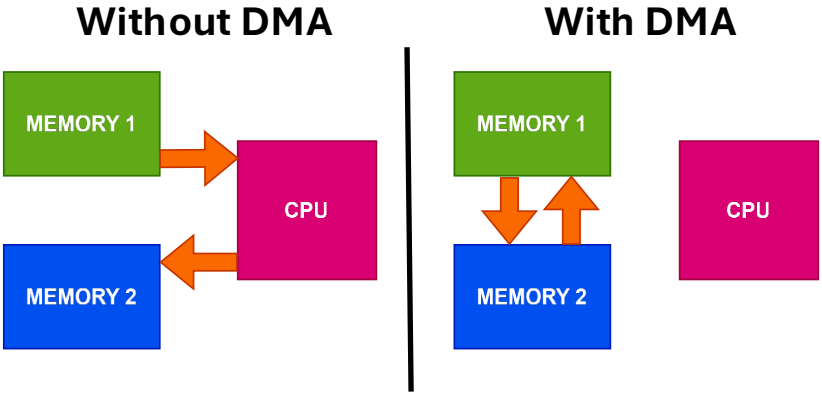
What Is DMA?

And When Should You Use It in Your Embedded System?



Fahad Bhatti
Founder Oxeltech (Embedded Development Service)

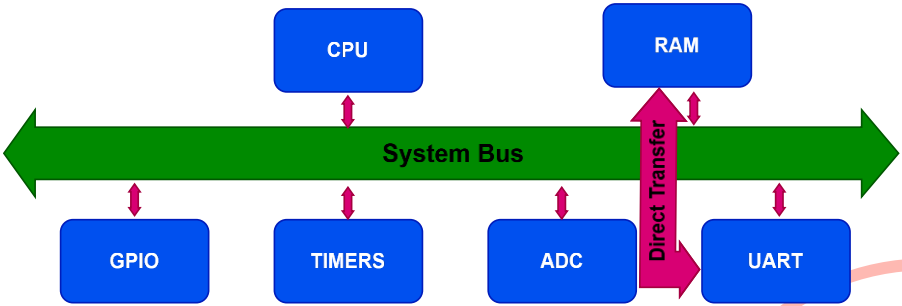
Swipe >

oxeltech

What Is DMA ?

Direct Memory Access (DMA) is a technique that moves data between **peripherals** (e.g. UART) and system **memory** (e.g. RAM) directly, **without involving the CPU**.

This frees CPU to perform other tasks while data transfers occur in parallel.



How DMA Works

1. Start Transfer: Device signals DMA to start a transfer.

2. Choose Device: DMA controller decides which device (ADC, UART etc.) can use the bus first.

3. Take Control: DMA takes control of system bus.

4. Move Data: Data moves between device and memory.

5. Notify CPU: DMA notifies CPU upon completion.

6. Process Data: CPU can now use the transferred data if needed.

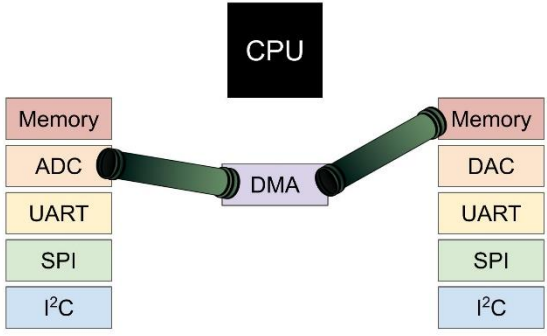
When to Use DMA

- You want to **move data quickly**
- You want to **offload** repetitive data transfer tasks from **CPU**
- You don’t want the CPU to be **too busy**
- You need your system to work in **real time**
- You're working with **large data** (>1 KB per burst), like from camera to memory
- Peripherals generate **frequent data** streams

Caution

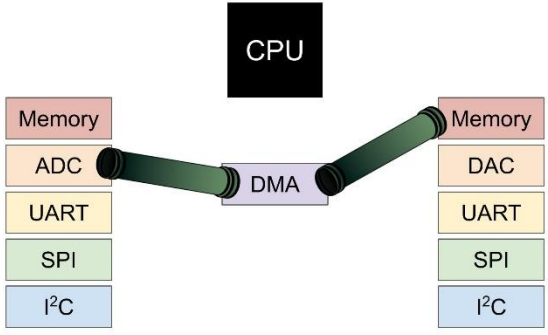
- For small data transfers, the time to set up DMA may outweigh its speed advantage.

- Too many devices using the bus can slow down data transfers



Simple Examples

- **UART with DMA**: Stream continuous sensor data without CPU blocking.
- **ADC with DMA**: Read temperature or voltage in the background
- **Camera with DMA**: Capture frames directly into memory.
- **Audio with DMA**: Stream audio smoothly without gaps.



oxeltech

Reach Out for Embedded, IoT, and Hardware Development Services

www.oxeltech.de

Fahad Bhatti
Founder Oxeltech
(Embedded Development Service)