AN1817

在B类应用中结合使用硬件或软件CRC与增强型内核PIC16F1XXX

作者： *Corey Simoncic*
Microchip Technology Inc.

1.0 简介

目前，越来越多的单片机使用B类安全程序来检测安全要求极高的应用中的故障。检测单片机程序存储器中故障的主要方法是使用由IEC 60730标准定义的循环冗余校验（Cyclic Redundancy Check，CRC）。

CRC可用于防止因程序存储器损坏而导致的应用故障，方法是执行周期性校验来确定校验值是否已变化。

本应用笔记将介绍如何实现软件CRC（作为B类安全软件库的一部分）以及实现所选单片机（本文档主要针对PIC16F161X系列）中使用的硬件CRC。

本应用笔记中讨论的两种方法均符合IEC 60730标准H.2.19.3.2，可测试不变存储器中的所有单位故障，覆盖率达99.6%。

有关B类安全软件和本应用笔记的完整示例代码的更多信息，请访问：www.microchip.com/classb。

有关CRC算法的更多信息，请参见“*A Painless Guide on CRC Algorithms*”，作者：Ross N. Williams（1993年8月19日）。

1.1 循环冗余校验

CRC采用一种与多项式长除法极为类似的方法来确定一个惟一校验值，此校验值类似于多项式长除法中的余数。这可通过选择一个非常具体的除数来实现，此除数称为CRC多项式。CRC多项式是为确定任意给定数据流中的最大错误量而选择的惟一多项式。本应用笔记中使用的CRC多项式为CRC-16-ANSI，如图2所示。另一种常用的CRC算法为CRC-16-CCITT。此算法主要用于通信CRC。

在B类应用中使用校验值的方法如下：先运行一个初始CRC，然后周期性地运行CRC来确认校验值未发生变化。

1.2 CRC实现

有多种常用方法可实现CRC。最常用的CRC硬件实现为线性反馈移位寄存器（Linear Feedback Shift Register，LFSR）。图2显示了用于CRC-16-ANSI的LFSR。这种实现将根据所选的CRC算法在相应位置处放置XOR门，以便将数据流馈入CRC。

一种常用的CRC软件实现是使用表。但是，这种方法会占用大量存储空间，因此在存储空间较小的PIC16器件上使用时效率不高。可以选择一种并行计算方法来执行B类安全软件库中的软件CRC。这可通过选择性地使用XOR的使能位来实现，具体取决于不同的奇偶校验形式。

1.3 CRC错误检测

为发现尽可能多的位错误，可以选择性地选取CRC计算中的多项式。CRC-16-CCITT和CRC-16-ANSI等常用CRC多项式均设计为提供最大错误检测范围。这些多项式专为识别所有单位错误、双位错误、奇数位错误和突发错误而设计。

对于上述错误以外的其他错误，各多项式的有效性仍存在一些争议，本应用笔记不对此进行讨论。

1.4 CRC术语

- 多项式——CRC算法的除数。这是各种CRC方法的主要区别。
- 初始值——CRC计算的起始值。大多数CRC算法具有预定义的初始值，用于确保获得最大错误检测范围。
- 填充零——在要执行CRC计算的数据序列末尾追加零。
- 字节顺序——确定数据移入CRC计算时的位顺序；既可先移入最高有效位，也可先移入最低有效位。
- 校验值——CRC计算的最终结果；也可以称作校验和或余数。

© 2015 Microchip Technology Inc.

1.5 B类CRC范围

在本应用笔记中，CRC使用的术语定义如下：

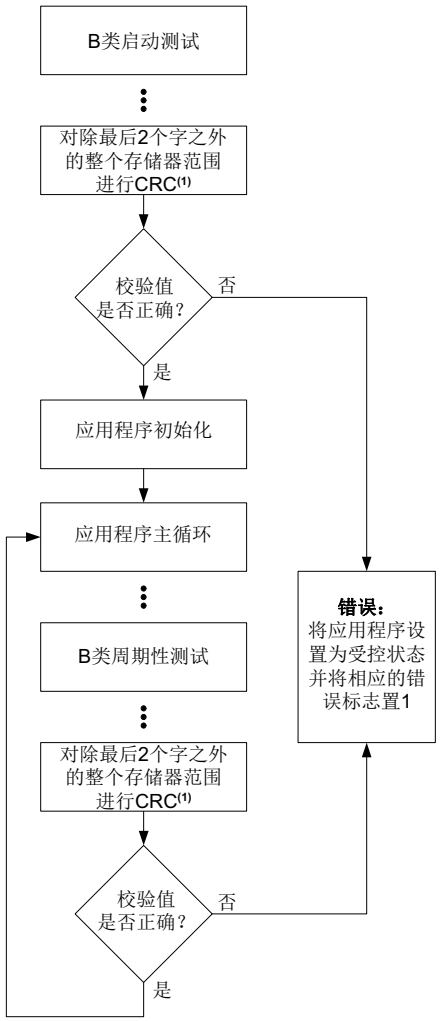
- 多项式 = CRC-16-ANSI
- 初始值 = 0xFFFF
- 将使用填充零
- 字节顺序 = 先移入最高有效位

