线程:

1. GUI 与按键操作

优先级低

CPU占用率很高,涉及到FLASH操作,LCD显示, 尽量不要直接操作FLASH

优先级最低

1. GPRS模块

这些操作时间相对较长(ms级别),所以优先级不能太高,考虑到GUI 操作容易使

优先级次低

2.1 串口2中断

处理办法是,建立一个缓冲区,串口中断填充这个缓冲区( 串口中断优先级比OS更高)

2.2 GPRS任务

任务中允许直接操作串口2及缓冲区,但必须加锁,并且这个锁不能是OS级别的,必须是CPU级别的(因为串口中断优先级比OS更高)

2.3 任务间的交互

一般来说都应该通过事件/消息机制交互

互斥量/信号量锁定要传输的数据

1. RF内部读头

所有关键操作都是在中断中完成的

所以对速度要求不高, 但优先级过低,会影响读卡速度

任务中的代码CPU占用率很低,不操作其他硬件,不存在慢速操作

优先级较高,不会影响其他任务

一个特殊的问题需要考虑, 设备允许串口通讯任务模拟RF读写器, 引起串口通讯任务需要直接操作RF, <RF任务>和<串口通讯任务>必须互斥的使用RF部件,需要有一个机制来确保进入通讯模式后,RF任务会暂停工作

1. CAN任务

CAN中断后,会发送消息到CAN-TASK, 任务优先级所以不会干扰中断

消息缓冲区的大小为2K, 一条消息14B, 所以可以缓存146条消息, 只要处理速度不是特别慢就可以

任务中的代码CPU占用率很低,不操作其他硬件,不存在慢速操作

优先级次高,不会影响其他任务

1. 时间片消息

有计时要求,精度要求高,必须很快 ,

实际上这个任务对时序要求并不高,但考虑到为简化编程, 把时间片消息优先级设置到最高,这样任务内操作就不需要考虑加锁

任务中的代码CPU占用率很低,不操作其他硬件,不存在慢速操作

优先级最高

1. RACE任务

后台操作, 所有业务逻辑处理和通讯

业务逻辑处理和通讯实际上不会同时操作,所以可以看作两个完全不同的任务,相互之间是隔离的

时钟处理也只允许在这个任务中直接操作

1. GUI任务

目前,所有主要操作都在这个任务中执行

GUI的操作会影响内部操作的效率,譬如通讯速率会受到干扰, 足环返回的机制也会受影响,所以最好是只操作GUI,不操作其它部件

但存在一个最大的麻烦,就是FLASH, GUI 操作FLASH 是非常频繁的(因为要读字库),导致干扰后台进程运行, 每次操作FLASH都必须加锁,改动量非常大

所以FLASH操作需要特别注意:

所有FLASH相关的操作都调用drv\_File\_\*\*\*

所有FLASH相关的操作都必须调用fns\_File\_\*\*\*