
如何使用 PIC32MX/PIC32MZ/PIC32MM 器件上的 DMA CRC 生成器

简介

循环冗余校验（Cyclic Redundancy Check, CRC）是一种稳健的错误校验算法，用于在处理数据之前确保数据的完整性。CRC 值（校验和）与报文或特定数据块相关联。无论是用于通信的数据包还是存储器中存储的数据块，都可以在处理之前利用 CRC 帮助进行验证。

CRC 计算是一个迭代过程，在软件中实现时需要占用大量 CPU 带宽。通过把带有 CRC 块的特殊功能模块（Special Function Module, SFM）集成到 PIC32MX/PIC32MZ/PIC32MM 器件上的 DMA 模块中，可有助于以最少的软件开销来实现快速高效的 CRC 校验和计算。

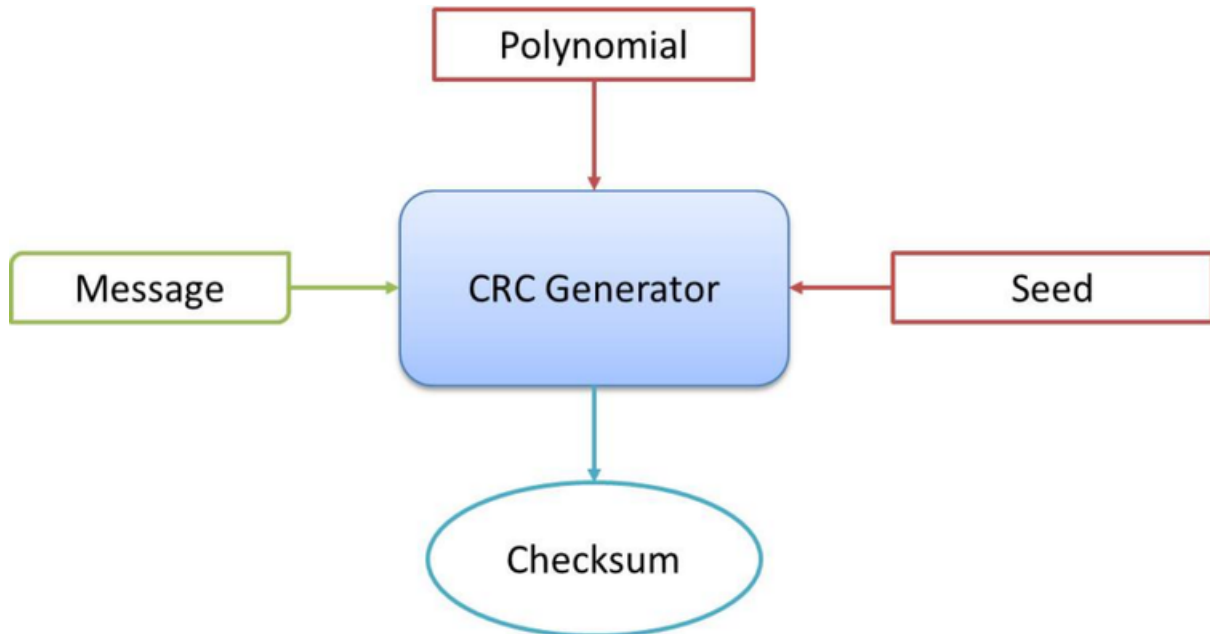
目录

简介.....	1
1. 概念.....	3
2. 解决方案.....	6
3. 相关资源.....	13
Microchip 网站.....	14
变更通知客户服务.....	14
客户支持.....	14
Microchip 器件代码保护功能.....	14
法律声明.....	15
商标.....	15
DNV 认证的质量管理体系.....	16
全球销售及服务网点.....	17

1. 概念

CRC 生成器引擎的框图如下所示。

图 1-1. CRC 生成器



多项式

CRC 计算是执行 XOR 运算的迭代过程。CRC 算法使用多项式进行计算。利用数字来表示除数、被除数和余数的式子称为二进制系数多项式。

例如，数字 19h (11001b) 可用以下公式来表示：

$$(1 \cdot x^4) + (1 \cdot x^3) + (0 \cdot x^2) + (0 \cdot x^1) + (1 \cdot x^0) \text{ 或 } x^4 + x^3 + x^0$$

该应用程序借助合适的生成器多项式来使用 CRC 检测错误。下表列出了一些广泛使用的多项式。

表 1-1. 常用 CRC 的多项式函数

名称	十六进制值	多项式
CRC-16	0x8005	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	0x1021	$x^{16} + x^{12} + x^5 + 1$
CRC-XMODEM	0x8408	$x^{16} + x^{15} + x^{10} + x^3$
CRC-32	0x04C11DB7	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$