1.6 在嵌入式应用中使用B类CRC

对于测试嵌入式应用（例如，采用PIC16F1613的应用）的程序存储器而言，CRC方法既准确又可靠。

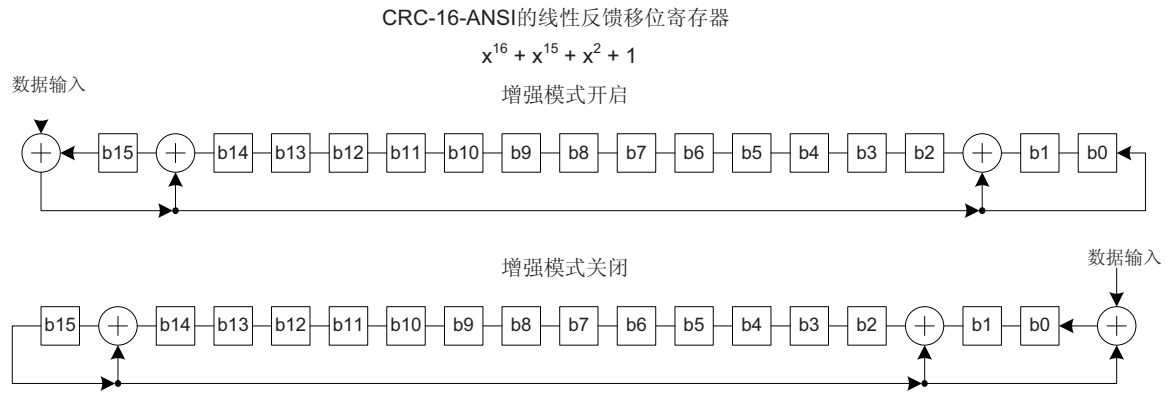
在典型应用中使用CRC（以及其余B类测试）的基本流程如图1所示。本应用笔记随附的CRC计算代码将用于在编程器件之前计算校验和。之后，会将校验值编程到程序存储器的最后两个地址中。这便是每次计算CRC时验证校验值的方式。

图1： B类流程图



注 1： 将使用最后两个地址来存储校验值。

图2： CRC LFSR



© 2015 Microchip Technology Inc.

2.0 用于B类的CRC外设

PIC16F161X系列产品已实现硬件CRC。本节将分步骤说明如何在B类应用中使用CRC外设，其中包括代码示例。

PIC16F161X的CRC外设中包含存储器扫描器。此存储器扫描器可用于将来自器件的闪存程序存储器中的数据直接装入CRC外设。

在PIC16F1613中使用硬件CRC时，第一步是配置用于CRC和存储器扫描器的寄存器。更多信息，请参见PIC16F1613数据手册（DS40001737A_CN）的第11.0节。

2.1 配置CRC模块

CRC-16的多项式：

```
CRCXORH = 0b10000000;
CRCXORL = 0b00000101;
```

最高有效位1由模块自动假设和实现。

初始值：

```
CRCACCH = 0b11111111;
CRCACCL = 0b11111111;
```

使能填充零：

```
CRCCON0bits.ACCM = 1;
```

字节顺序（首先移入最高有效位）：

```
CRCCON0bits.SHIFTM = 0;
```

为保持适应性，CRC模块设计为接受一系列不同的多项式和数据字符串长度。因此，还必须为模块配置多项式长度和数据长度。

16位多项式长度：

```
CRCCON1bits.PLEN = 0b1111;
```

16位数据长度：

```
CRCCON1bits.DLEN = 0b1111;
```

> **注：** PIC16F1613 器件的程序存储器的数据长度实际上为14位。此处选择16位长度的目的是针对CRC计算器和软件CRC验证校验值。这种更改会向每个程序存储器字追加两个零以构成16位字。更多信息，请参见例1。

