

UART

基本概念

DMA的概念

直接存储器访问 (DMA)：用于在外设与存储器之间以及存储器与存储器之间进行高速数据传输。DMA传输过程的初始化和启动由CPU完成，传输过程由DMA控制器来执行，无需CPU参与，从而节省CPU资源，提高利用率。

DMA数据传输的四个要素：

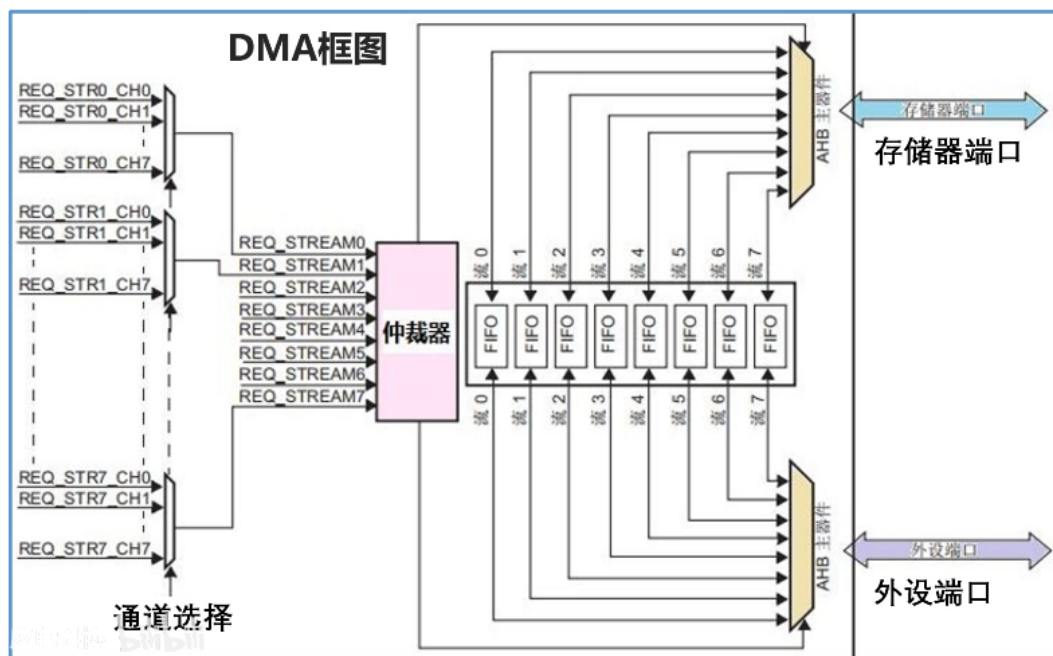
- ① 传输源：DMA数据传输的来源
- ② 传输目标：DMA数据传输的目的
- ③ 传输数量：DMA传输数据的数量
- ④ 触发信号：启动一次DMA数据传输的动作

UART

AM控制器

STM32的DMA控制器特点

1. STM32F411微控制器具备两个DMA控制器：DMA1和DMA2，每个控制器有8个数据流，每个数据流可以映射到8个通道（或请求）；
2. 每一个DMA控制器用于管理一个或多个外设的存储器访问请求，并通过总线仲裁器来协调各个DMA请求的优先级；
3. 数据流（stream）是用于连接传输源和传输目标的数据通路，每个数据流可以配置为不同的传输源和传输目标，这些传输源和传输目标称为通道（Channel）；
4. 具备16字节的FIFO。使能FIFO功能后，源数据先送入FIFO，达到FIFO的触发阈值后，再传送到目标地址。



DMA数据传输方式

UART

DMA

➤
数据传输方式

DMA数据传输方式

普通模式

传输结束后（即要传输数据的数量达到零），将不再产生DMA操作。若开始新的DMA传输，需在关闭DMA通道情况下，重新启动DMA传输。

循环模式

可用于处理环形缓冲区和连续数据流（例如ADC扫描模式）。当激活循环模式后，每轮传输结束时，要传输的数据数量将自动用设置的初始值进行加载，并继续响应DMA请求。

接口函数：

1.串口DMA方式发送函数

1 串口DMA方式发送函数：HAL_UART_Transmit_DMA	
函数原型	HAL_StatusTypeDef HAL_UART_Transmit_DMA (UART_HandleTypeDef *huart,uint8_t *pData, uint16_t Size)
功能描述	在DMA方式下发送一定数量的数据
入口参数1	huart: 串口句柄的地址
入口参数2	pData: 待发送数据的首地址
入口参数3	Size: 待发送数据的个数
返回值	HAL状态值: HAL_OK表示发送成功; HAL_ERROR表示参数错误; HAL_BUSY表示串口被占用
注意事项	1. 该函数将启动DMA方式的串口数据发送 2. 完成指定数量的数据发送后, 可以触发DMA中断, 在中断中将调用发送中断回调函数HAL_UART_TxCpltCallback进行后续处理 3. 该函数由用户调用

