[**STM32-DMA学习笔记**](http://blog.csdn.net/zzwdkxx/article/details/9026173)

**DMA**，全称为：Direct Memory Access，即直接存储器访问。DMA传输方式无需CPU 直接控制传输，也没有中断处理方式那样保留现场和恢复现场的过程，通过硬件为RAM 与I/O设备开辟一条直接传送数据的通路，能使CPU 的效率大为提高。

       STM32中 DMA1有7个通道，DMA2有5个通道（DMA2 仅存在大容量产品中）。DMA挂载的时钟为AHB总线,其时钟为72Mhz，所以可以实现高速数据搬运。
       STM32F103RBT6 只有1 个DMA控制器，DMA1 ，下面我们就针对DMA1 进行介绍。
       从外设（TIMx、ADC、SPIx 、I2Cx 和USARTx ）产生的DMA请求，通过逻辑或输入到DMA控制器，这就意味着同时只能有一个请求有效。外设的DMA请求，可以通过设置相应的外设寄存器中的控制位，被独立地开启或关闭。

       DMA1各通道一览：

这里我们要使用的是串口 1 的 DMA 传送，也就是要用到通道 4。

**DMA1通道4的配置方法如下：**







**dma.c主要代码：**

1. u16 DMA1\_MEM\_LEN;//保存DMA每次数据传送的长度
2. //DMA1的各通道配置
3. //这里的传输形式是固定的,这点要根据不同的情况来修改
4. //从存储器->外设模式/8位数据宽度/存储器增量模式
5. //DMA\_CHx:DMA通道CHx
6. //cpar:外设地址
7. //cmar:存储器地址
8. //cndtr:数据传输量
9. void MYDMA\_Config(DMA\_Channel\_TypeDef\*DMA\_CHx,u32 cpar,u32 cmar,u16 cndtr)
10. {
11. RCC->AHBENR|=1<<0;     //开启DMA1时钟
12. delay\_ms(1);            //等待DMA时钟稳定
13. DMA\_CHx->CPAR=cpar;      //DMA1 外设地址
14. DMA\_CHx->CMAR=(u32)cmar;//DMA1,存储器地址
15. DMA1\_MEM\_LEN=cndtr;     //保存DMA传输数据量
16. DMA\_CHx->CNDTR=cndtr;   //DMA1,传输数据量
17. DMA\_CHx->CCR=0X00000000;//复位
18. DMA\_CHx->CCR|=1<<4;  //从存储器读
19. DMA\_CHx->CCR|=0<<5;  //普通模式
20. DMA\_CHx->CCR|=0<<6;  //外设地址非增量模式
21. DMA\_CHx->CCR|=1<<7;  //存储器增量模式
22. DMA\_CHx->CCR|=0<<8;  //外设数据宽度为8位
23. DMA\_CHx->CCR|=0<<10; //存储器数据宽度8位
24. DMA\_CHx->CCR|=1<<12; //中等优先级
25. DMA\_CHx->CCR|=0<<14; //非存储器到存储器模式
26. }
27. //开启一次DMA传输
28. void MYDMA\_Enable(DMA\_Channel\_TypeDef\*DMA\_CHx)
29. {
30. DMA\_CHx->CCR&=~(1<<0);       //关闭DMA传输
31. DMA\_CHx->CNDTR=DMA1\_MEM\_LEN; //DMA1,传输数据量
32. DMA\_CHx->CCR|=1<<0;          //开启DMA传输
33. }

u16 DMA1\_MEM\_LEN;//保存DMA每次数据传送的长度

//DMA1的各通道配置

//这里的传输形式是固定的,这点要根据不同的情况来修改

//从存储器->外设模式/8位数据宽度/存储器增量模式

//DMA\_CHx:DMA通道CHx

//cpar:外设地址

//cmar:存储器地址

//cndtr:数据传输量

void MYDMA\_Config(DMA\_Channel\_TypeDef\*DMA\_CHx,u32 cpar,u32 cmar,u16 cndtr)

{

 RCC->AHBENR|=1<<0; //开启DMA1时钟

 delay\_ms(1); //等待DMA时钟稳定

 DMA\_CHx->CPAR=cpar; //DMA1 外设地址

 DMA\_CHx->CMAR=(u32)cmar;//DMA1,存储器地址

 DMA1\_MEM\_LEN=cndtr; //保存DMA传输数据量

 DMA\_CHx->CNDTR=cndtr; //DMA1,传输数据量

 DMA\_CHx->CCR=0X00000000;//复位

 DMA\_CHx->CCR|=1<<4; //从存储器读

 DMA\_CHx->CCR|=0<<5; //普通模式

 DMA\_CHx->CCR|=0<<6; //外设地址非增量模式

 DMA\_CHx->CCR|=1<<7; //存储器增量模式

 DMA\_CHx->CCR|=0<<8; //外设数据宽度为8位

 DMA\_CHx->CCR|=0<<10; //存储器数据宽度8位

 DMA\_CHx->CCR|=1<<12; //中等优先级

 DMA\_CHx->CCR|=0<<14; //非存储器到存储器模式

}

//开启一次DMA传输

void MYDMA\_Enable(DMA\_Channel\_TypeDef\*DMA\_CHx)

{

 DMA\_CHx->CCR&=~(1<<0); //关闭DMA传输

 DMA\_CHx->CNDTR=DMA1\_MEM\_LEN; //DMA1,传输数据量

 DMA\_CHx->CCR|=1<<0; //开启DMA传输

}

}  

**在主函数里主要有这几个语句完成DMA传输：**

**1.首先配置DMA1通道4相关参数**

**2.然后将待发送内容装入存储器**

**3.然后开启一次DMA传输**

**4.监控传送进度。**