

- 1, `module_init(ft5306_touch_init);`
- 2, `static int __init ft5306_touch_init(void)`
 - a) `i2c_add_driver(&ft5306_touch_driver);`
- 3, `static struct i2c_driver ft5306_touch_driver` (就是向系统注册一些程序, 在系统规定的时机执行)
 - a) `.probe = ft5306_touch_probe`, 注册检测程序和一些初始化信息, 检测是否准备好程序 (开机时执行)
 - b) `.remove = ft5306_touch_remove`, 基本不执行, 只在卸载
 - c) `.suspend = ft5306_touch_suspend` (`#ifndef CONFIG_HAS_EARLYSUSPEND`) 注册挂起程序 (休眠时执行)
 - d) `.resume = ft5306_touch_resume` (`#ifndef CONFIG_HAS_EARLYSUSPEND`) 注册唤醒程序 (唤醒时执行)
- 4, `ft5306_touch_probe`, 检测是否准备好
 - a) `ft5306_touch_phys_init`, 初始化
 - b) `input_register_device(input_dev);`, 注册设备
 - c) `request_irq` ()。申请中断号, 只是注册但不执行
 - d) `INIT_WORK(&touch_work.work, ft5306_touch_worker);`完成结构初始化工作, 为将工作提交到工作队列做准备, 即为顶半部准备好 `work_struct` 结构体。只是完成初始化, 但不执行
 - e) `register_early_suspend(&ft5306_touch_early_suspend_desc);`在这里注册唤醒及休眠管理函数 (`#ifndef CONFIG_HAS_EARLYSUSPEND`)
 - f) `ft5306_virtual_keys_init` (), 虚拟按键初始化程序, 这就是虚拟按键程序的入口
- 5, 如果不做任何操作, 则进入休眠状态。
 - a) `ft5306_touch_early_suspend` ()
 - i. `ft5306_touch_suspend(touch->client, state);`
- 6, 如果唤醒, 则进入唤醒初始化程序
 - a) `ft5306_touch_late_resume` () 唤醒
 - i. `ft5306_touch_resume` () 唤醒
 1. `ft5306_touch_phys_init(data);`初始化
- 7, 如果还不做任何操作, 则循环第 4 步
(唤醒和休眠的入口, 在 `ft5306_touch_probe` 函数中
`#ifndef CONFIG_HAS_EARLYSUSPEND`
`register_early_suspend(&ft5306_touch_early_suspend_desc);`
`#endif`
由 `ft5306_touch_early_suspend_desc` 进入唤醒以及初始化, `ft5306_touch_probe` 函数中也调用初始化函数。)
- 8, 如果触摸屏幕, 则产生中断, 进入中断顶半部 `ft5306_touch_irq_handler` (), 把工作提交到工作队列, 等在安全的时候再执行底半部
- 9, 执行底半部, `ft5306_touch_worker(struct work_struct *work)`
 - a) `ft5306_touch_read_points(points);` 获取触摸屏信息数据
 - b) `ft5306_touch_finger_report(data, id, x, y, z);`向应用层上报信息
- 10, `module_exit(ft5306_touch_exit);`退出时调用

- 11, ft5306_touch_exit
- a) i2c_del_driver(&ft5306_touch_driver);调用.remove = ft5306_touch_remove
 - b) ft5306_touch_remove ()
 - i.