2.串口DMA方式接收函数

2 串口DMA方式接收函数：HAL_UART_Receive_DMA	
函数原型	HAL_StatusTypeDef HAL_UART_Receive_DMA (UART_HandleTypeDef *huart,uint8_t *pData, uint16_t Size)
功能描述	在DMA方式下接收一定数量的数据
入口参数1	huart: 串口句柄的地址
入口参数2	pData: 待接收数据的首地址
入口参数3	Size: 待接收数据的个数
返回值	HAL状态值: HAL_OK表示接收成功; HAL_ERROR表示参数错误; HAL_BUSY表示串口被占用
注意事项	1. 该函数将启动DMA方式的串口数据接收 2. 完成指定数量的数据接收后, 可以触发DMA中断, 在中断中将调用接收中断回调函数HAL_UART_RxCpltCallback进行后续处理 3. 该函数由用户调用。

3. 获取未传输数据个数

3 获取未传输数据个数函数：__HAL_DMA_GET_COUNTER

接口函数：__HAL_DMA_GET_COUNTER	
函数原型	__HAL_DMA_GET_COUNTER(__HANDLE__)
功能描述	获取DMA数据流中未传输数据的个数
参数	__HANDLE__：串口句柄的地址
返回值	无
注意事项	1. 该函数是宏函数，进行宏替换，不发生函数调用 2. 函数需要由用户调用，用于获取未传输数据的个数

不定长数据的收发：

一.空闲中断

空闲中断

空闲中断的特点

1. 在一帧数据传输结束后，通信线路将会维持高电平，这个状态称为空闲状态；

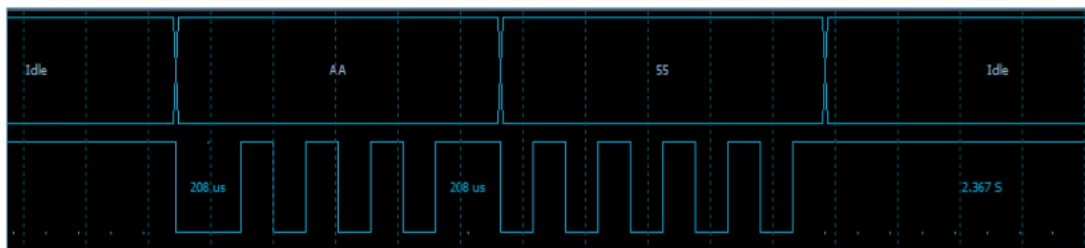
2. 当CPU检测到通信线路处于空闲状态时，空闲状态标志IDLE将由硬件置1。如果串口控制寄存器CR1中的IDLEIE位为1，将会触发空闲中断（IDLE中断）；

3. 由于空闲标志是在一帧数据传输完成后才置位，在有效数据传输过程中不会置位，因此借助空闲中断，可以实现不定长数据的收发。

时序图

空闲中断时序图

假设一帧数据由2个字符构成，分别是0xaa、0x55，传输的时序图如下：



从图中可以看到，在发送该帧数据之前和之后，数据线为高电平，处于IDLE状态。在该帧有效数据发送过程中，即0xaa字节与0x55字节之间，没有IDLE状态出现，这样利用IDLE状态就可以判断不定长数据的发送结束。