种子

种子是算法开始计算时使用的 CRC 的期望初始值。种子可以是直接值，也可以是非直接值。直接种子算法不需要在报文末尾添加 n 个零位就能进行 n 位 CRC 计算。非直接种子算法需要在报文末尾添加 n 个零位才能进行 n 位 CRC 计算。16 位 CRC 计算需要在报文末尾附加 16 个零位，32 位 CRC 计算需要在报文末尾附加 32 个零位。

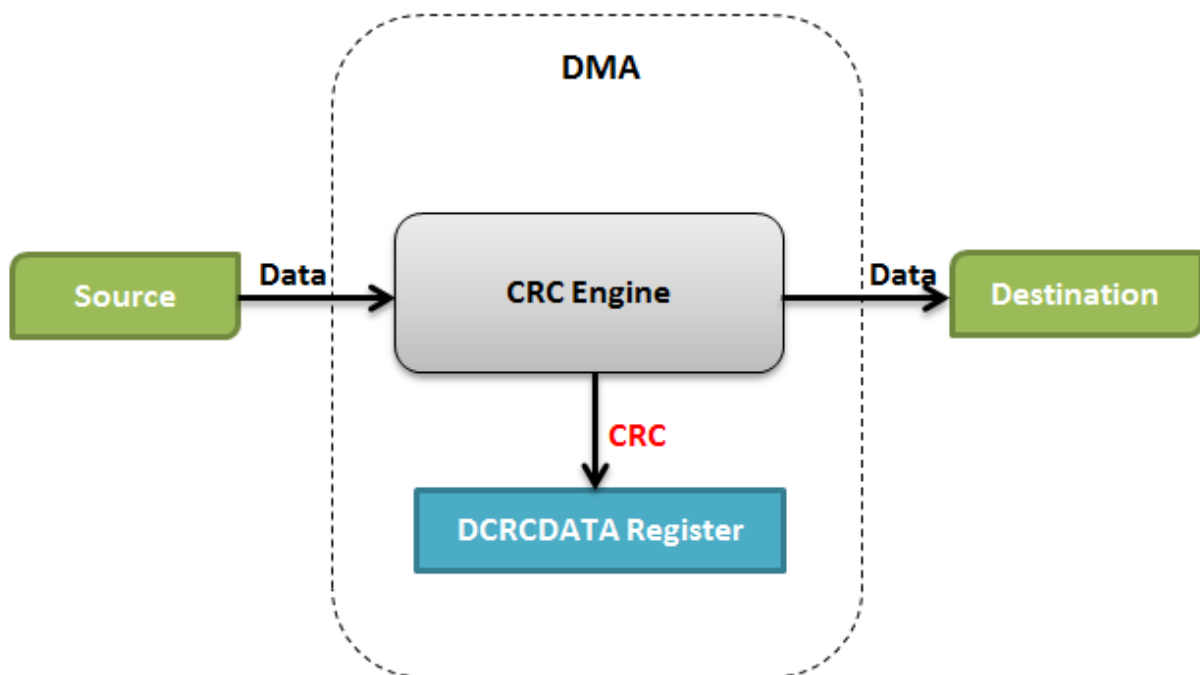
非直接种子可通过 XOR 直接种子获得。无论使用直接还是非直接种子算法，计算得到的 CRC 是相同的。选择初始种子为直接值还是非直接值取决于 CRC 算法是否可避免在报文末尾附加零位。

注： PIC32MX/PIC32MZ/PIC32MM 器件上的 DMA 模块带有一个 SFM。SFM 包含一个线性反馈移位寄存器（Linear Feedback Shift Register, LFSR）CRC 引擎。借助连接到 DMA 的 LFSR CRC 引擎，可在通过 DMA 通道将数据从源传输到目标时进行并行 CRC 计算。DMA CRC 引擎与直接和非直接种子 CRC 计算都兼容。DMA CRC 引擎有两种运行模式，即后台模式和追加模式。

后台模式

连接到 PIC32MX/PIC32MZ/PIC32MM 器件上 DMA 的 CRC 引擎能以后台模式工作。在此模式下，用于 CRC 计算的 DMA 通道保持默认行为。DMA 从源读取数据，通过 CRC 模块传递数据，并将其写入目标，如下图所示。