2.2 配置存储器扫描器

由于CRC是对安全要求极高的任务，因此将在突发模式下使用存储器扫描器。这意味着，CRC运行时将暂停所有CPU功能。在8 MHz Fosc频率下，在整个存储器面板（2000个程序字）上运行CRC意味着将使CPU暂停4.086 ms。

突发模式：

```
SCANCON0bits.MODE = 0b01;
```

如果4.086 ms过长，则可为扫描器设置一个附加位，用于在中断期间暂停扫描器。这意味着，仍可以执行对安全要求极高的中断。

可选的中断服务：

```
SCANCON0bits.INTM = 1;
```

配置存储器扫描器的最后一步是设置起始地址和最终地址。SCANLADRH:L寄存器对保存存储器扫描器的起始地址。SCANHADRH:L寄存器对保存要扫描的最终地址。扫描操作期间，SCANLADR寄存器对将递增以显示正在获取的当前地址。在例1中，将使用最后两个存储器地址来存储计算出的校验值。

第一个地址：

SCANLADRH = 0x00;

SCANLADRL = 0xFD;

最后一个地址：

SCANHADRH = 0x07;

SCANHADRL = 0xFB;

例1： CRC外设代码

```
uint16_t HWCRC (uint16_t lastAddress)
{
    uint16_t HWCRCresult;

    CRCACCL = 0xFF; //将种子值设置为0xFFFF
    CRCACCH = 0xFF;
    CRCXORH = 0x80; //使用CRC-16-ANSI 0x8005
    CRCXORL = 0x05;
    CRCCON1bits.DLEN = 15; //使用16位数据长度以匹配软件CRC
                           //将高2位视为0。
    CRCCON1bits.PLEN = 15; //使用最长17位多项式（-2）
    CRCCON0bits.ACCM = 1; //使能填充零
    CRCCON0bits.SHIFTM = 0; //先移入最高有效位（正常）
    SCANCON0bits.MODE = 0b01; //使能"突发模式"，
                              //以停止全部其他任务的执行，直到CRC完成
    SCANLADRH = 0x00; //设置存储器扫描的第一个地址
    SCANLADRL = 0x00;
    SCANHADRH = lastAddress >> 8; //设置存储器扫描的最后一个地址
    SCANHADRL = lastAddress;

    SCANCON0bits.EN = 1;
    CRCCON0bits.EN = 1;
    CRCCON0bits.CRCGO = 1; //使能CRC
    SCANCON0bits.SCANGO = 1; //使能扫描以开始CRC
    //这应暂停CPU的执行，
    //直到扫描器完成扫描并且最后一个存储单元位于CRC中

    while(CRCCON0bits.BUSY);

    HWCRCresult = ((CRCACCH<<8) | CRCACCL);
    return HWCRCresult;
}
```


© 2015 Microchip Technology Inc.

2.3 运行CRC

CRC和扫描器现在均已配置为在所需存储器区域运行CRC-16-ANSI算法。现在可使能CRC和扫描器模块。

首先将CRCGO位置1以开始CRC，然后将SCANGO位置1。此时，CPU将暂停正常代码执行，因为扫描已设置为突发模式（出现之前所述的可能中断时除外）。

CPU将关闭约4.086 ms。当CRC操作完成时，硬件会将CRCCON0的BUSY位清零。此位清零后，最终校验值将保存到CRCACCH:L寄存器对中。

2.4 CRC外设时序

CRC外设的时序取决于数据宽度、Fosc以及正在扫描的地址数。

如果使用16位数据宽度，则每测试一个程序存储器字，CRC便需要4个指令周期。要测试闪存的2046个地址，CRC需要2046 * 4个指令周期来完成，另加上一些用于调用和返回的其他指令。

在本应用笔记中，CRC在8 MHz频率下需要8211个指令周期，花费的时间刚好超过4 ms。此时序由器件上的信号测量定时器（Signal Measurement Timer，SMT）决定。

