1. What is the purpose of the NVIC peripheral?

The NVIC peripheral is short for “Nested Vectored Interrupt Controller”, which handles both interrupts and execution exceptions. It is also capable of handling sleep modes, allowing the microcontroller to operate at very low power states and “wake up” when necessary. It provides a non-maskable interrupt, zero-jitter interrupt options, and four interrupt priority levels.

1. What is the difference between interrupt tail-chaining and nesting?

Interrupt tail-chaining means that the processor will immediately handle the next interrupt after the first one without entering any other state. That is, right on the “tail” end of one ISR completing, the second ISR will be handled without restoring the previous processor state from before the first ISR.

Nesting, on the other hand, assigns priorities to each ISR. Higher-priority interrupts can pre-emptively execute and suspend a lower-priority ISR until it is completed, at which point the lower-priority ISR will resume.

1. In what file are the CMSIS libraries that control the NVIC?

The libraries are in the core\_cm0.h file.

1. What is the purpose of the EXTI peripheral?

The EXTI, or “Extended Interrupts and Events Controller”, is a peripheral that allows for *non-peripheral* devices to trigger interrupts. Most commonly, it is used to generate interrupts from the GPIO pins of the device.

1. What file has the defined names for interrupt numbers?

File en.DM00051352.pdf (the STM32F0 series Cortex-M0 programming manual) has the names for interrupt numbers in the table “IPSR bit definitions” (Table 6). Table 12, “Properties of the different exception types”, has additional detail including priority levels.

1. What file has the Vector table implementation?

The STM32F0 series Cortex-M0 programming manual contains the Vector table implementation on page

17.