

# 什么是DMA

DMA，就是direct memory access，直接内存访问。DMA技术的核心就是在交换数据时不需要CPU参与，模块可以自己完成。（详情可以见冯诺依曼结构中数据的传输方式）

## 为什么要用DMA？

串口发送/接收要频繁的折腾CPU造成CPU反复切换上下文导致系统效率低下，不如让他在内存区直接整完。

## 有什么优势？

传统的串口工作方式(无FIFO无DMA)效率是最低的，数据的接收/发送都在CPU里进行，适合没那么多数据要处理的低端单片机；高端单片机上CPU除了大量数据处理还有一大堆事要做会非常繁忙，所以都需要串口能够自己完成大量数据发送/接收，这时候就需要FIFO或者DMA模式。FIFO模式是一种轻量级的解决方案(只能从一定程度上解决问题，譬如几十字节的数据)，DMA模式适合大量数据迸发式的发送/接收(可以从根本上解决问题，可处理大量的数据)，具体来说可以实现在数据准备阶段，CPU与外设并行工作。

总之，会提高效率。

## 如何配置串口DMA接收

1. 使能DMA时钟，并等待数据流可配置。使能DMA时钟，才可配置DMA相关的寄存器。要对DMA的配置寄存器DMA\_SxCR进行配置，则要等其最低位为0----即DMA传输禁止，才可配置。
2. 配置串口的DMA通道

USART1 Configuration→DMA Settings →Add

3. 串口全局中断

USART1 Configuration→HVIC Settings →USART1 global interrupt ✓

```
/* USART1 init function */

void MX_USART1_UART_Init(void)
{
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 115200;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        _Error_Handler(__FILE__, __LINE__);
    }
}
```

```
//使能串口空闲中断
__HAL_UART_ENABLE_IT(&huart1, UART_IT_IDLE);
```

```

//DMA接收地址设置
HAL_UART_Receive_DMA(&huart1,uart1RxBuf,RX_SIZE);
//添加空闲中断处理函数
void UART_RxIdleCallback(UART_HandleTypeDef *huart)
{
    if(__HAL_UART_GET_FLAG(huart,UART_FLAG_IDLE)){
        __HAL_UART_CLEAR_IDLEFLAG(huart);

        HAL_UART_DMAStop(huart);

        hdma_usart2_rx.Instance->CNDTR = BUFFER_SIZE;
        rxSize = RX_SIZE-hdma_usart2_rx.Instance->CNDTR;
        printf("uart2 receive CNT: %d\n",rxSize);
        printf("data:%s\n",uart1RxBuf);
        memset(uart1RxBuf,0,sizeof(uart1RxBuf));
        HAL_UART_Receive_DMA(huart,uart1RxBuf,RX_SIZE);
    }
}

```

其中uart1RxBuf 为DMA 接收缓存数组

在 USART1\_IRQHandler(void) 中调用该函数。

## 各参数的意义

CHSEL[2:0] 3位8个通道选择，即对于流选择哪个通道

PL[1:0] 2位，设置流的软件优先级，DMA控制器的仲裁器根据软件优先级，辅以硬件优先级来仲裁进行哪个数据流哪个通道的数据传输。

MSIZE[1:0] 2位，存储器的数据宽度(8位、16位、32位)

PSIZE[1:0] 2位，外设的数据宽度

MINC 1位，设置存储器递增模式，存储器地址指针时递增还是固定

PINC 1位，设置外设递增模式，外设地址指针递增还是固定

DIR 2位，数据传输方向

TCIE 1位，传输完成中断使能

HTIE 1位，半传输中断使能

TEIE 1位，传输错误中断使能

DMEIE 1位，直接模式错误中断使能

EN 1位，0禁止该数据流，此时可配置该数据流

1使能该数据，参与仲裁器的竞争

