3-14-11-rtos.ppt

CTI One Corporation

Version: x0.1
Date: March 21, 2019
Project Lead: Harry Li, Ph.D.

Group Leaders:

Team members:

RTOS Tutorial Course

http://www.FreeRTOS.org/Documentation

Reference: FreeRTOSConfig.h

http://www.SafeRTOS.com - A version certified for use in safety critical systems.

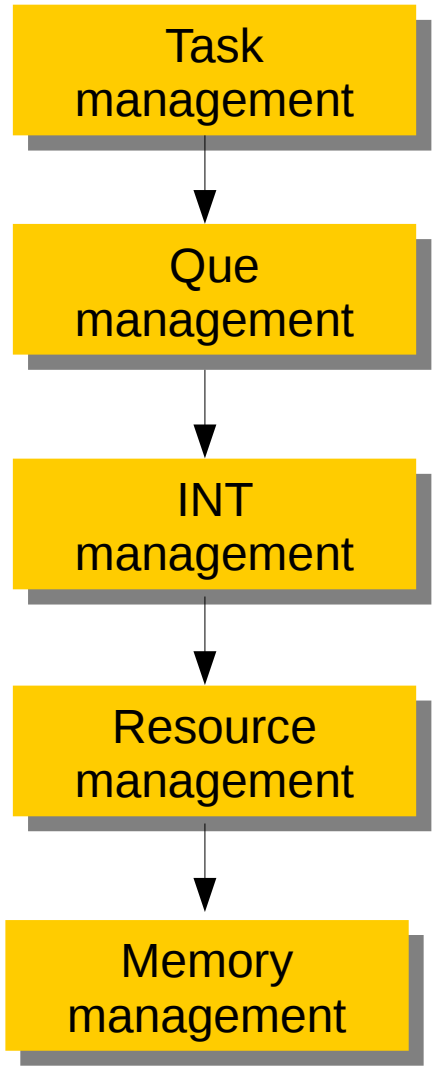
http://www.OpenRTOS.com - Commercial support, development, porting, licensing and training services.

Total 19 Exercise Homework

Having a problem? Start by reading the FAQ
http://www.FreeRTOS.org/FAQHelp.html

Mastering the FreeRTOS Real Time Kernel - a Hands On Tutorial Guide
FreeRTOS V10.0.0 Reference Manual
Book companion source code

RTOS Introduction On Tasks



Example 1. Creating tasks
Example 2. Using the task parameter
Example 3. Experimenting with priorities
Example 4. Using the Blocked state to create a delay

Example 5. Converting example tasks using vTaskDelayUntil()
Example 6. Combining blocking and non-blocking tasks
Example 7. Defining idle task hook
Example 8. Changing task priorities
Example 9. Deleting tasks

xTaskCreate() API
vTaskDelayUntil() API
vTaskPrioritySet() API
uxTaskPriorityGet() API
vTaskDelete() API