图 1-2. 后台模式 CRC

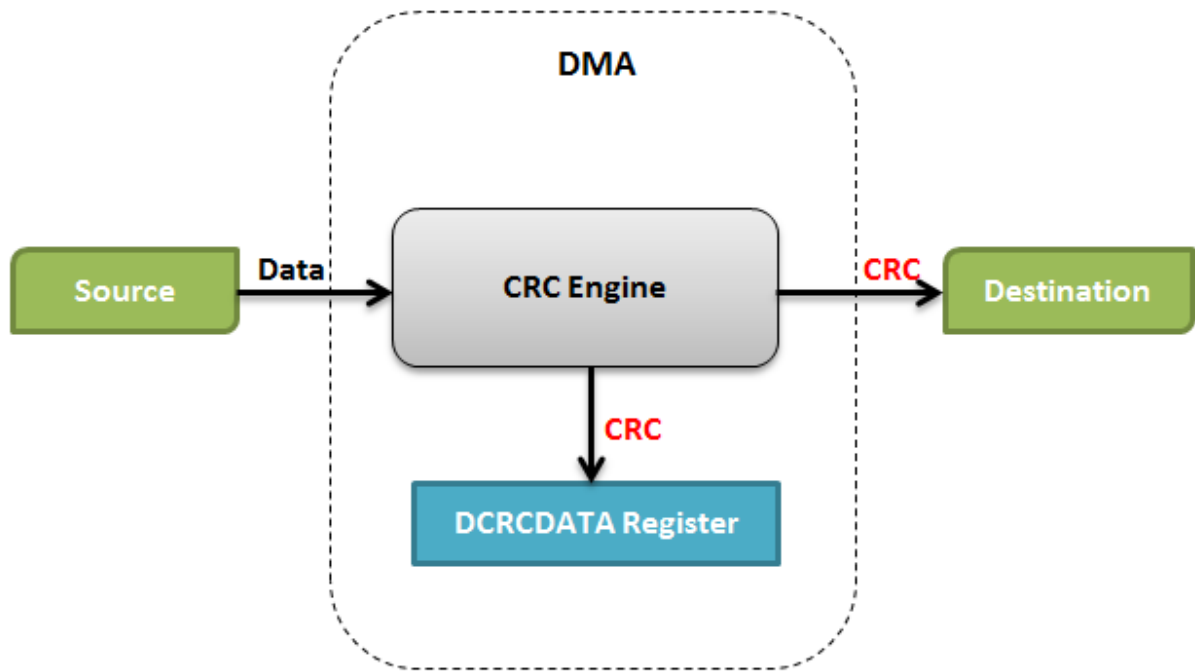


当数据传输完成后，在 DCRCDATA 寄存器中可获得计算出的 CRC。

追加模式

连接到 PIC32MX/PIC32MZ/PIC32MM 器件上 DMA 的 CRC 引擎还具备追加模式。在此模式下，用于 CRC 计算的 DMA 通道仅将数据从源发送到 CRC 引擎。它不会将数据写入目标，而是在数据传输完成时，将计算出的 CRC 写入目标和 DCRCDATA 寄存器，如下图所示。

图 1-3. 追加模式 CRC



2. 解决方案

Microchip MPLAB Harmony 软件框架下的 DMA 系统服务为实现 CRC 计算提供支持。以下是在 MPLAB Harmony 项目中使用与 DMA 外设相连的 CRC 引擎来计算数据 CRC 的步骤。

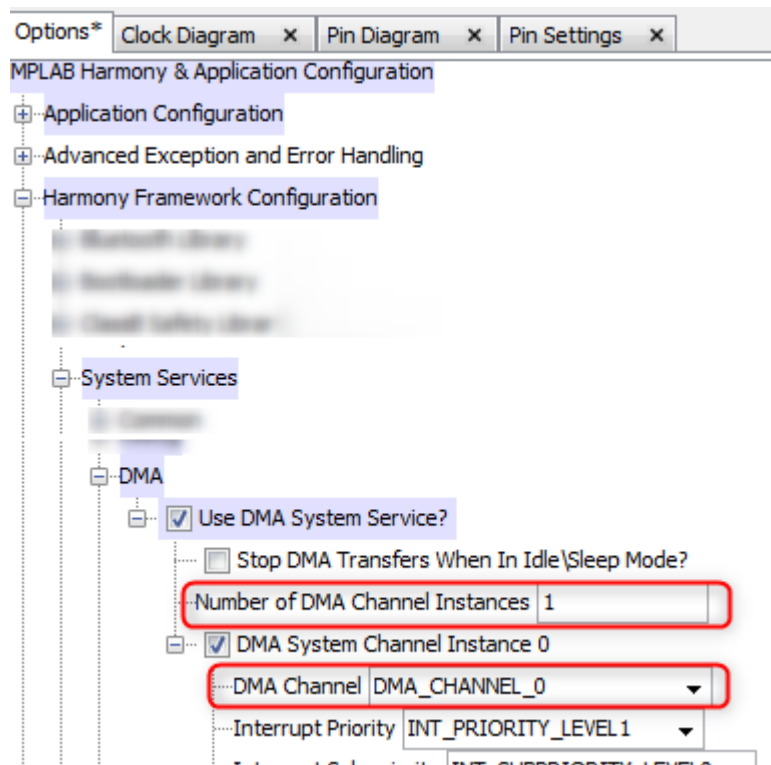
1. 使用 MHC 配置 DMA 通道。
2. 打开 DMA 通道。
3. 注册 DMA 通道事件处理程序。
4. 设置 DMA 工作特性。
5. 设置 CRC 计算。
6. 设置和启动数据传输。
7. 观察 CRC 结果。

