

32 位 微控制器

HC32F146/HC32M140 系列的 DMA

适用对象

系列	产品型号
HC32F146	HC32F146F8TA
	HC32F146J8UA
	HC32F146J8TA
	HC32F146KATA
HC32M140	HC32M140F8TA
	HC32M140J8UA
	HC32M140J8TA
	HC32M140KATA

目 录

1	摘要	3
2	简介	3
3	HC32F146 / HC32M140 系列的 DMA	4
3.1	简介	4
3.2	说明	4
3.2.1	寄存器介绍	4
3.2.2	工作流程介绍	4
4	样例代码	8
4.1	代码介绍	8
4.2	代码运行	9
5	总结	11
6	版本信息	12

1 摘要

本篇应用笔记主要介绍如何使用 HC32F146 / HC32M140 系列芯片的 DMA 模块传输数据。

2 简介

什么是 DMA？

DMA（直接内存访问控制器）功能块可以不通过 CPU 高速传输数据。使用 DMA 能提高系统性能。

DMA 的重要特征？

DMA 独立于 CPU 总线的总线，所以即便是在使用 CPU 总线的时候，DMA 也可进行传输操作。

3 HC32F146 / HC32M140 系列的 DMA

3.1 简介

华大 HC32F146 / HC32M140 系列 MCU 内部集成 DMAC 模块，可以不通过 CPU 高速的传输数据。

3.2 说明

- DMAC 配有独立于 CPU 总线的总线，所以即便是在使用 CPU 总线的时候，DMAC 也可进行传输操作。
- 最多由 2 条通道组成，能执行 2 种相互独立的 DMA 传输。
- 可设置传输目标地址、传输源地址、传输数据大小、传输请求源以及传输模式，并能控制各通道的传输操作启动、传输的强行终止以及传输的暂停。
- 可控制所有通道批量传输的启动、强行终止及暂停。
- 多通道同时操作，可用固定方法或循环方法选择操作通道的优先级。
- 支持使用外设中断信号的硬件 DMA 传输。
- 遵从系统总线，支持 32 位地址空间。

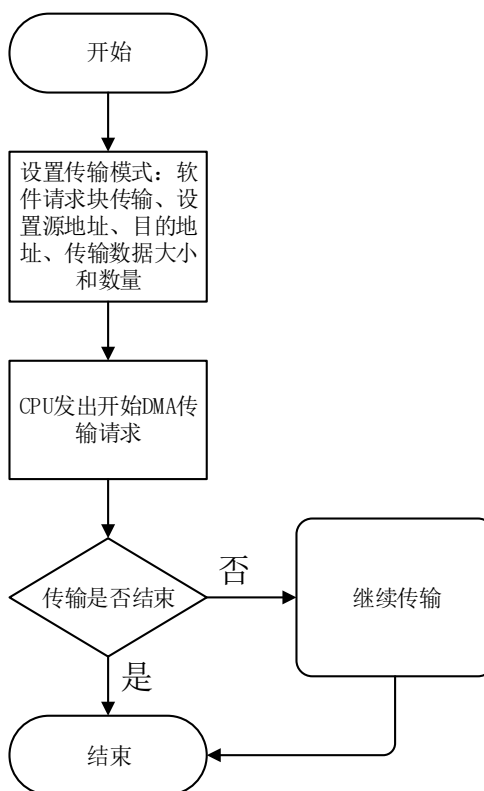
3.2.1 寄存器介绍

- 1) DMACR: 整体 DMAC 配置寄存器，主要配置 DMA 的整体功能，开始或者停止 DMA 传输。
- 2) DMACA: 配置 A 寄存器，主要配置 DMA 的输入源，以及传输计数。
- 3) DMACB: 配置 B 寄存器，主要是 DMA 传输模式的相关配置。
- 4) DMACSA: 传输源地址寄存器，主要是配置 DMA 的传输源地址。
- 5) DMACDA: 传输目的地址寄存器，主要是配置 DMA 的目的地址。