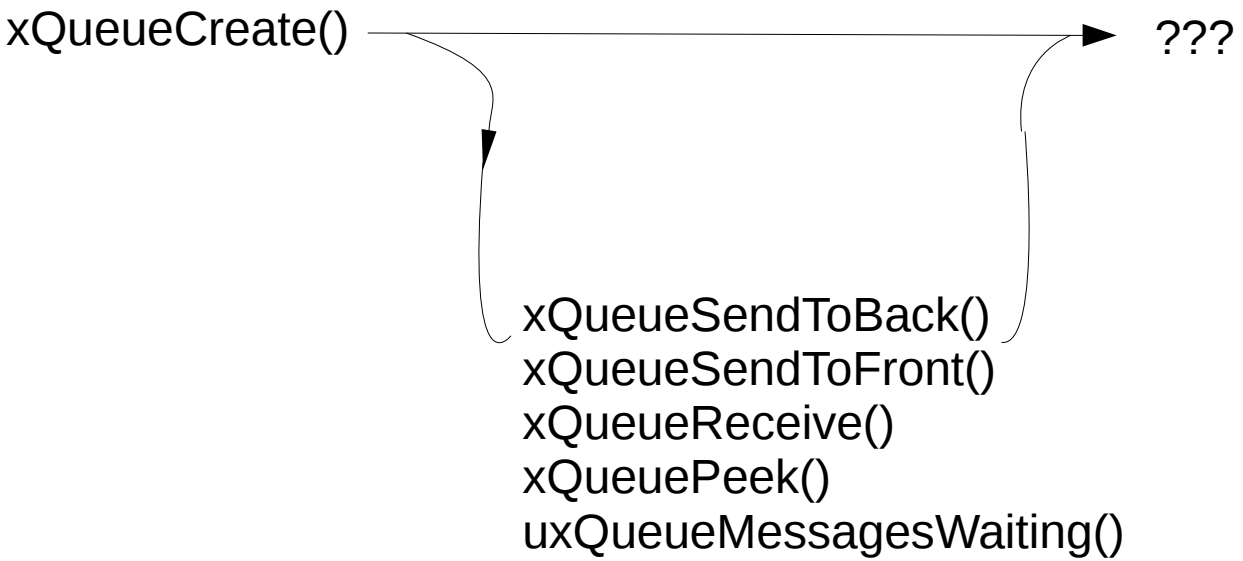
Que Management

What is a Queue and its Characteristics
Data Storage
Access by Multiple Tasks
Blocking on Queue Reads
Blocking on Queue Writes
Using a Queue
Using Queues to Transfer Compound Types
Working with Large Data

xQueueCreate()
xQueueSendToBack()
xQueueSendToFront()
xQueueReceive()
xQueuePeek()
uxQueueMessagesWaiting()

Example 10. Blocking when receiving from a queue

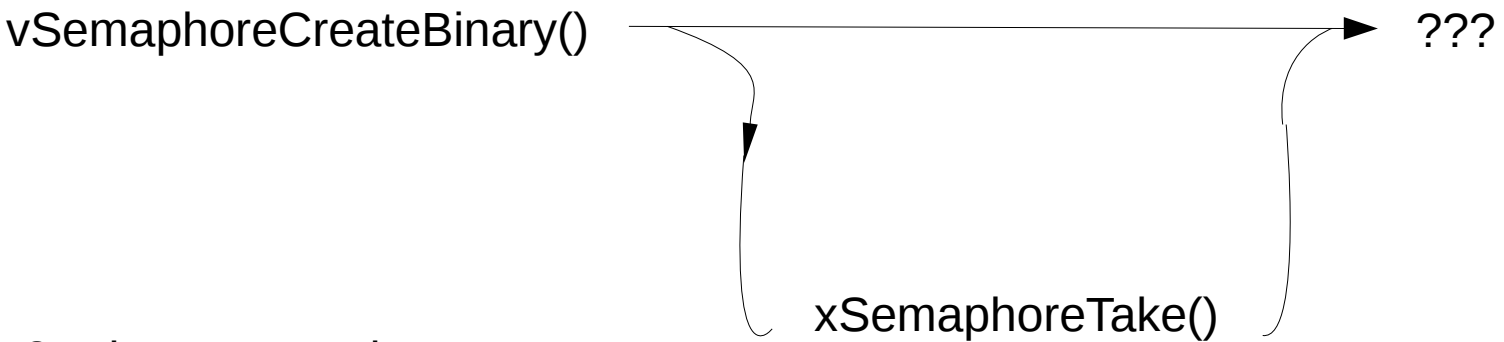
Example 11. Blocking when sending to a queue or sending structures on a queue



INT Management

Events and Scope
Deferred Interrupt Processing
Binary Semaphores for Synchronization
Writing Interrupt Handlers
Counting Semaphores
Queues within an Interrupt Service
Efficient Queue
Interrupt Nesting

vSemaphoreCreateBinary()
xSemaphoreTake()
xSemaphoreGiveFromISR()
xSemaphoreCreateCounting()
xQueueSendToFrontFromISR()
xQueueSendToBackFromISR()