使用 MHC 配置 DMA 通道：

用于 CRC 计算的 DMA 通道可使用 MPLAB Harmony 配置器（MPLAB Harmony Configurator, MHC）工具进行配置，如下图所示。

展开 *MPLAB Harmony Configurator*（*MPLAB Harmony 配置器*）> *Options*（*选项*）> *System Services*（*系统服务*）> *DMA* 选项树。

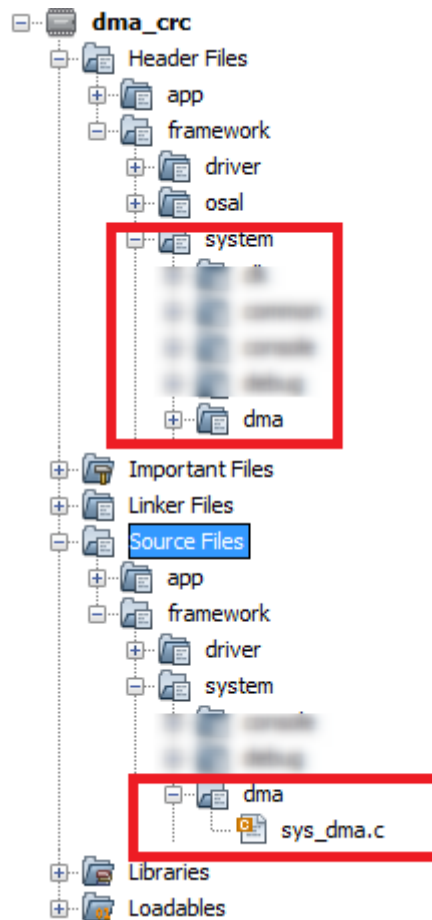
图 2-1. DMA 通道配置



注：

1. 如果要将数据传输到与 Harmony 驱动程序相关联的外设或从该外设传输数据，可将配置的 DMA 通道与 Harmony 驱动程序（USART 和 SPI 等）进行关联。
2. 上述配置会将 DMA 系统服务库添加到 Harmony 项目中。当使用上述 DMA 通道配置生成代码时，将会向该项目添加如下图所示的文件夹结构。

图 2-2. DMA 系统服务文件夹



3. 生成的代码通过调用文件 `system_init.c` 中的函数 `SYS_DMA_Initialize` 来初始化配置的 DMA 通道。

打开 DMA 通道:

如下图所示，在应用程序代码中调用 DMA 系统服务函数 `SYS_DMA_ChannelAllocate` 来获取配置的 DMA 通道的句柄。

图 2-3.

```
SYS_DMA_CHANNEL_HANDLE channelHandle;
/* Allocate a DMA channel */
channelHandle = SYS_DMA_ChannelAllocate(DMA_CHANNEL_0);
```

注：如果配置的 DMA 通道与 Harmony 驱动程序（USART 和 SPI 等）相关联，则应用程序不需要显式调用 `SYS_DMA_ChannelAllocate` 函数，因为该通道将由驱动程序分配。

注册 DMA 通道事件处理程序:

如下图所示，在应用程序代码中调用 DMA 系统服务函数 `SYS_DMA_ChannelTransferEventhandlerSet` 来为打开的 DMA 通道注册事件处理程序。

图 2-4.

```
uint32_t sysContext;

static void App_Mem2Mem_Event_Handler(SYS_DMA_TRANSFER_EVENT event,
SYS_DMA_CHANNEL_HANDLE handle, uintptr_t contextHandle);

/* Register an event handler for the channel */
SYS_DMA_ChannelTransferEventHandlerSet(channelHandle,
App_Mem2Mem_Event_Handler, (uintptr_t)&sysContext);
```

