#include #include #include //probe function static int pen\_probe(struct usb\_interface \*interface, const struct usb\_device\_id \*id) { printk(KERN\_INFO "[\*] Soliduscode Pen Drive (%04X:%04X) plugged\n", id->idVendor, id- >idProduct); return 0; } //disconnect static void pen\_disconnect(struct usb\_interface \*interface){ printk(KERN\_INFO "[\*] Soliduscode Pen Drive removed\n"); } //usb\_device\_id static struct usb\_device\_id pen\_table[] = { //( 0781:5567) { USB\_DEVICE( 0x0781, 0x5567) },{} /\* Terminating entry \*/ }; MODULE\_DEVICE\_TABLE (usb, pen\_table); // usb\_driver static struct usb\_driver pen\_driver = { .name = "pen\_driver", .id\_table = pen\_table, //usb\_device\_id .probe = pen\_probe, .disconnect = pen\_disconnect, }; static int pen\_init(void){ int ret = -1; printk(KERN\_INFO "[\*] Constructor of Driver"); printk(KERN\_INFO "\tRegistering Driver with Kernel"); ret = usb\_register(&pen\_driver); printk(KERN\_INFO "\tRegistration is complete"); return ret; } static void pen\_exit(void){ //deregister printk(KERN\_INFO "[\*] Destructor of driver"); usb\_deregister(&pen\_driver); printk(KERN\_INFO "\tunregistration complete!"); } module\_init(pen\_init); module\_exit(pen\_exit); MODULE\_LICENSE("GPL"); MODULE\_AUTHOR("Vitians"); MODULE\_DESCRIPTION("USB Pen Registration Driver");