有关如何使用SMT获取CRC或任意其他外设的时序的更多信息，请参见**附录B：“SMT时序”**。

3.0 CRC计算器

本应用笔记包括一个易于使用的CRC计算器，此计算器支持多个多项式以及CRC外设的所有功能。此工具将用于验证由CRC外设和B类库完成的所有CRC计算结果。

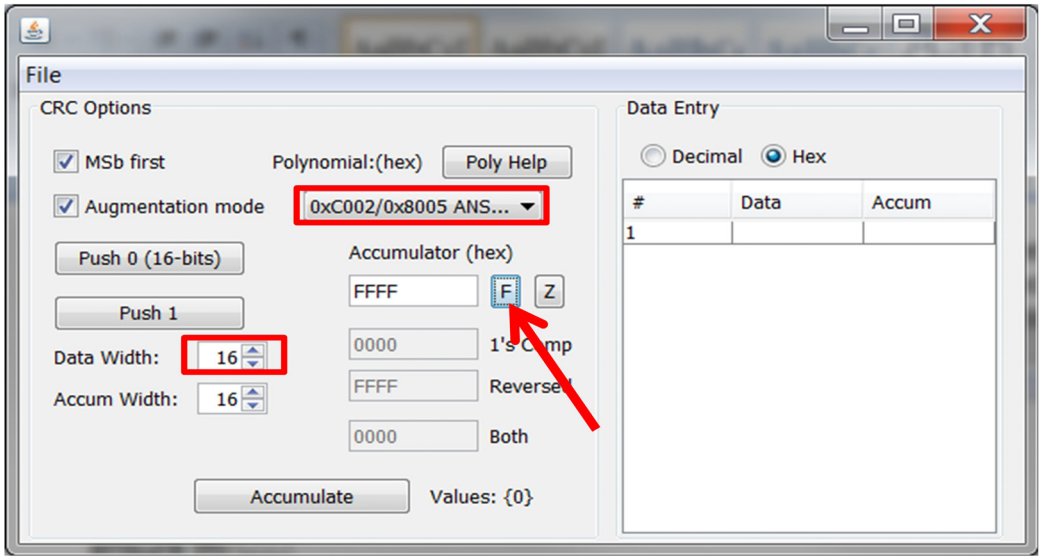
3.1 功能

CRC计算器具有CRC外设的所有功能。其中包括：可供选择的多个多项式，关闭增强模式的选项，在先移入最高有效位与先移入最低有效位之间切换，以及可选的数据宽度和累加器宽度。CRC计算器有两种输入数据的方法。第一种是将数据手动输入右侧的 Data（数据）列。第二种是导入用行分隔的十六进制值文件，类似于可从MPLAB X IDE获取的文件（如**第3.3节“使用MPLAB® X IDE填充表数据”**所述）。

3.2 CRC-16-ANSI的设置

要为CRC-16-ANSI设置CRC计算器，必须将计算器中的一些选项更改为非默认值。首先，从下拉多项式选择框中选择0x8005多项式。然后按下累加器下方的F，这会将初始值设置为0xFFFF。最后，必须将Data Width（数据宽度）增大至16。其余默认选项可保持不变（见图3）。

图3： CRC-16-ANSI的CRC计算器设置



3.3 使用MPLAB® X IDE填充表数据

可使用MPLAB X IDE中的程序存储器视图快速填充CRC计算器的数据表。为此，首先在MPLAB X IDE中打开程序存储器视图，如图4所示。使用复制快捷键（Windows系统中为CTRL+C）复制所需的操作码，如图5所示。操作码将以标准行分隔格式复制到剪切板，之后可粘贴至标准文本文件中（见图6）。接下来，保存文件并打开CRC计算器，然后转至*file -> import file*（文件 -> 导入文件）（见图7）。选择保存的文件，此时数据表将自动填入来自MPLAB X IDE项目的操作码。

© 2015 Microchip Technology Inc.

