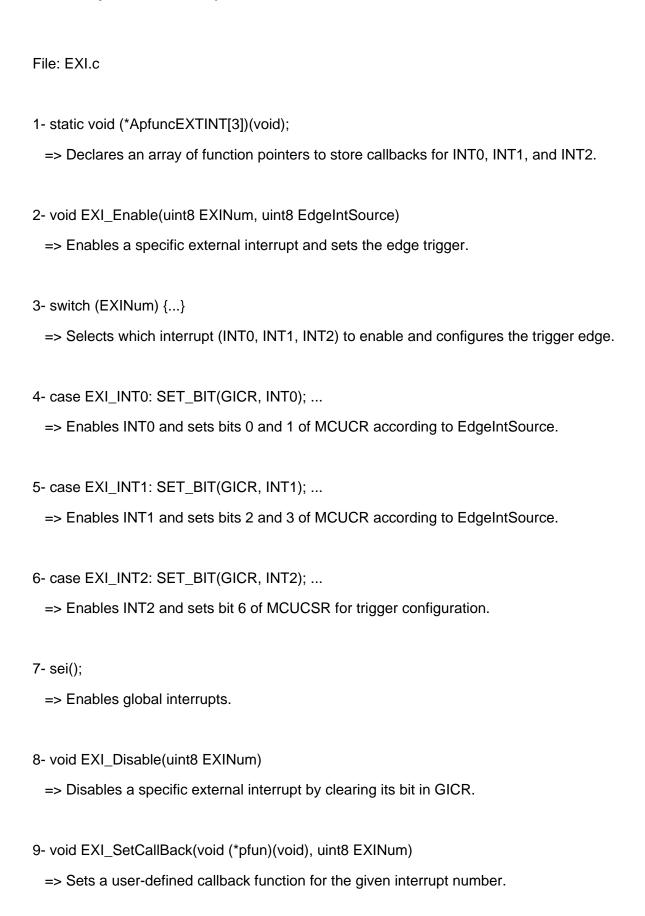
External Interrupt Driver for ATmega32 - Explanation

7. Line-by-Line Code Explanation



External Interrupt Driver for ATmega32 - Explanation

```
10- ISR(INT0_vect) { ApfuncEXTINT[EXI_INT0](); }
  => ISR for INT0, calls the registered callback.
11- ISR(INT1_vect) { ApfuncEXTINT[EXI_INT1](); }
  => ISR for INT1, calls the registered callback.
12- ISR(INT2_vect) { ApfuncEXTINT[EXI_INT2](); }
  => ISR for INT2, calls the registered callback.
File: EXI_interface.h
13- #define EXI_INT0 0 ... etc.
  => Macros defining symbolic names for INT0, INT1, INT2 and edge types.
14- Function declarations for enabling, disabling, and setting callbacks.
File: EXI_private.h
15- #define cli() and sei()
  => Inline assembly macros to disable/enable global interrupts.
16- #define INT0_vect ... etc.
  => Interrupt vector names for use with ISR macro.
17- #define ISR(vector) ...
  => Macro to simplify ISR declarations.
18- Bit definitions for GICR, MCUCR, and MCUCSR registers.
```