Zephyr DMA UART TX Demo — STM32F407

🎯 Objective

Demonstrate **DMA-based Memory → Peripheral** transfer using **USART2 (PA2 TX)** on **STM32F407** under **Zephyr RTOS**.
The DMA controller streams a buffer from memory directly into the USART2 data register — the CPU doesn't touch each byte.

🧩 Project Structure

```
zephyr_dma_uart_demo/
├── CMakeLists.txt
├── prj.conf
├── boards/
│   └── arm/
│       └── stm32f4_disco.overlay
└── src/
    └── main.c
```

1️⃣ prj.conf

```
# Enable Zephyr kernel essentials
CONFIG_MAIN_STACK_SIZE=1024
CONFIG_HEAP_MEM_POOL_SIZE=1024
```

```
# UART driver
CONFIG_SERIAL=y

# Use async UART API (enables DMA)
CONFIG_UART_ASYNC_API=y

# Disable console over UART2 to free it for our DMA demo
CONFIG_CONSOLE=n
CONFIG_UART_CONSOLE=n
CONFIG_SERIAL_CONSOLE=n

# Logging (optional)
CONFIG_LOG=y
CONFIG_LOG_MODE_DEFERRED=y
CONFIG_LOG_DEFAULT_LEVEL=3
```

2️⃣ Devicetree Overlay — stm32f4_disco.overlay

```
/* Enable USART2 with DMA TX/RX for STM32F407
 * TX = PA2, RX = PA3
 * DMA1 Stream6 (TX), Stream5 (RX)
 */

&usart2 {
```

```
    status = "okay";
    current-speed = <115200>;
    hw-flow-control = "none";

    /* DMA configuration: memory → peripheral for TX */
    dmas = <&dma1 6 4 0 1 0x0>,   /* TX: Stream6, Channel4 */
           <&dma1 5 4 0 0 0x0>;   /* RX: Stream5, Channel4 */
    dma-names = "tx", "rx";

    pinctrl-0 = <&usart2_tx_pa2 &usart2_rx_pa3>;
    pinctrl-names = "default";
};

/* Ensure DMA1 is active */
&dma1 {
    status = "okay";
};

/* Define pinmux entries if not already present */
&pinctrl {
    usart2_tx_pa2: usart2_tx_pa2 {
        pinmux = <STM32_PINMUX('A', 2, STM32_AF7_USART2)>;
        bias-pull-up;
        drive-push-pull;
```

```
        slew-rate = <3>;
    };

    usart2_rx_pa3: usart2_rx_pa3 {
        pinmux = <STM32_PINMUX('A', 3, STM32_AF7_USART2)>;
        bias-pull-up;
    };
};
```

3️⃣ Application Code — src/main.c

```
/*
 * Zephyr DMA UART TX Demo on STM32F407
 * ------------------------------------
 * Sends a buffer via uart_tx(), which internally uses DMA1 Stream6
→ USART2->DR.
 *
 * Connections:
 *   PA2 (USART2_TX) → USB-UART adapter RX
 *   GND             → USB-UART GND
 *
 * Open terminal at 115200 8N1
 * Expected Output: "Hello from DMA via USART2 under Zephyr!"
 */
```

```
#include <zephyr/kernel.h>
#include <zephyr/device.h>
#include <zephyr/drivers/uart.h>

static const uint8_t dma_msg[] =
    "Hello from DMA via USART2 under Zephyr!\r\n";

/* Async event callback */
static void uart_cb(const struct device *dev,
                    struct uart_event *evt,
                    void *user_data)
{
    ARG_UNUSED(dev);
    ARG_UNUSED(user_data);

    switch (evt->type) {
    case UART_TX_DONE:
        printk("UART DMA TX DONE (%u bytes)\n", evt->data.tx.len);
        break;
    case UART_TX_ABORTED:
        printk("UART DMA TX ABORTED at %u bytes\n",
evt->data.tx.len);
        break;
    default:
```

```
        break;
    }
}

void main(void)
{
    const struct device *uart_dev;

    uart_dev = DEVICE_DT_GET(DT_NODELABEL(usart2));
    if (!device_is_ready(uart_dev)) {
        printk("USART2 device not ready!\n");
        return;
    }

    int ret = uart_callback_set(uart_dev, uart_cb, NULL);
    if (ret) {
        printk("uart_callback_set failed: %d\n", ret);
        return;
    }

    printk("Starting DMA TX...\n");
    ret = uart_tx(uart_dev, dma_msg, sizeof(dma_msg) - 1,
SYS_FOREVER_MS);
    if (ret) {
        printk("uart_tx failed: %d\n", ret);
```

```
        return;
    }

    printk("DMA TX started, waiting for completion...\n");

    while (1) {
        k_sleep(K_MSEC(500));
    }
}
```

⚙️ Build and Flash

🧱 Build

```
west build -b stm32f4_disco .
```

🔥 Flash

```
west flash
```

🧭 Monitor Output

Connect UART adapter:

- **PA2 (TX)** → adapter **RX**
- **GND** → adapter **GND**

Then run:

```
minicom -D /dev/ttyUSB0 -b 115200
```

Expected output:

```
Hello from DMA via USART2 under Zephyr!

UART DMA TX DONE (38 bytes)
```

🧠 Notes and Verification

✅ `CONFIG_UART_ASYNC_API` ensures Zephyr uses **DMA** for TX/RX if available.
✅ `uart_tx()` in async mode returns immediately — non-blocking DMA transfer.
✅ The STM32 UART driver automatically maps:

- USART2_TX → DMA1 Stream6 Channel4
- USART2_RX → DMA1 Stream5 Channel4

🧰 Troubleshooting

Issue	Possible Cause	Fix
`USART2 device not ready!`	Overlay not applied / wrong board name	Ensure overlay path matches `boards/arm/stm32f4_disco.overlay`
Garbage on serial	Baud mismatch	Check `current-speed` and terminal baud rate
TX done never prints	Missing DMA mapping	Confirm `dmas` and `dma-names` in overlay

Build fails: unknown pinctrl label	Add pinctrl section manually (see overlay section)

✅ Summary

Layer	Implementation	Description
Application	`uart_tx()`	Requests async TX
Zephyr driver	STM32 UART async driver	Uses DMA1 Stream6 for USART2 TX
Hardware	DMA1 Controller	Moves bytes from memory to USART2->DR
Peripheral	USART2	Shifts bytes out on PA2

Result:
A true DMA-based memory → peripheral transfer running under Zephyr RTOS on STM32F407.