему.

Главное преимущество RL-ARM состоит в том, что все компоненты полностью совместимы, каждый из них может работать по отдельности или вместе, как полнофункциональная система.

Использование RL-ARM существенно облегчает процесс разработки и внедрения новых компонентов, так как в библиотеке приведен целый ряд примеров для разработчиков. Даны примеры для управления в режиме реального времени, сетевого взаимодействия, веб-сервера и приложений для удаленного сбора данных. Это примеры реальных приложений, которые могут быть использованы в собственных разработках.

### Заключение

Библиотека Keil Real-Time предлагает реальные преимущества для разработчиков встраиваемых систем. Обеспечение работоспособности системы, сложные сетевые решения и совместимость различных компонентов системы теперь легки и интуитивно понятны.

Существует много других путей для достижения того же результата, однако библиотека RL-ARM предлагает решение, не требующее дополнительных лицензионных выплат при небольшой стоимости всего пакета. У такого решения немного альтернатив, которые давали бы разработчику такие возможности в выборе компонентов и были бы так просты во взаимодействии с другими приложениями, а также легко встраивались в их конечную систему. ■

### Литература

1. [www.keil.com](http://www.keil.com)
2. [www.arm.com](http://www.arm.com)