



应用层
最上层的文件节点，每个文件节点可以当做一个SPI外设

Driver 核心层
SPI Device 通过名字与SPI Driver 匹配，有指针指向SPI Master，挂在SPI Bus总线上

SPI Driver 是针对不同SPI外设写的驱动程序，调用SPI Core 提供的函数实现读写功能，挂在SPI Bus总线上

SPI 控制层
SPI Master 有控制传输的函数，SPI Master 是控制层的驱动所创建的，控制传输的功能也在控制层实现

SPI Control Driver 主要的功能包括寄存器的配置、中断注册、传输控制的实现和创建SPI Master，当做普通driver 挂在platform Bus总线上

SPI Control Devcie 会传输一些信息给SPI Control Driver(中断号，寄存器基地址，频率)，SPI Control Devcie 可以看作是SPI的channel，当做普通device 挂在platform Bus总线上

SOC 控制
SPI Control 是芯片内部的SPI控制模块，可以支持多个channel，每个channel会通过控制层的驱动抽象出一个SPI Master，通过SPI Master 的函数指针来控制数据的传输

woker-woke
创建线程 kthread_worker_fn
判断wake_list 是否为空，为空则schedule
非空则执行work->func(work)
work->func 会调用spi 发送函数

发送数据过程
创建message、transfer
transfer 添加到message的list中
调用spi_sync
给wake_list 添加成员
wait_for_completion