3.2.2 工作流程介绍

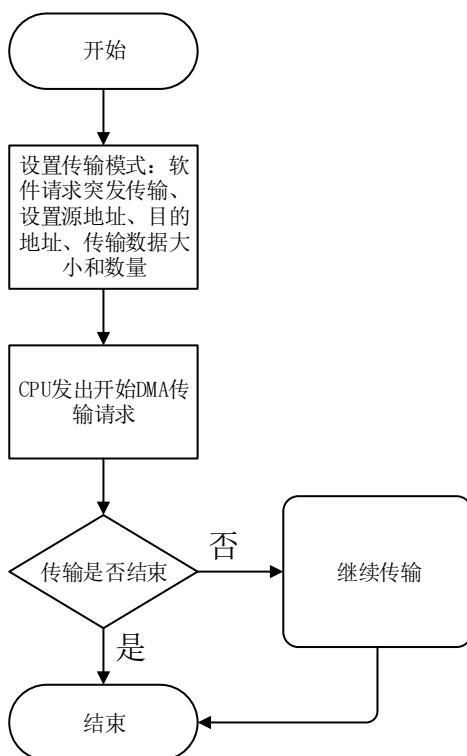
在本章节主要介绍 DMA 的设置和运行流程，主要有 4 种传输模式。

1) 软件-块传输



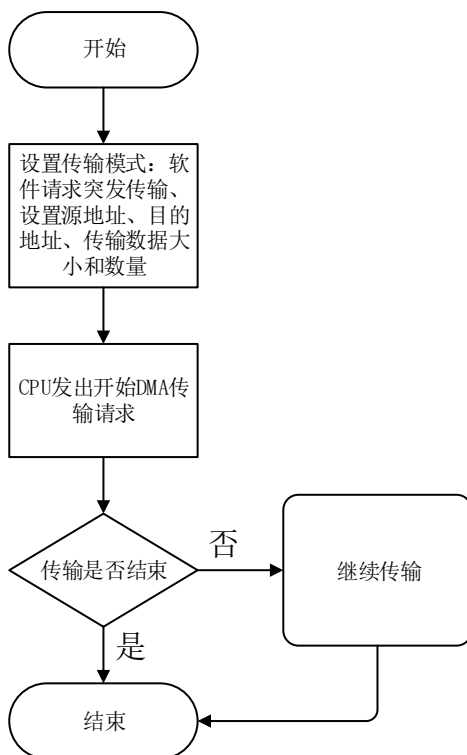
块传输时，每完成一次块传输会产生一个传输间隔，传输完成后 DMAC 会通知 CPU 传输完成。

2) 软件-突发传输

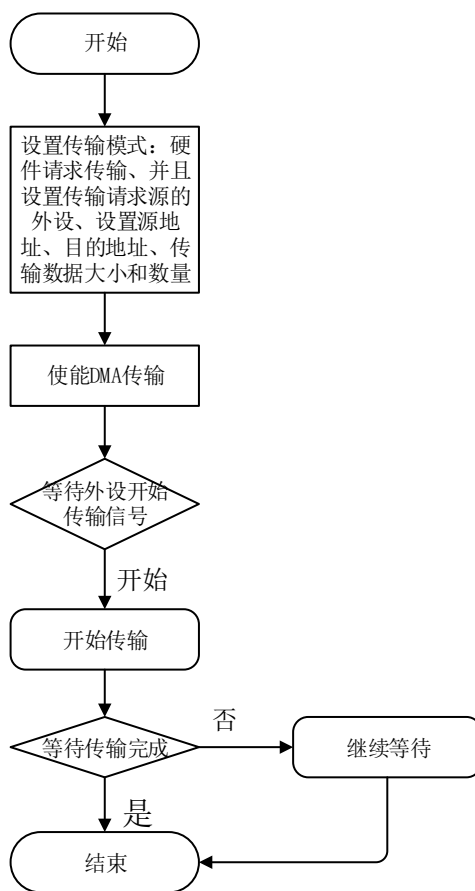


和块传输不同的是突发传输不会产生传输间隔，传输完成后 DMAC 会通知 CPU 传输完成。

3) 硬件-请求传输



4) 硬件-块传输及突发传输



通过 CPU 指定传输内容，指令启动操作，DMAC 等待外设的传输请求，收到传输请求后，DMAC 执行一次传输，然后等待下一次传输请求。

4 样例代码

4.1 代码介绍

用户可根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载到设备驱动库（Device Driver Library, DDL）的样例代码并使用其中的 DMAC 的 Example 进行验证。