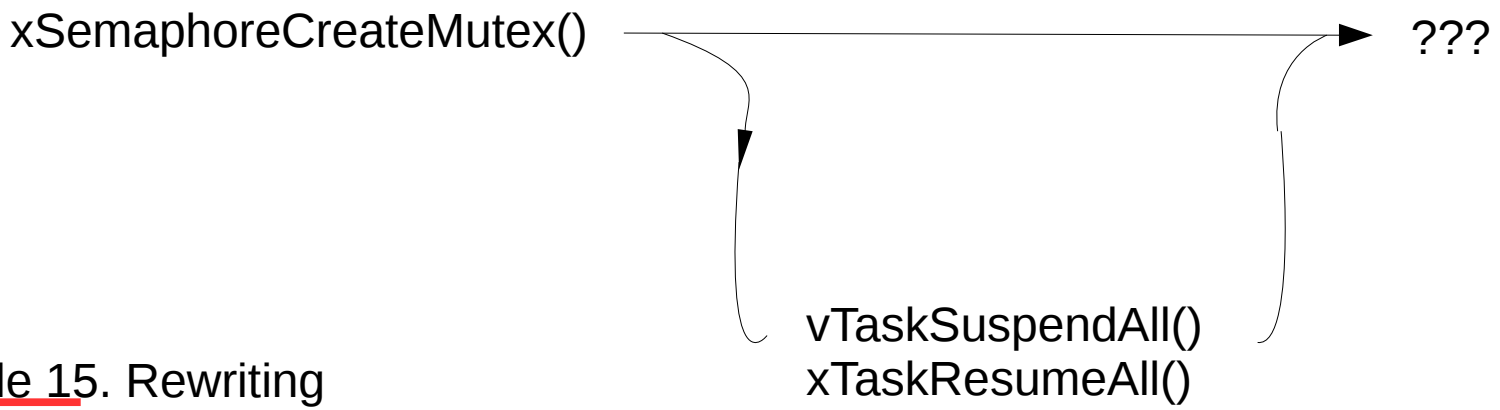


Example 12. Binary semaphore to synchronize a task with INT
Example 13. A counting semaphore to synchronize a task with an interrupt
Example 14. Sending and receiving a queue from within an interrupt

Resources Management

Mutual Exclusion and Scope
Critical Sections and Suspending the Scheduler
Basic Critical Sections
Suspending (or Locking) the Scheduler
Mutexes (and Binary Semaphores)
Priority Inversion
Priority Inheritance
Deadlock (or Deadly Embrace)
Gatekeeper Tasks

vTaskSuspendAll()
xTaskResumeAll()
xSemaphoreCreateMutex()



Example 15. Rewriting vPrintString() using a semaphore
Example 16. Re-writing vPrintString() using gatekeeper task

Definition:

Memory Management

Memory Allocation Schemes

xPortGetFreeHeapSize()

Example 17: Heap_1.c
Example 18: Heap_2.c
Example 19: Heap_3.c

Definition:

Introduction

(1) FreeRTOS is a real-time kernel (scheduler) on top of which LPC17xx applications can be built to meet their hard real-time requirements.

(2) It allows LPC17xx applications to be organized as a collection of independent threads;

(3) As the LPC17xx has only one core, in reality only a single thread can be executing at any one time. The kernel decides which thread should be executing by priority assigned to each thread by designer.

(4) The designer can assign higher priorities to threads that implement hard real-time requirements, and lower priorities to those soft real-time requirements.

The kernel is responsible for execution timing and provides a time-related API to the application. The benefits are:
(1) Maintainability/Extensibility
(2) Modularity
(3) Team development
(4) Easier testing
(5) Code reuse
(6) Idle task creation
(7) Flexible interrupt handling
(8) Easier control over peripherals
Gatekeeper tasks can be used to serialize access to peripherals.