图4： 程序存储器视图

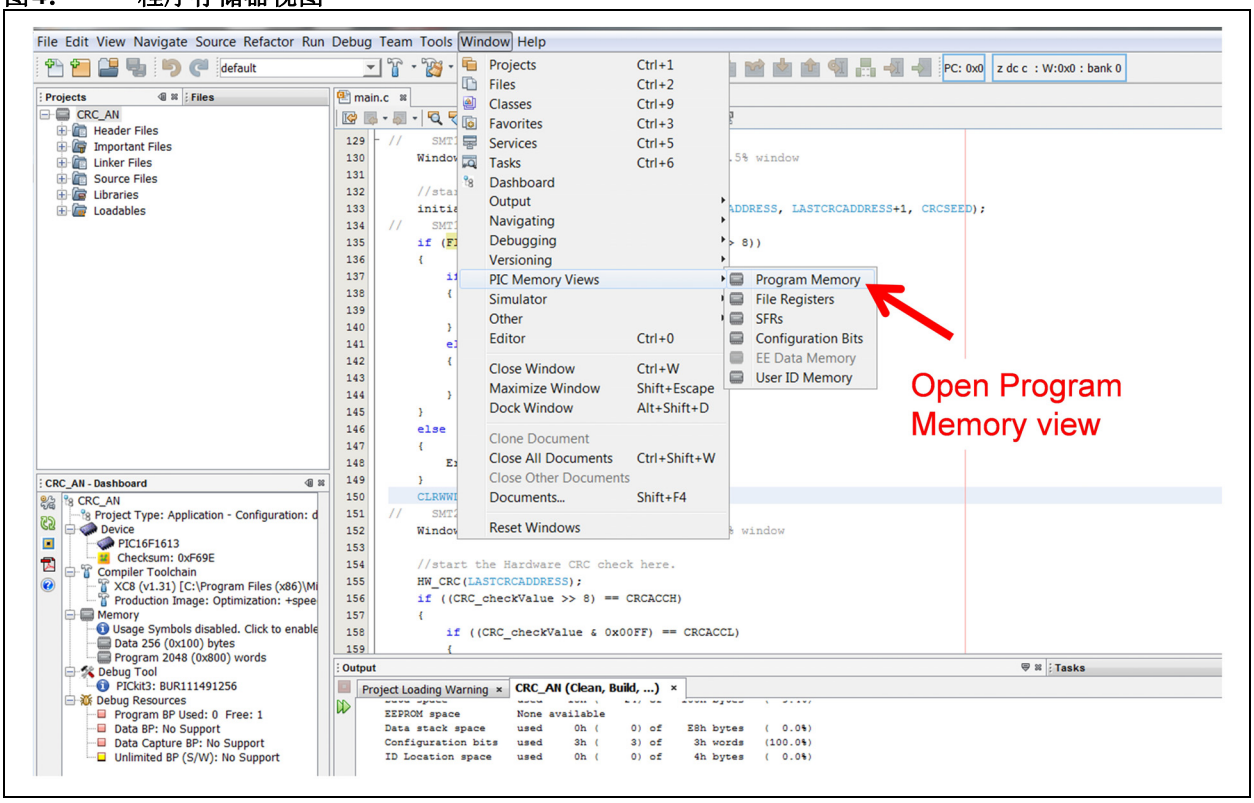


图5： 从程序存储器复制操作码

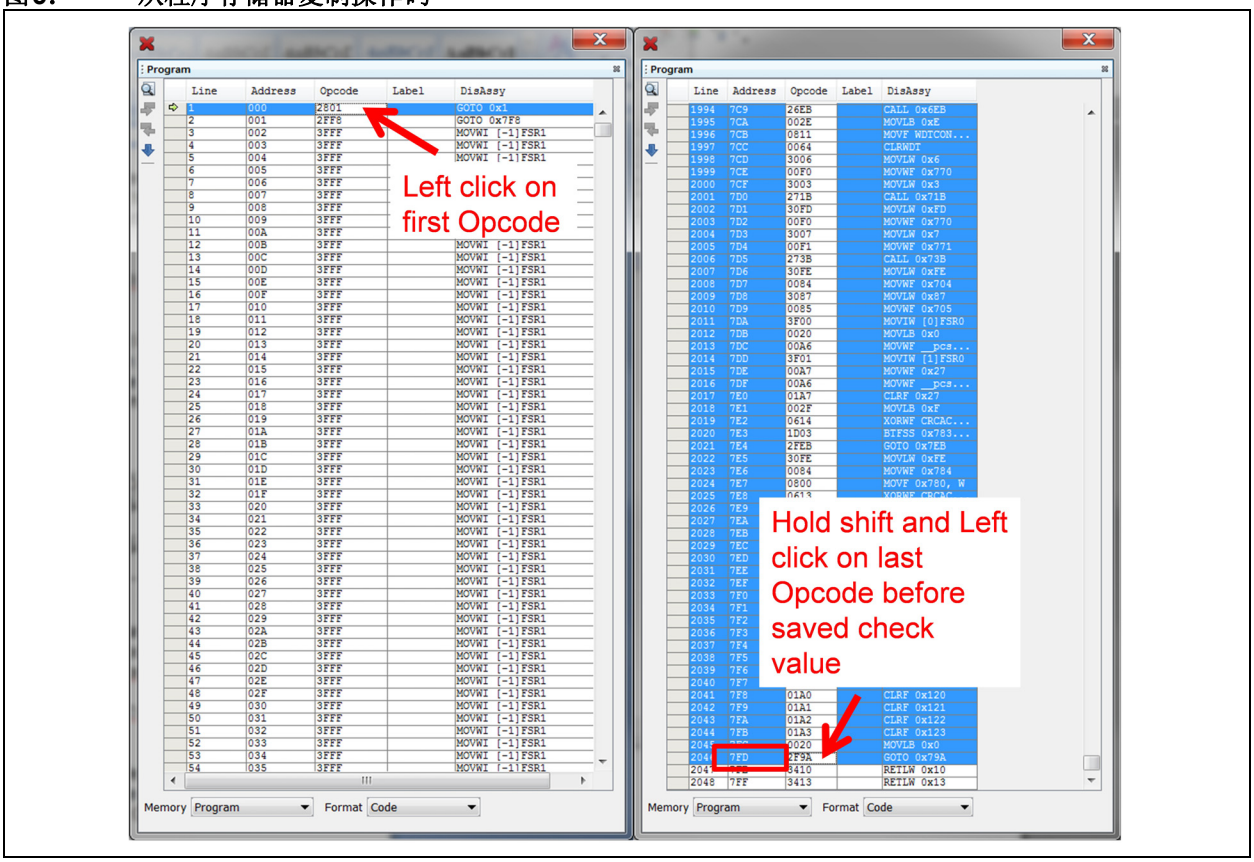


