一个驱动需要初始化和退出, 在初始化内部将driver加入内核, 在退出时从内核中删除. driver结构体需要有名称, 匹配设备的名称, 常用的函数接口, probe(驱动添加的最终工作), remove(驱动移除), shutdown(休眠), suspend(挂起), resume(唤醒).

驱动完成后要在同目录下的Makefile中添加obj-$(驱动别名) +=驱动文件, 之后在Kconfig中加入 驱动别名的配置, 同时在kernel/msm-\*/arch/arm64/configs/vendor/\*\_defconfig, kernel/msm-\*/arch/arm64/configs/vendor/\*-perf\_defconfig,

加入CONFIG\_驱动别名=y

常用驱动框架

[字符驱动](char-pc8736x_gpio.txt) : 字符驱动需要在module\_init或probe实现设备号申请和分配, cdev的初始化和加入到内核.

[misc驱动 : misc框架只需要misc\_register和misc\_deregister即可实现设备注册.](misc-scif_main.txt)

[platform驱动 : platform框架加入了resource来映射设备的io地址, 使用platform框架需要platform\_device和platform\_driver的名称来匹配. 只需要platform\_device\_register和platform\_device\_unregister即可实现设备注册.](C:\\Users\\admin\\Desktop\\杜军考核文档\\platform-spear13xx_pcie_gadget.txt)

[i2c驱动 : i2c驱动使用i2c\_add\_driver和i2c\_del\_driver来添加或删除i2c驱动. 不需要进行设备注册,设备挂载client.dev中, 只需要获取i2c\_driver中probe函数的形参client, 并通过client即可进行读写操作.](i2c-nq-nci.txt)

内核和用户空间交互

[使用设备节点 : 字符驱动需要在module\_init或probe实现设备号申请和分配, cdev的初始化和加入到内核.](dev_node-uleds.txt)

[使用文件节点 : 字符驱动需要在module\_init或probe实现设备号申请和分配, cdev的初始化和加入到内核.](sys_node-pn553.txt)