以下部分简要介绍该代码 DMA 软件突发传输模式的各个部分。

1) 配置 DMAC 的相关参数：

```
stcDma0Cfg.bFixedDestination = FALSE;
stcDma0Cfg.bFixedSource = FALSE;
stcDma0Cfg.bReloadCount = TRUE;
stcDma0Cfg.bReloadDestination = TRUE;
stcDma0Cfg.bReloadSource = TRUE;
stcDma0Cfg.bTouchNvic = TRUE;
stcDma0Cfg.enDmaIdrq = Software;
stcDma0Cfg.enTransferMode = DmaBlockTransfer;
stcDma0Cfg.enTransferWidth = Dma8Bit;
stcDma0Cfg.pstcIrqCb = &stcIrqCb;
stcDma0Cfg.pstcIrqEn = &stcIrqSel;
stcDma0Cfg.u16TransferCount = DMAC0_TC;
stcDma0Cfg.u8BlockCount = DMAC0_BC;
stcDma0Cfg.u32DestinationAddress = (uint32_t)&au8Des[0];
stcDma0Cfg.u32SourceAddress = (uint32_t)&au8Src[0];
```

2) 使能传输完成中断：

```
stcIrqSel.bCompleteIrq = TRUE;
stcIrqCb.pfnDmaCompletionIrqCb = Dma0CompleteCb;
```

3) 写入配置：

```
Dma_InitChannel(DMACH0, &stcDma0Cfg);
Dma_SetChannel(DMACH0, TRUE, FALSE, TRUE);
```

4) 使能 DMA 传输：

```
Dma_Enable();
```

5) 等待传输完成，传输完成后，会产生传输完成中断，在中断处理函数中，置传输完成标记。

```
while(trans_done != enDmaTransStat)
{
    asm("nop");
}
```

