FreeRTOS调度修改

基本思路：将调度算法改为EDF调度。

具体修改：

（1）首先是在FreeRTOS.h文件中添加了一个全局变量：

int configUSE\_EDF\_SCHEDULER = 1;

值为1表示使用EDF调度算法。

其余的修改都是对tasks.c进行。

（2）在结构体tskTaskControlBlock中添加如下代码：

#if ( configUSE\_EDF\_SCHEDULER == 1 )

TickType\_t xTaskPeriod;

#endif

（3）定义EDF调度的就绪任务链表：

#if ( configUSE\_EDF\_SCHEDULER == 1)

PRIVILEGED\_DATA static List\_t xReadyTasksListEDF;

#endif

（4）将任务放进合适的就绪队列中：

#if configUSE\_EDF\_SCHEDULER == 0

#define prvAddTaskToReadyList( pxTCB ) \

traceMOVED\_TASK\_TO\_READY\_STATE( pxTCB ); \

taskRECORD\_READY\_PRIORITY( ( pxTCB )->uxPriority ); \

vListInsertEnd( &( pxReadyTasksLists[ ( pxTCB )->uxPriority ] ), &( ( pxTCB )->xGenericListItem ) )

#else

#define prvAddTaskToReadyList( pxTCB ) \

vListInsert( &(xReadyTasksListEDF), &( ( pxTCB )->xGenericListItem ) );

#endif

（5）添加方法：

BaseType\_t xTaskPeriodicCreate( TaskFunction\_t pxTaskCode, const char \* const pcName, const uint16\_t usStackDepth, void \* const pvParameters, UBaseType\_t uxPriority, TaskHandle\_t \* const pxCreatedTask, StackType\_t \* const puxStackBuffer, const MemoryRegion\_t \* const xRegions , TickType\_t period )；

（6）修改方法void vTaskStartScheduler( void )，添加如下代码：

#if (configUSE\_EDF\_SCHEDULER == 1)

{

tickType initIDLEPeriod = 100;

xReturn = xTaskCreatePeriodic( prvIdleTask, "IDLE", tskIDLE\_STACK\_SIZE, (void \* ) NULL, ( tskIDLE\_PRIORITY | portPRIVILEGE\_BIT ), NULL,initIDLEPeriod );

}

#else