

FreeRTOS列表和列表项

1，列表和列表项的简介（熟悉）

列表简介：列表是 FreeRTOS 中的一个数据结构，概念上和链表有点类似，列表被用来跟踪 FreeRTOS中的任务。

列表项简介：列表项就是存放在列表中的项目

特点：

列表相当于链表，列表项相当于节点，FreeRTOS 中的列表是一个双向环形链表

列表的特点：列表项间的地址非连续的，是人为的连接到一起的。列表项的数目是由后期添加的个数决定的，随时可以改变

数组的特点：数组成员地址是连续的，数组在最初确定了成员数量后期无法改变

在OS中任务的数量是不确定的，并且任务状态是会发生改变的，所以非常适用列表(链表)这种数据结构

列表结构体

列表项结构体

迷你列表项结构体

2，列表相关API函数介绍（掌握）

列表初始化函数vListInitialise()

1、初始化时，列表中只有 xListEnd，因此 pxIndex 指向 xListEnd

2、xListEnd 的值初始化为最大值，用于列表项升序排序时，排在最后

3、初始化时，列表中只有 xListEnd，因此上一个和下一个列表项都为 xListEnd 本身

4、初始化时，列表中的列表项数量为 0（不包含 xListEnd）

列表项初始化函数vListInitialiseItem()

1、初始化时，列表项不属于任何一个列表，所以为空

列表项插入函数vListInsert()

1、获取新插入的列表项的值

2、判断新插入的列表项数值大小

如果数值等于末尾列表项的数值。就插入到末尾列表项前面

否则遍历列表中的列表项，找到插入的位置

3、将列表项插入前面所找到的位置

4、更新待插入列表项所在列表

5、更新列表中列表项的数量

末尾列表项插入函数vListInsertEnd()

1、获取列表 pxIndex 指向的列表项

2、将待插入的列表项插入到 pxIndex所指向的列表项前面

3、更新待插入列表项的所在列表

4、更新列表中列表项的数量

列表项移除函数uxListRemove()

1、获取所要移除的列表项的所在列表

2、从列表中移除列表项

3、如果 pxIndex 正指向待移除的列表项，将其指向带移除的上一个列表项

4、将待移除列表项的所在列表指针清空

5、列表的列表项数目减一

6、返回列表项移除后列表中列表项的数量

3，列表项的插入和删除实验（掌握）

1、实验目的：学会对FreeRTOS 列表和列表项的操作函数使用，并观察运行结果和理论分析是否一致

2、实验设计：将设计三个任务：start_task、task1、task2

start_task	用来创建其他的2个任务
task1	实现LED0每500ms闪烁一次，用来提示系统正在运行
task2	调用列表和列表项相关API函数，并且通过串口输出相应的信息，进行观察