Interrupts, DMA & Efficiency

D.1 Interrupt Basics

- **Interrupt:** A signal to CPU to stop current task & execute a handler (ISR).
- **ISR (Interrupt Service Routine):** A small, fast function to handle the event.
- **Polling vs Interrupts:**
    - Polling → CPU keeps checking flag (wastes cycles).
    - Interrupt → CPU free until event occurs.
- **NVIC (Nested Vectored Interrupt Controller):**
    - Present in ARM Cortex-M (STM32, ESP32).
    - Handles priority, nesting (one ISR preempting another).
- **Types:**
    - **External interrupts** → Button press, sensor trigger.
    - **Peripheral interrupts** → UART Rx, Timer overflow, ADC conversion done.

Practical: Configure button interrupt → toggle LED.

D.2 Advanced Interrupt Usage

- **Timer Interrupts:** Periodic task execution (e.g., blink every 1s).
- **UART Interrupts:** Trigger when Rx buffer full or byte received → efficient comms.
- **ADC Interrupts:** Trigger when conversion complete → process data immediately.
- **Best Practices:**
    - Keep ISRs short & fast.
    - Avoid printf in ISR.
    - Use flags/queues to communicate with main loop.

Practical:

- Timer interrupt → blink LED.
- UART Rx ISR → echo received data.
- ADC ISR → read sensor value & store.

D.3 DMA (Direct Memory Access)

- **Why DMA?** CPU doesn't need to move data manually.
- Transfers directly from **Peripheral → Memory** or **Memory → Peripheral**.
- **Use Cases:**
    - ADC continuously storing samples in buffer.
    - UART sending large string.
    - SPI transferring data to display.
- **DMA Flow:**
    - Configure source, destination, size.
    - Enable DMA channel.
    - Interrupt on transfer complete.

Practical:

- ADC + DMA → fill buffer of 100 samples.
- UART + DMA → send "Hello World" repeatedly without CPU load.

D.4 Efficiency & Low-Power

- **Busy-waiting = Bad:** CPU stuck in while loop wastes energy.
- **Interrupts = Better:** CPU works only when needed.
- **DMA = Best:** CPU can sleep while data moves automatically.
- **Low Power Modes:**
    - **Sleep:** CPU off, peripherals running.
    - **Stop/Standby:** deeper modes, wakeup by interrupt.

Practical: MCU in sleep mode, wakes on button interrupt.

Summary

- Polling → Interrupt → DMA is the **efficiency ladder**.
- ISRs must be short; heavy work done in main loop.
- DMA frees CPU → critical for IoT low-power devices.

References

- Interrupts in ARM Cortex-M
- STM32 External Interrupt Tutorial
- DMA Basics