图6： 操作码列表

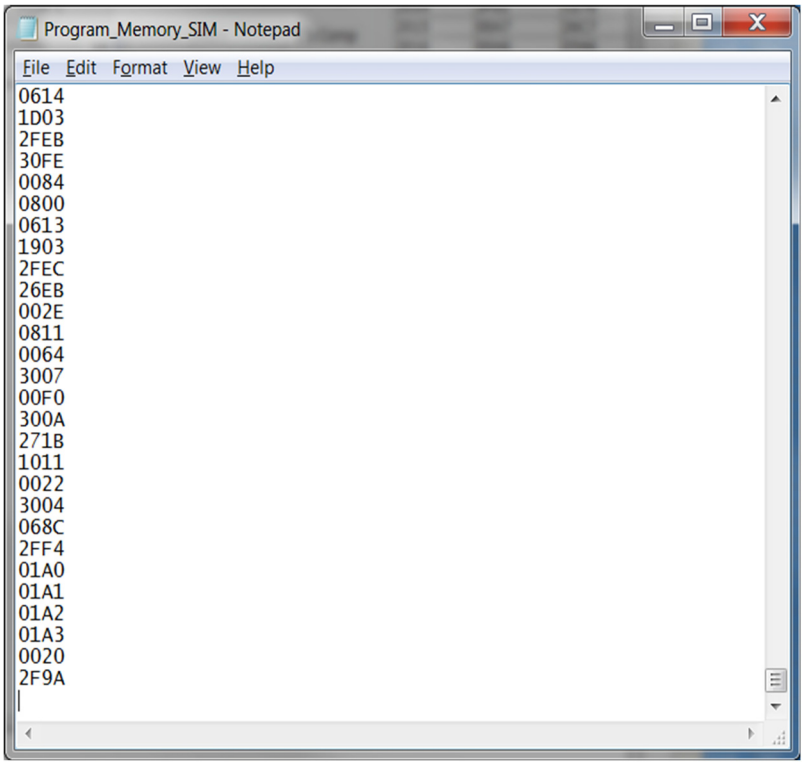
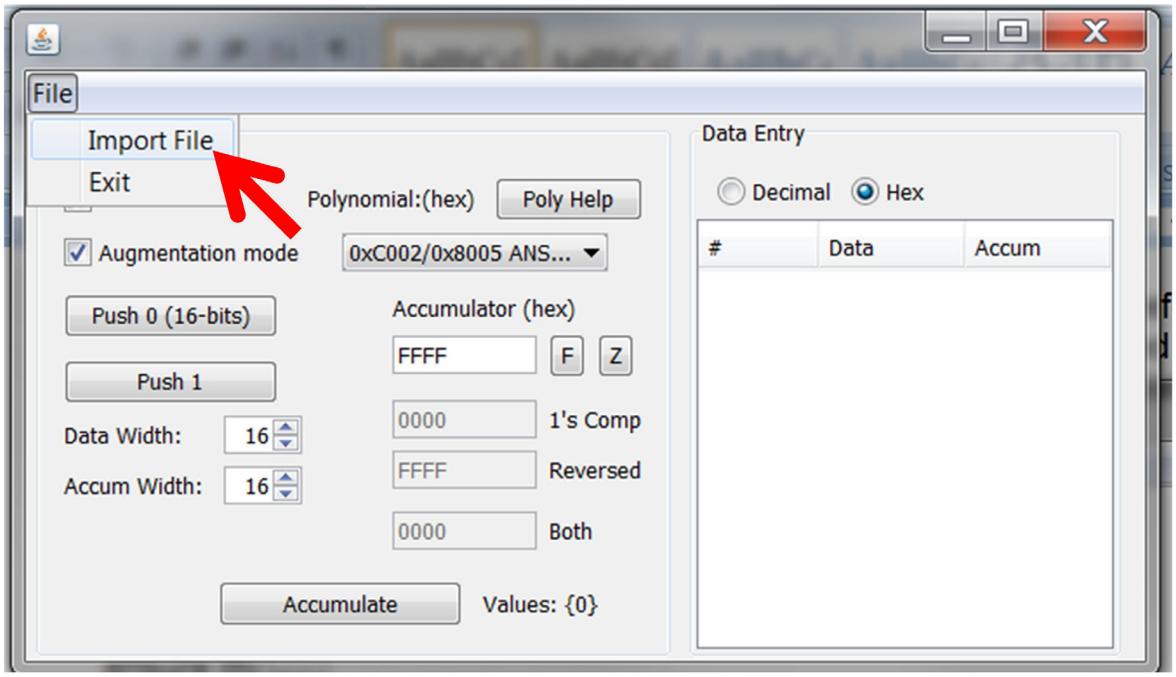