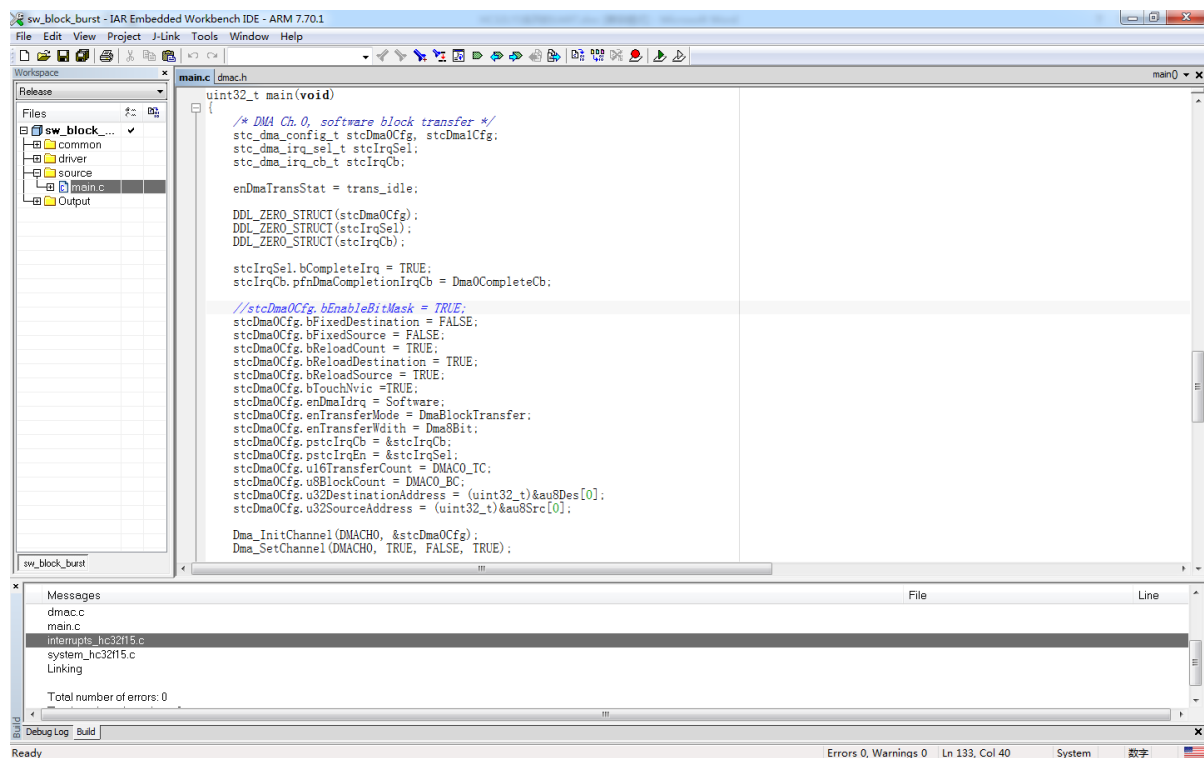

4.2 代码运行


用户可以通过华大半导体的网站下载到 DMA 的样例代码，并配合评估用板（SK-HC32F146_64L V10）运行相关代码学习使用 DMA 模块。

以下部分主要介绍如何在评估板上运行 DMA 样例代码并观察结果：

- 确认安装正确的 IAR（或 Keil，此处使用 IAR 做样例说明，Keil 中操作方法类似）工具（请从华大半导体完整下载相应的安装包，并参考用户手册进行安装）。
- 从华大半导体网站下载 DMA 样例代码。
- 下载并运行样例代码：

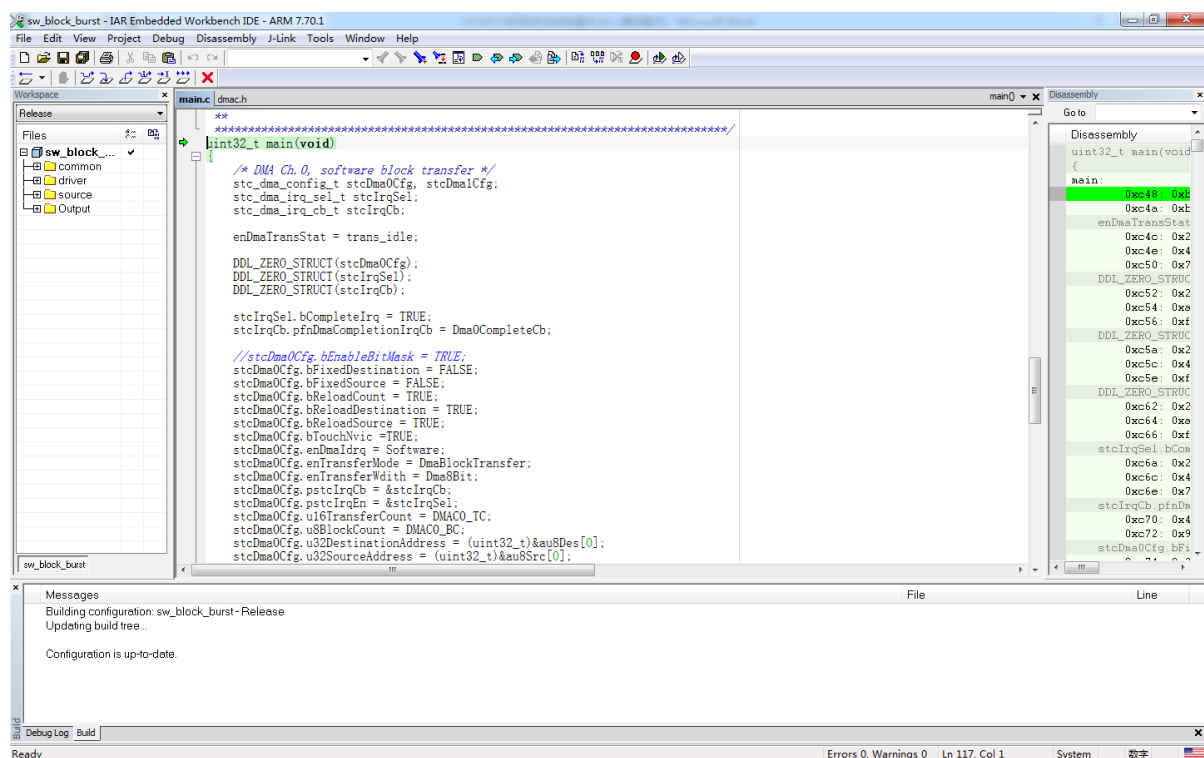
1) 打开 DMA-> sw_block_burst 项目，并打开‘main.c’如下视图：