注：如果配置的 DMA 通道与 Harmony 驱动程序（USART 和 SPI 等）相关联，则应用程序不需要显式调用 `SYS_DMA_ChannelTransferEventHandlerSet` 函数，因为该通道的事件处理程序由驱动程序注册。

设置 DMA 工作特性：

如下图所示，在应用程序代码中调用 DMA 系统服务函数 `SYS_DMA_ChannelSetup` 来设置已注册的 DMA 通道（工作模式和通道触发源），以便执行 CRC 运算。

图 2-5.

```
/* Setup the channel */
SYS_DMA_ChannelSetup(channelHandle,
(SYS_DMA_CHANNEL_OP_MODE_BASIC | SYS_DMA_CHANNEL_OP_MODE_CRC ),
DMA_TRIGGER_SOURCE_NONE);
```

注：如果配置的 DMA 通道与 Harmony 驱动程序（USART 和 SPI 等）相关联，可调 DMAHarmonyPLIB 函数 `PLIB_DMA_CRCEnable` 来使能 CRC 运算，如以下代码所示。

```
PLIB_DMA_CRCENABLE ( DMA_ID_0 );
```

设置 CRC 计算：

如下图所示，按照所需值设置 `crc` 结构字段：`type`、`bitOrder`、`byteOrder` 和 `writeOrder`。

图 2-6.

```
SYS_DMA_CHANNEL_OPERATION_MODE_CRC crc;

crc.type           = DMA_CRC_LFSR;
crc.bitOrder       = DMA_CRC_BIT_ORDER_MSB;
crc.byteOrder      = DMA_CRC_BYTEORDER_NO_SWAPPING;
crc.writeOrder     = SYS_DMA_CRC_WRITE_ORDER_MAINTAIN;
crc.mode           = SYS_DMA_CHANNEL_CRC_MODE_BACKGROUND;
```

注：

1. 若希望使用后台模式，请将字段模式设置为 `SYS_DMA_CHANNEL_CRC_MODE_BACKGROUND`。
2. 若希望使用追加模式，则将字段模式设置为 `SYS_DMA_CHANNEL_CRC_MODE_APPEND`。

字段 `xorBitMask`（多项式值）、`polyLength`（以位为单位的多项式长度）和 `data`（初始种子值）是 CRC 引擎的输入。

若要以 0xFFFF 为初始种子值计算 16 位 CRC-CCIT 多项式（0x1021）的 CRC，`crc` 字段的结构设置将如下图所示。

图 2-7.

```

crc.xorBitMask      = 0x1021;
crc.polyLength      = 16;
crc.data            = 0xFFFF;

```

若要以 0xFFFFFFFF 为初始种子值计算 32 位 CRC-32 多项式 (0x4C11DB7) 的 CRC, *crc* 字段的结构设置将如下图所示。

图 2-8.

```

crc.xorBitMask      = 0x4C11DB7;
crc.polyLength      = 32;
crc.data            = 0xFFFFFFFF;

```

如下图所示, 在应用程序代码中调用 DMA 系统服务函数 *SYS_DMA_ChannelCRCSet* 来为用于 CRC 运算的 DMA 通道设置 CRC 计算。

```
SYS_DMA_ChannelCRCSet (channelHandle, crc) ;
```

注:

1. 如果已配置的 DMA 通道与 Harmony 驱动程序 (USART 和 SPI 等) 相关联, 可通过实现以下函数 (使用 DMA PLIB 函数) 来为已配置的 DMA 通道设置 CRC 计算。

图 2-9.

```

void APP_DMA_ChannelCRCSet(DMA_CHANNEL channelNumber,
                           SYS_DMA_CHANNEL_OPERATION_MODE_CRC crc)
{
    PLIB_DMA_CRCChannelSelect(DMA_ID_0, channelNumber);
    PLIB_DMA_CRCTypeSet(DMA_ID_0, crc.type);
    PLIB_DMA_CRCPolynomialLengthSet(DMA_ID_0, (crc.polyLength-1));
    PLIB_DMA_CRCBitOrderSelect(DMA_ID_0, crc.bitOrder);
    PLIB_DMA_CRCByteOrderSelect(DMA_ID_0, crc.byteOrder);
    if(SYS_DMA_CRC_WRITE_ORDER_MAINTAIN == crc.writeOrder)
        PLIB_DMA_CRCWriteByteOrderMaintain(DMA_ID_0);
    else if(SYS_DMA_CRC_WRITE_ORDER_CHANGE == crc.writeOrder)
        PLIB_DMA_CRCWriteByteOrderAlter(DMA_ID_0);
    PLIB_DMA_CRCDataWrite(DMA_ID_0, crc.data);
    PLIB_DMA_CRCXOREnableSet(DMA_ID_0, crc.xorBitMask);
    if(SYS_DMA_CHANNEL_CRC_MODE_APPEND == crc.mode)
        PLIB_DMA_CRCAppendModeEnable(DMA_ID_0);
    else
        PLIB_DMA_CRCAppendModeDisable(DMA_ID_0);
}

