

Interrupts & DMA Function Prototypes

1. External Interrupt (Button → LED Toggle)

// Function Prototypes

```
void gpio_init(void);          // Configure LED + Button
void exti_init(void);          // Configure external interrupt line
void NVIC_config(void);        // Enable interrupt in NVIC
void EXTIx_IRQHandler(void);   // ISR for button interrupt
```

// Program Flow

```
int main(void) {
    gpio_init();
    exti_init();
    NVIC_config();
    while(1) {
        // Main loop idle (interrupt-driven)
    }
}
```

2. Timer Interrupt (Periodic LED Blink)

// Function Prototypes

```
void timer_init(void);         // Configure timer
void NVIC_config(void);        // Enable timer interrupt
void TIMx_IRQHandler(void);     // ISR for timer overflow
```

// Program Flow

```
int main(void) {
    gpio_init();
    timer_init();
    NVIC_config();
    while(1) {
```

```
        // Other work here (ISR handles LED toggle)
    }
}
```

3. UART Interrupt (Rx Echo Example)

// Function Prototypes

```
void uart_init(void);          // UART configuration
void NVIC_config(void);       // Enable UART interrupt
void USARTx_IRQHandler(void); // ISR for UART Rx/Tx
```

// Program Flow

```
int main(void) {
    uart_init();
    NVIC_config();
    while(1) {
        // Main loop free for other tasks
    }
}
```

4. ADC Interrupt (Conversion Complete)

// Function Prototypes

```
void adc_init(void);          // ADC config
void NVIC_config(void);       // Enable ADC interrupt
void ADCx_IRQHandler(void);   // ISR to read ADC result
```

// Program Flow

```
int main(void) {
    adc_init();
    NVIC_config();
    while(1) {
```

```
        // Process sensor data updated by ISR
    }
}
```

5. DMA with ADC (Auto Buffering)

// Function Prototypes

```
void adc_dma_init(void);    // Configure ADC + DMA
void dma_config(void);      // Set DMA source/dest/size
void NVIC_config(void);    // Enable DMA interrupt
void DMAx_IRQHandler(void); // ISR when DMA transfer complete
```

// Program Flow

```
int main(void) {
    adc_dma_init();
    dma_config();
    NVIC_config();
    while(1) {
        // Buffer filled by DMA, process when flag set
    }
}
```

6. Low Power Mode + Interrupt Wakeup

// Function Prototypes

```
void gpio_init(void);      // Button + LED
void exti_init(void);      // Button interrupt config
void enter_sleep_mode(void); // MCU sleep mode
void EXTIx_IRQHandler(void); // Wake-up ISR
```

// Program Flow

```

int main(void) {
    gpio_init();
    exti_init();
    while(1) {
        enter_sleep_mode();    // CPU sleeps
        // ISR wakes CPU, toggles LED
    }
}

```

Quick Reference — ISR Writing Rules

- Keep ISRs **short and fast**.
- Do not use heavy functions (printf, malloc, long loops).
- Use **flags or buffers** → process data in main().
- Example:
- volatile uint8_t flag = 0;
-
- void USARTx_IRQHandler(void) {
- if (RxFlagSet) {
- flag = 1; // set flag
- }
- }
-
- int main(void) {
- while(1) {
- if (flag) {
- flag = 0;
- // process received data
- }
- }}