不定长数据的收发：

设计思路：

设计思路

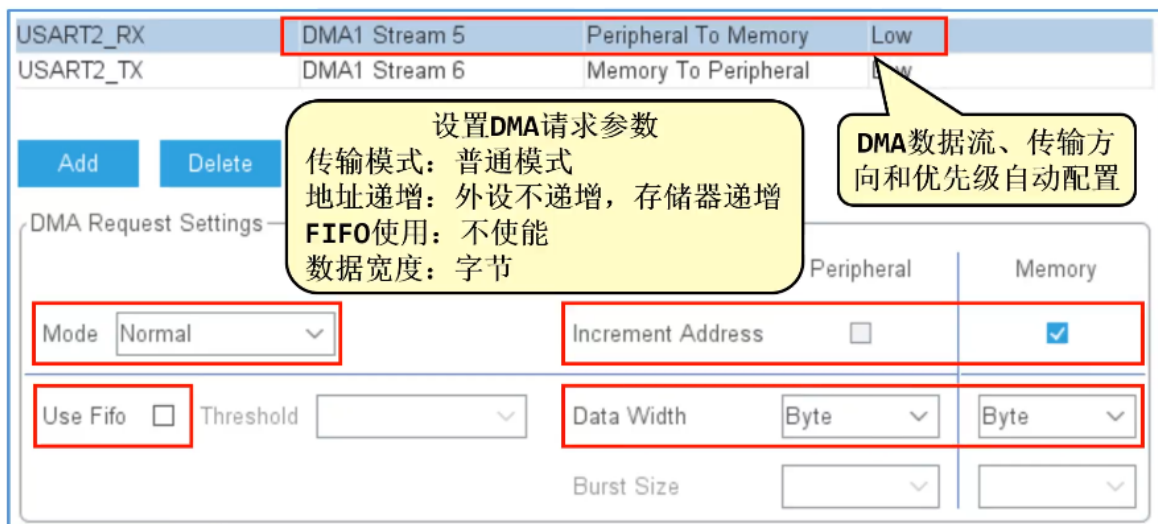
设计思路

1. 使能IDLE中断，在串口2的中断服务程序USART2_IRQHandler中添加对IDLE中断的判断，该函数位于stm32f4xx_it.c文件；
2. 设置传输模式为普通模式，启动DMA传输。串口一旦接收到数据，则触发DMA操作，将数据存放到用户定义的接收缓冲区；
3. 当一帧数据发送完成后，线路处于IDLE状态，将触发IDLE中断，调用IDLE中断回调函数，设置数据接收完成标志；
4. 主程序检测到接收完成标志置位后，将接收的一帧数据原样发回到PC，并禁能DMA，以触发DMA中断。DMA中断将调用接收中断回调函数，在回调函数中重新启动DMA传输。

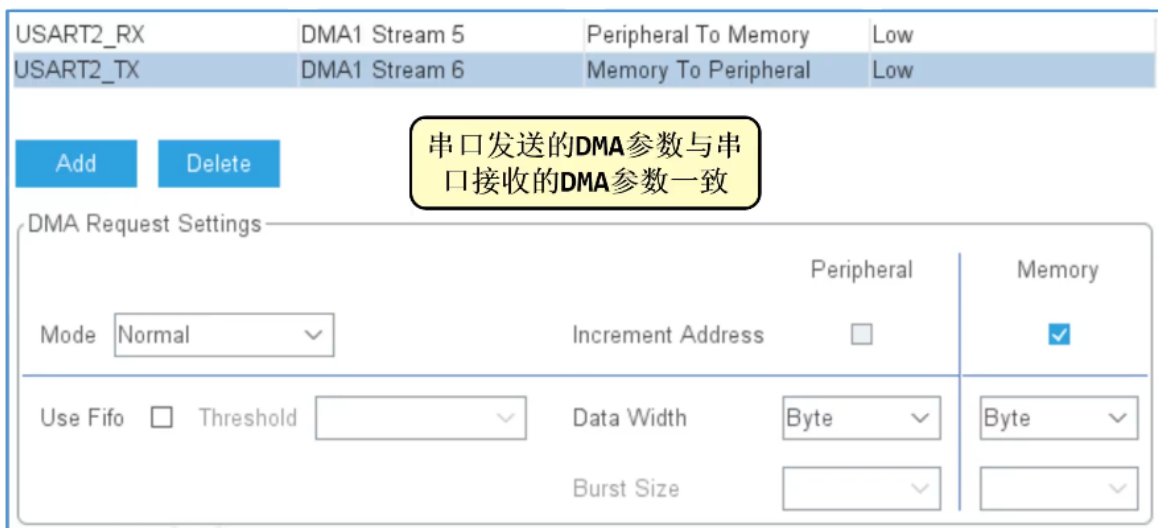
1. CubeMX配置



设置串口接收的DMA参数



设置串口发送的DMA参数



使能串口中断及DMA中断

DMA数据流的中断使能
由CubeMX自动勾选

Parameter Settings	User Constants	NVIC Settings	DMA Settings	GPIO Settings
NVIC Interrupt Table		Enabled	Preemption Priority	Sub Priority
DMA1 stream5 global interrupt		<input checked="" type="checkbox"/>	0	0
DMA1 stream6 global interrupt		<input checked="" type="checkbox"/>	0	0
USART2 global interrupt		<input checked="" type="checkbox"/>	0	0

使能串口2的中断
中断优先级使用默认值