**Предисловие. FreeRTOS и Cortex-M3.**

**Многозадачность на микроконтроллере Cortex-M3.**

**Введение в многозадачность и небольшие встраиваемые системы.**

Микроконтроллеры (MCU), которые содержат ARM Cortex-M3 ядро доступны от многих производителей и идеально подходит для глубоко встраиваемых приложений реального времени. Обычно, приложения такого типа включают в себя смесь обоих типов требований реального времени - как жестких, так и мягких.

Мягкие требования реального времени - это те которые устанавливают срок времени, но не выполнение этих сроков не делают систему бесполезной. Например, медленный отклик на нажатия клавиш может сделать систему разражающе безразличной, фактически не делая ее не применимой.

Жесткие требования реального времени - это те, которые устанавливают срок времени и невыполнение в эти сроки приведет к результату абсолютного провала системы. Например, подушка безопасности водителя была бы бесполезной, если откликалась на датчик аварии слишком медленно.

FreeRTOS это ядро реального времени (или планировщик реального времени), поверх которого могут быть построены приложения микроконтроллеров Cortex-M3 для выполнения их требований жесткого реального времени. Это позволяет организовывать приложения Cortex-M3 как коллекцию независимых потоков выполнения. Так как большинство микроконтроллеров Cortex-M3 имеют только одно ядро, то реально только один поток может выполняться в каждый момент времени. Ядро решает какой поток должен выполняться на основе приоритетов, назначеных каждому потоку разработчиком приложения. В самом простейшем случае, разработчик приложения может назначить более высокие приоритеты потокам, которые реализуют жесткие требования реального времени и менее высокие потокам, которые реализуют мягкие требования реального времени. Это гарантировало бы, что потоки жесткого реального времени выполнялись бы вперед потоков мягкого реального времени, но распределение приоритетов не всегда настолько упрощено.

Не беспокойтесь, если не полностью понимаете понятия из предыдущего параграфа. Следующие главы предоставляют детальное объяснение, с большим количеством примеров, чтобы помочь вам понять как использовать ядро реального времени, и FreeRTOS в частонсти.

**Замечание о терминологии.**

Во FreeRTOS, каждый поток выполнения называется "задача". Внутри сообщества встраиваемых систем не существует соглашений о терминологии, но предпочтение "задачи" "потоку" может иметь более определенное значение в некоторых областях приложения.