```

2. 如以下代码所示, 调用 *APP_DMA_ChannelCRCSet* 函数来为配置的 DMA 通道设置 CRC 计算。

```
APP_DMA_ChannelCRCSet (DMA_CHANNEL_0, crc) ;
```

如果需要使用非直接种子进行 CRC 计算, 为了比较结果, 可将直接种子转换为非直接种子后使用, 如下图所示。

图 2-10.

```

uint32_t CalculateNonDirectSeed(uint32_t seed, uint32_t polynomial, uint8_t polynomialOrder)
{
    uint8_t lsb;
    uint8_t i;
    uint32_t msbmask;
    msbmask = ((unsigned long)1)<<(polynomialOrder-1);
    for (i=0; i<polynomialOrder; i++)
    {
        lsb = seed & 1;
        if (lsb) seed ^= polynomial;
        seed >>= 1;
        if (lsb) seed |= msbmask;
    }
    return seed; // return the non-direct CRC initial value
}

```

如下图所示，在调用 `SYS_DMA_ChannelCRCSet` 或 `APP_DMA_ChannelCRCSet` 函数之前，先调用 `CalculateNonDirectSeed` 函数将直接种子转换为非直接种子。

图 2-11.

```

crc.data = CalculateNonDirectSeed(crc.data, crc.xorBitMask, crc.polyLength);
SYS_DMA_ChannelCRCSet(channelHandle, crc);

```

或

图 2-12.

```

crc.data = CalculateNonDirectSeed(crc.data, crc.xorBitMask, crc.polyLength);
APP_DMA_ChannelCRCSet(DMA_CHANNEL_0, crc);

```

设置和启动数据传输：

存储器之间的数据传输

调用 `SYS_DMA_ChannelTransferAdd` 函数来设置数据传输，并调用 `SYS_DMA_ChannelForceStart` 函数来启动数据传输。如下图所示，在计算 CRC 时，可将 9 个字节的 ASCII 数据“123456789”从闪存传输到 RAM。

图 2-13.

```

const uint8_t flashBuff[] = {'1','2','3','4','5','6','7','8','9'}; // Source
uint8_t attribute ((coherent)) ramBuff[20]; // Destination
/* Add the memory block transfer request */
SYS_DMA_ChannelTransferAdd(channelHandle, flashBuff, 9,
                           ramBuff, 9, 9);
/* Start the DMA transfer */
SYS_DMA_ChannelForceStart(channelHandle);

```

注：如果在 CRC 计算中应用非直接种子（如之前 **设置 CRC 计算** 部分所述），则追加额外的零位（16 位或 32 位，具体取决于多项式长度），如下图所示。

图 2-14. 对于 16 位多项式

```
const uint8_t flashBuff[] = {'1','2','3','4','5','6','7','8','9',0,0}; // Source
uint8_t __attribute__((coherent)) ramBuff[20]; // Destination

/* Add the memory block transfer request */
SYS_DMA_ChannelTransferAdd(channelHandle, flashBuff, 11,
                           ramBuff, 11, 11);
/* Start the DMA transfer */
SYS_DMA_ChannelForceStart(channelHandle);
```

- 在调用函数 `SYS_DMA_ChannelTransferAdd` 时，参数 `srcAddr`（源）和 `destAddr`（目标）需要附加 2 个字节的零位（对应多项式长度）。
- 在调用 `SYS_DMA_ChannelTransferAdd` 时，参数 `srcSize`（源大小）、`destSize`（目标大小）和 `cellSize`（单元大小）需要加 2（取决于多项式长度）。

图 2-15. 对于 32 位多项式

```
const uint8_t flashBuff[] = {'1','2','3','4','5','6','7','8','9',0,0,0,0}; // Source
uint8_t __attribute__((coherent)) ramBuff[20]; // Destination

/* Add the memory block transfer request */
SYS_DMA_ChannelTransferAdd(channelHandle, flashBuff, 13,
                           ramBuff, 13, 13);
/* Start the DMA transfer */
SYS_DMA_ChannelForceStart(channelHandle);
```