图7： 导入文件

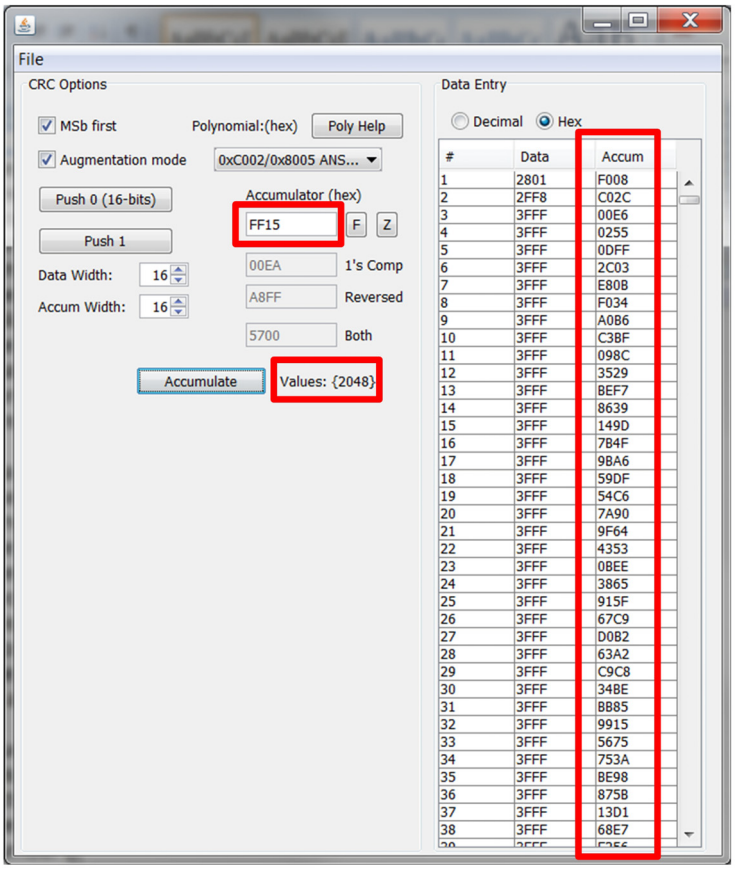


© 2015 Microchip Technology Inc.

