

/*

freeRTOSConfig.h

FreeRTOS V7.0.1 - Copyright (C) 2011 Real Time Engineers Ltd.

```
*****
*
* FreeRTOS tutorial books are available in pdf and paperback.
* Complete, revised, and edited pdf reference manuals are also
* available.
*
* Purchasing FreeRTOS documentation will not only help you, by
* ensuring you get running as quickly as possible and with an
* in-depth knowledge of how to use FreeRTOS, it will also help
* the FreeRTOS project to continue with its mission of providing
* professional grade, cross platform, de facto standard solutions
* for microcontrollers - completely free of charge!
*
* >>> See http://www.FreeRTOS.org/Documentation for details. <<<
*
* Thank you for using FreeRTOS, and thank you for your support!
*
*****
```

This file is part of the FreeRTOS distribution.

FreeRTOS is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License (version 2) as published by the Free Software Foundation AND MODIFIED BY the FreeRTOS exception.

>>>NOTE<<< The modification to the GPL is included to allow you to distribute a combined work that includes FreeRTOS without being obliged to provide the source code for proprietary components outside of the FreeRTOS kernel. FreeRTOS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License and the FreeRTOS license exception along with FreeRTOS; if not it can be viewed here: <http://www.freertos.org/a00114.html> and also obtained by writing to Richard Barry, contact details for whom are available on the FreeRTOS WEB site.

1 tab == 4 spaces!

<http://www.FreeRTOS.org> - Documentation, latest information, license and contact details.

```

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

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

/* The following #error directive is to remind users that a batch file must be
executed prior to this project being built. The batch file *cannot* be
executed from within CCS4! Once it has been executed, re-open or refresh
* the CCS4 project and remove the #error line below.
*/
#error Ensure CreateProjectDirectoryStructure.bat has been executed before building. See comment immediately above.

#ifndef FREERTOS_CONFIG_H
#define FREERTOS_CONFIG_H

/*-----
* Application specific definitions.
*
* These definitions should be adjusted for your particular hardware and
* application requirements.
*
* THESE PARAMETERS ARE DESCRIBED WITHIN THE 'CONFIGURATION' SECTION OF THE
* FreeRTOS API DOCUMENTATION AVAILABLE ON THE FreeRTOS.org WEB SITE.
*
* See http://www.freertos.org/a00110.html.
*-----*/

#define configUSE_PREEMPTION 1 // This is set to 1 if the preemptive kernel is desired.
#define configUSE_IDLE_HOOK 1 // An idle task hook will execute a function during each cycle of the idle task.
#define configUSE_TICK_HOOK 0 // A tick hook function will execute on each RTOS tick interrupt if this value is set to 1.
// Again, this will be useful for "probing" the system.
#define configCPU_CLOCK_HZ ( 168000000UL ) // This is the max microcontroller clock frequency.
#define configTICK_RATE_HZ ( ( portTickType ) 1000 ) // This is the frequency at which the RTOS tick will operate. 1000 means
1KHz.
#define configMAX_PRIORITIES ( ( unsigned portBASE_TYPE ) 5 ) // The total number of priority levels that can be assigned when
prioritizing a task.
#define configMINIMAL_STACK_SIZE ( ( unsigned short ) 200 ) // This is the minimal size stack.
#define configTOTAL_HEAP_SIZE ( ( size_t ) ( 50 * 1024 ) ) // It is the total size of the array.
#define configMAX_TASK_NAME_LEN ( 10 ) // The maximum number of characters that can be used to name a task.
#define configUSE_TRACE_FACILITY 0
#define configUSE_16_BIT_TICKS 0 // This configuration item controls whether or not to use a 16-bit or 32-bit counter for recording
// elapsed time.
// A 32-bit value will perform better on the STM32 because the native counter size is 32-bits.

```

```

#define configIDLE_SHOULD_YIELD 1 // 0==NO 16bit Ticks
// This configuration item controls how processes that are running with idle priority react to a
// preemption request from a higher priority process.
// If it is set to 0, a process with idle priority is not preempted until the end of its
// time slice. If it is set to 1, a process with idle priority will yield immediately to the
// higher priority process. However, the higher priority process will only be given whatever time was
// left within the time slice originally assigned to the idle task (i.e., it will not have a whole
// time slice to compute within).
//
//
#define configQUEUE_MUTEXES 0
#define configQUEUE_REGISTRY_SIZE 0
#define configGENERATE_RUN_TIME_STATS 0
#define configCHECK_FOR_STACK_OVERFLOW 2
#define configUSE_RECURSIVE_MUTEXES 0
#define configUSE_MALLOC_FAILED_HOOK 0
#define configUSE_APPLICATION_TASK_TAG 0
#define configUSE_COUNTING_SEMAPHORES 1

/* Co-routine definitions. */
#define configUSE_CO_ROUTINES 0
#define configMAX_CO_ROUTINE_PRIORITIES ( 2 )

/* Software timer definitions. */
#define configUSE_TIMERS 1
#define configTIMER_TASK_PRIORITY ( 3 )
#define configTIMER_QUEUE_LENGTH 6
#define configTIMER_TASK_STACK_DEPTH ( configMINIMAL_STACK_SIZE )

/* Set the following definitions to 1 to include the API function, or zero
to exclude the API function. */
#define INCLUDE_vTaskPrioritySet 1
#define INCLUDE_uxTaskPriorityGet 1
#define INCLUDE_vTaskDelete 1
#define INCLUDE_vTaskCleanUpResources 1
#define INCLUDE_vTaskSuspend 1
#define INCLUDE_vTaskDelayUntil 1
#define INCLUDE_vTaskDelay 1

/* Use the system definition, if there is one */
#ifndef __NVIC_PRIO_BITS
#define configPRIO_BITS __NVIC_PRIO_BITS
#else
#define configPRIO_BITS 4 /* 15 priority levels */
#endif

```

```
#define configLIBRARY_LOWEST_INTERRUPT_PRIORITY    15
#define configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY 5

/* The lowest priority. */
#define configKERNEL_INTERRUPT_PRIORITY          ( configLIBRARY_LOWEST_INTERRUPT_PRIORITY << (8 - configPRIO_BITS) )
/* Priority 5, or 95 as only the top four bits are implemented. */
#define configMAX_SYSCALL_INTERRUPT_PRIORITY    ( configLIBRARY_MAX_SYSCALL_INTERRUPT_PRIORITY << (8 - configPRIO_BITS) )

#define configASSERT( x ) if( ( x ) == 0 ) { taskDISABLE_INTERRUPTS(); for( ;; ); }

#define vPortSVCHandler SVC_Handler
#define xPortPendSVHandler PendSV_Handler
#define xPortSysTickHandler SysTick_Handler

#endif /* FREERTOS_CONFIG_H */
```