- 在调用函数 `SYS_DMA_ChannelTransferAdd` 时，参数 `srcAddr`（源）和 `destAddr`（目标）需要附加 4 个字节的零位（对应多项式长度）。
- 在调用 `SYS_DMA_ChannelTransferAdd` 时，参数 `srcSize`（源大小）、`destSize`（目标大小）和 `cellSize`（单元大小）需要加 4（取决于多项式长度）。

存储器和外设之间的数据传输

调用外设驱动程序函数来设置并使能数据传输。如果在缓冲数据模型中使用 DMA 将数据从存储器传输到 USART（用于传输），则需调用 `DRV_USART_BufferAddWrite` 函数。

注：如果在 CRC 计算过程中应用了非直接种子（如之前 **设置 CRC 计算** 部分所述），则为参数 `buffer`（源）附加额外的零位（2 或 4，取决于多项式长度），并对参数 `size`（大小）加上一个值（2 或 4，取决于多项式长度）。

观察 CRC 计算结果：

- 如果在存储器之间传输数据，数据传输完成时，将调用通过 `SYS_DMA_ChannelTransferEventhandlers` 函数注册的事件处理程序。在事件处理程序中，调用 `SYS_DMA_ChannelCRCGet` 函数并观察生成的 CRC。

图 2-16.

```
uint32_t blockCrc;

static void App_Mem2Mem_Event_Handler(SYS_DMA_TRANSFER_EVENT event,
    SYS_DMA_CHANNEL_HANDLE handle, uintptr_t contextHandle)
{
    /* Success event */
    if(SYS_DMA_TRANSFER_EVENT_COMPLETE == event)
    {
        blockCrc = SYS_DMA_ChannelCRCGet();
    }
    /* Failure Event */
    else if(SYS_DMA_TRANSFER_EVENT_ABORT == event)
    {
    }
}
```

- 如果在存储器和外设之间传输数据，在数据传输完成时，将调用通过外设驱动程序注册的事件处理程序。在驱动程序事件处理程序中，调用 *SYS_DMA_ChannelCRCGet* 函数并观察生成的 CRC。

3. 相关资源

有关其他信息，请下载和参阅下述文档：

循环冗余校验（CRC）

- <http://ww1.microchip.com/downloads/en/AppNotes/01148a.pdf>

CRC 生成与校验

- <http://ww1.microchip.com/downloads/en/AppNotes/00730a.pdf>

32 位可编程循环冗余校验（CRC）

- <http://ww1.microchip.com/downloads/en/DeviceDoc/60001336c.pdf>

面向嵌入式网络的循环冗余码（CRC）多项式选择

- http://users.ece.cmu.edu/~koopman/roes/dsn04/koopman04_crc_poly_embedded.pdf

Microchip 网站

Microchip 网站 <http://www.microchip.com/> 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问，网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 <http://www.microchip.com/>。在“支持”（Support）下，点击“变更通知客户”（Customer Change Notification）服务后按照注册说明完成注册。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（FAE）
- 技术支持

客户应联系其代理商、代表或应用工程师（FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过以下网站获得技术支持：<http://www.microchip.com/support>

Microchip 器件代码保护功能

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿意与关心代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如

果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

法律声明

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip** 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。**Microchip** 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 **Microchip** 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 **Microchip** 免于承担法律责任，并加以赔偿。除非另外声明，否则在 **Microchip** 知识产权保护下，不得暗中以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、**Microchip** 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 和 XMEGA 是 **Microchip Technology Incorporated** 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 为 **Microchip Technology Incorporated** 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICKtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQL、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 为 **Microchip Technology Incorporated** 在美国和其他国家或地区的商标。

SQTP 为 **Microchip Technology Inc.** 在美国的服务标记。

Silicon Storage Technology 为 **Microchip Technology Inc.** 在除美国外的国家或地区的注册商标。

GestIC 是 **Microchip Technology Inc.** 的子公司 **Microchip Technology Germany II GmbH & Co. KG** 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2018, **Microchip Technology Incorporated** 版权所有。