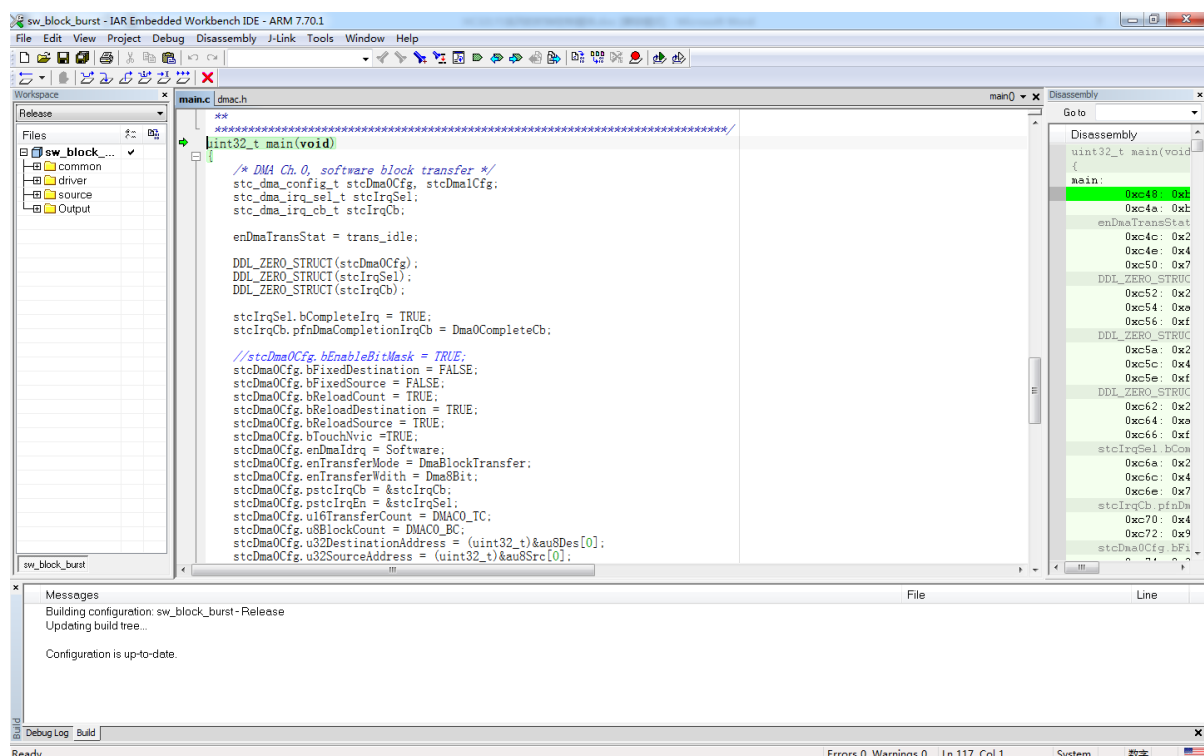
2) 点击  重新编译链接整个项目。

3) 点击  将代码下载到评估板上。

4) 可以看见类似如下的视图:

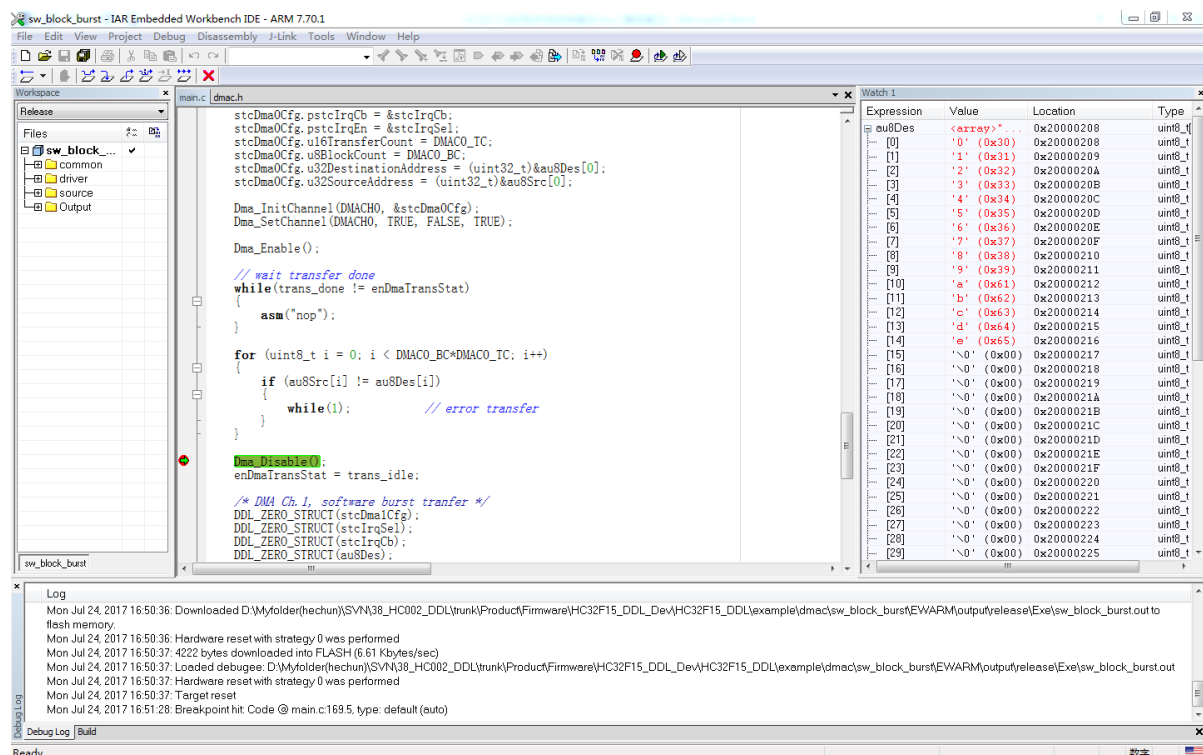


5) 将全局变量‘au8Des’添加到‘watch1’窗口中，如下图所示:



6) 点击  运行。

7) 运行后按  停止，观察并记录‘au8Des’的值。



8) 运行完毕后可以关闭项目文件。

9) 用户亦可通过修改代码中 DMA 的相关配置参数或初始化数据来进一步学习 DMA 模块的功能。

5 总结

以上章节简要介绍了 HC32F146 / HC32M140 系列的 DMA，并详细说明了 DMA 模块的寄存器及操作流程，并且演示了如何使用 DMA，在实际开发中用户可以根据自己的需要配置和使用 DMA。

6 版本信息

日期	版本	修改记录
2017-11-13	Rev1.0	DMA 控制器应用笔记初版发布



如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: mcu@hdsc.com.cn

网址: www.hdsc.com.cn

通信地址: 上海市张江高科园区碧波路 572 弄 39 号

邮编: 201203

