

hardware_communication ::InterruptHandler
# InterruptManager * interrupt_manager
# uint8_t interruptNumber
+ virtual uintptr_t handle Interrupt(uintptr_t esp)=0
# InterruptHandler(uint8 _t interruptNumber, Interrupt Manager *interrupt_manager)
# ~InterruptHandler()

driver::Driver
+ Driver()
+ virtual ~Driver()
+ virtual void activate()=0
+ virtual void deactivate()=0
+ virtual int reset()=0

driver::MouseDriver
+ static int8_t __mouse_x_ + static int8_t __mouse_y_ + static uint16_t old _char_under_mouse_pointer - int8_t buffer - uint8_t buttons - hardware_communication ::Port8Bit commandPort - hardware_communication ::Port8Bit dataPort - MouseEventHandler * mouseEventHandler - uint8_t offset
+ MouseDriver(hardware _communication::InterruptManager *interrupt_manager, MouseEventHandler *mouseEventHandler) + ~MouseDriver() + void activate() override + void deactivate() override + uint32_t handleInterrupt (uint32_t esp) override + int reset() override + static uint16_t mouse _block_video_mem_value (uint16_t old_char_under _mouse_pointer, uint8_t mouse _pointer_color)