ISBN: 978-1-5224-3499-3

DNV 认证的质量管理体系

ISO/TS 16949

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC[®] MCU 和 dsPIC[®] DSC、KEELOQ[®]跳码器件、串行 EEPROM、单片机外设、非易失性存储器及模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 1-480-792-7200 传真: 1-480-792-7277 技术支持: http://www.microchip.com/support 网址: www.microchip.com 亚特兰大 德卢斯, 乔治亚州 电话: 1-678-957-9614 传真: 1-678-957-1455 奥斯汀, 德克萨斯州 电话: 1-512-257-3370 波士顿 韦斯特伯鲁, 马萨诸塞州 电话: 1-774-760-0087 传真: 1-774-760-0088 芝加哥 艾塔斯卡, 伊利诺伊州 电话: 1-630-285-0071 传真: 1-630-285-0075 达拉斯 艾迪生, 德克萨斯州 电话: 1-972-818-7423 传真: 1-972-818-2924 底特律 诺维, 密歇根州 电话: 1-248-848-4000 休斯敦, 德克萨斯州 电话: 1-281-894-5983 印第安纳波利斯 诺布尔斯维尔, 印第安纳州 电话: 1-317-773-8323 传真: 1-317-773-5453 电话: 1-317-536-2380 洛杉矶 米申维耶霍, 加利福尼亚州 电话: 1-949-462-9523 传真: 1-949-462-9608 电话: 1-951-273-7800 罗利, 北卡罗来纳州 电话: 1-919-844-7510 纽约, 纽约州 电话: 1-631-435-6000 圣何塞, 加利福尼亚州 电话: 1-408-735-9110 电话: 1-408-436-4270 加拿大 - 多伦多 电话: 1-905-695-1980 传真: 1-905-695-2078	中国 - 北京 电话: 86-10-8569-7000 中国 - 成都 电话: 86-28-8665-5511 中国 - 重庆 电话: 86-23-8980-9588 中国 - 东莞 电话: 86-769-8702-9880 中国 - 广州 电话: 86-20-8755-8029 中国 - 杭州 电话: 86-571-8792-8115 中国 - 南京 电话: 86-25-8473-2460 中国 - 青岛 电话: 86-532-8502-7355 中国 - 上海 电话: 86-21-3326-8000 中国 - 沈阳 电话: 86-24-2334-2829 中国 - 深圳 电话: 86-755-8864-2200 中国 - 苏州 电话: 86-186-6233-1526 中国 - 武汉 电话: 86-27-5980-5300 中国 - 西安 电话: 86-29-8833-7252 中国 - 厦门 电话: 86-592-2388138 中国 - 香港特别行政区 电话: 852-2943-5100 中国 - 珠海 电话: 86-756-3210040 台湾地区 - 高雄 电话: 886-7-213-7830 台湾地区 - 台北 电话: 886-2-2508-8600 台湾地区 - 新竹 电话: 886-3-577-8366	澳大利亚 - 悉尼 电话: 61-2-9868-6733 印度 - 班加罗尔 电话: 91-80-3090-4444 印度 - 新德里 电话: 91-11-4160-8631 印度 - 浦那 电话: 91-20-4121-0141 日本 - 大阪 电话: 81-6-6152-7160 日本 - 东京 电话: 81-3-6880-3770 韩国 - 大邱 电话: 82-53-744-4301 韩国 - 首尔 电话: 82-2-554-7200 马来西亚 - 吉隆坡 电话: 60-3-7651-7906 马来西亚 - 檳榔嶼 电话: 60-4-227-8870 菲律宾 - 马尼拉 电话: 63-2-634-9065 新加坡 电话: 65-6334-8870 泰国 - 曼谷 电话: 66-2-694-1351 越南 - 胡志明市 电话: 84-28-5448-2100	奥地利 - 韦尔斯 电话: 43-7242-2244-39 传真: 43-7242-2244-393 丹麦 - 哥本哈根 电话: 45-4450-2828 传真: 45-4485-2829 芬兰 - 埃斯波 电话: 358-9-4520-820 法国 - 巴黎 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 德国 - 加兴 电话: 49-8931-9700 德国 - 哈恩 电话: 49-2129-3766400 德国 - 海布隆 电话: 49-7131-67-3636 德国 - 卡尔斯鲁厄 电话: 49-721-625370 德国 - 慕尼黑 电话: 49-89-627-144-0 传真: 49-89-627-144-44 德国 - 罗森海姆 电话: 49-8031-354-560 以色列 - 赖阿南纳 电话: 972-9-744-7705 意大利 - 米兰 电话: 39-0331-742611 传真: 39-0331-466781 意大利 - 帕多瓦 电话: 39-049-7625286 荷兰 - 德卢内市 电话: 31-416-690399 传真: 31-416-690340 挪威 - 特隆赫姆 电话: 47-7288-4388 波兰 - 华沙 电话: 48-22-3325737 罗马尼亚 - 布加勒斯特 电话: 40-21-407-87-50 西班牙 - 马德里 电话: 34-91-708-08-90 传真: 34-91-708-08-91 瑞典 - 哥德堡 电话: 46-31-704-60-40 瑞典 - 斯德哥尔摩 电话: 46-8-5090-4654 英国 - 沃金厄姆 电话: 44-118-921-5800 传真: 44-118-921-5820