edge-oasis-bridgeware-gd-v2-freertos项目开发指南

现行框架说明

本项目基于FreeRTOS Kernel V10.5.1版本，移植到了GD32F103CBRC芯片上。

主要文件说明

- core：包含了工程项目的业务文件
  - inc：包含了所有的头文件
  - src：包含了所有的源文件
- firmware：包含了项目的依赖文件
  - CMSIS：包含了armcm3的配置文件
  - FreeRTOS：包含了FreeRTOS的依赖文件
  - GD32F10x_standard_peripheral：包含了GD32F10x的头文件和源文件
  - 其他文件
- CLI:包含支持命令行交互文件
- others

代码结构说明

代码维护方法

内存管理

- 对总体堆栈的配置：在FreeRTOS_Src工程目录下的FreeRTOSConfig.h文件中配置内存管理相关参数。这里默认配置configMINIMAL_STACK_SIZE为128，configTOTAL_HEAP_SIZE为5*1024，即128字节的栈空间，5K的堆空间。

> freeRTOSConfig.h 文件详细说明见FreeRTOSConfig.h

```
#define configMINIMAL_STACK_SIZE        ( ( unsigned short ) 128 )
#define configTOTAL_HEAP_SIZE           ( ( size_t ) ( 5 * 1024 ) )
```

- 对CLI部分的配置：在FreeRTOS_Src工程目录下的FreeRTOSConfig.h文件中配置内存管理相关参数。这里默认配置configCOMMAND_INT_MAX_OUTPUT_SIZE为1000，configQUEUE_REGISTRY_SIZE为200，即1000字节的输出缓冲区，200个队列注册空间。

```
#define configCOMMAND_INT_MAX_OUTPUT_SIZE 1000
#define configQUEUE_REGISTRY_SIZE 200
```

- 对具体任务栈的配置：在Application工程目录下的main.c文件中的具体task里配置内存管理相关参数。例如下列任务用于打印目前系统状态。这里默认配置Task_system_state的栈空间为128字节。

```
    taskENTER_CRITICAL();
    /*add tasks here*/
    xTaskCreate((TaskFunction_t )Task_system_state,            /* task function */
                            (const char*    )"Task_system_state",          /* task name for
                            (uint16_t       )128,                   /* task stack size */
                            (void*          )NULL,                   /* optional task startu
                            (UBaseType_t    )2,                      /* initial priority */
                            NULL);               /* optional task handle to create */
  // ···other tasks
    taskEXIT_CRITICAL();
```

- 对CLI注册任务的内存配置，在UARTCommandConsole.c配置申请的cmd任务交流字节大小：

```
/* Dimensions the buffer into which input characters are placed. */
#define cmdMAX_INPUT_SIZE               512

/* Dimentions a buffer to be used by the UART driver, if the UART driver uses a
buffer at all. */
#define cmdQUEUE_LENGTH                 512
```

中断优先级管理

- 对总体中断优先级的配置：在FreeRTOS_Src工程目录下的FreeRTOSConfig.h文件中配置中断优先级相关参数。这里默认配置configMAX_SYSCALL_INTERRUPT_PRIORITY为191，configKERNEL_INTERRUPT_PRIORITY为255，即255个内核中断优先级，屏蔽了最高的64个中断优先级，191个系统调用中断优先级。configLIBRARY_KERNEL_INTERRUPT_PRIORITY为15，即15个内核中断优先级。

> configMAX_SYSCALL_INTERRUPT_PRIORITY 详细解读见
> configMAX_SYSCALL_INTERRUPT_PRIORITY

```
/* This is the value being used as per the ST library which permits 16
priority values, 0 to 15.  This must correspond to the
configKERNEL_INTERRUPT_PRIORITY setting.  Here 15 corresponds to the lowest
NVIC value of 255. */
#define configLIBRARY_KERNEL_INTERRUPT_PRIORITY 15


/* This is the raw value as per the Cortex-M3 NVIC.  Values can be 255
(lowest) to 0 (1?) (highest). */
#define configKERNEL_INTERRUPT_PRIORITY                 255
/* !!!! configMAX_SYSCALL_INTERRUPT_PRIORITY must not be set to zero !!!!
See http://www.FreeRTOS.org/RTOS-Cortex-M3-M4.html. */
#define configMAX_SYSCALL_INTERRUPT_PRIORITY    191 /* equivalent to 0xb0, or priority 11. ucMax
// #define configMAX_SYSCALL_INTERRUPT_PRIORITY         207 /* equivalent to 0xc0, or priority 
```

- 对具体任务中断优先级的配置：在Application工程目录下的main.c文件中的具体task里配置中断优先级相关参数。例如下列任务用于打印目前系统状态。这里默认配置Task_system_state的中断优先级为2。

```
taskENTER_CRITICAL();
/*add tasks here*/
xTaskCreate((TaskFunction_t )Task_system_state,            /* task function */
            (const char*    )"Task_system_state",          /* task name for kernel awareness
            (uint16_t       )128,                   /* task stack size */
            (void*          )NULL,                   /* optional task startup argument */
            (UBaseType_t    )2,                      /* initial priority */
            NULL);              /* optional task handle to create */
// ···other tasks
taskEXIT_CRITICAL();
```