3.4 计算校验值

在数据表中填入所需信息后，单击*Accumulate*（累加）按钮可在累加器文本框中生成最终校验值。Values（值）对话框中将指示使用CRC计算的总字数。每次向CRC中输入一个数据字，数据输入部分的Accum（累加）列都会显示累加器值。这对于调试来说非常有用（见图8）。

图8：　使用CRC计算器进行累加



4.0 使用B类库CRC函数

本节将介绍如何实现库中的闪存程序存储器CRC函数。

B类安全软件库随附软件实现的CRC函数，用于测试闪存程序存储器和EEPROM。

与CRC外设不同，软件CRC受限于CRC-16-ANSI算法。

有关使用B类库的更多信息，请参见DS00001799。

4.1 API

要对整个闪存运行CRC测试，必须调用如下函数：

`CLASSB_CRCFlashTest(…);`

此函数有3个参数：要测试的第一个地址、测试的长度以及CRC算法的种子值。

为匹配CRC外设，这些参数的值如下：

- `uint16_t FlashAddress = 0x00`
- `uint16_t Flashlength = 0x07FC`
- `uint16_t crcSeed = 0xFFFF`

> **注：** 此处的长度包含第一个地址。测试将返回16位校验值。更多信息，请参见例2。

例2： CRC库代码

```
uint16_t flashAddress = 0x00;
uint16_t flashLength = 0x07FC;
uint16_t crcSeed = 0xFFFF;
uint16_t CRC-libraryResult;

CRC_libraryResult = CLASSB_CRCFlashTest(myAddress,length,crcSeed);
```

4.2 软件CRC时序

对于相同大小的存储空间，相比于CRC外设所需的8172个指令周期，软件CRC需要216,956个指令周期才能完成。这等于108.478 ms。

© 2015 Microchip Technology Inc.

5.0 使用校验值进行错误校验

使用CRC计算应用程序，可将参考校验值编程到器件中。之后，可针对通过CRC外设或B类库CRC函数计算出的校验值周期性地校验此参考值。如果两个值匹配，则可在很大程度上确定程序存储器中不存在故障。如果两个值不匹配，则意味着存储器已损坏，此时应采取相应措施来确保器件的安全。

如果由于程序存储器中存储的校准值或其他运行时值而无法使用参考校验值，则可以（在存储器值发生变化后）运行两次CRC并比较这两个校验值。为此，可将之前的校验值存储在闪存程序存储器（利用PIC® MCU的自写功能）或者静态RAM地址单元中。

由于CRC算法的性质，无法确定实际错误在程序存储器中的确切位置。根据具体的应用，可执行以下几个步骤。

- 可将错误标志置1，暂停操作
- 可使器件保持复位状态
- 可使器件进入无限循环，暂停所有其他操作
- 可发出错误信号来通知用户出现错误

5.1 参考校验值

使用`const`变量编程时，可将使用CRC计算应用程序（见**第3.0节“CRC计算器”**）确定的参考校验值直接存储到程序存储器中。

例如，如果确定参考校验值为0x123，则可以使用如下代码来存储该值：

```
const uint16_t CRC_checkValue @ 0x7FE =
0x1234
```

此代码会将0x3434和0x3412分别存入地址单元0x7FE和0x7FF。最高有效位包含0x34，因为这是`RETLW`指令的操作码。如表1所示。

表1： 程序存储器中的校验值

行	地址	操作码	DisAssy
2047	7FE	3434	`RETLW` 0x34
2048	7FF	3412	`RETLW` 0x12

之后，可将参考校验值与通过库或外设CRC测得的校验值进行比较（见例3和例4）。

例3：　　与库测得的校验值进行比较

```
if (CRC_checkValue == CRC_libraryResult)
{
    //不执行任何操作，校验值匹配
}
else
{
    ErrorMode();
}
```

例4：　　与外设测得的校验值进行比较

```
if ((CRC_checkValue >> 8) == CRCACCH)
{
    if ((CRC_checkValue & 0x00FF) == CRCACCL)
    {
       //不执行任何操作，校验值匹配
    }
    else
    {
       ErrorMode();
    }
}
else
{
     ErrorMode();
}
```


© 2015 Microchip Technology Inc.

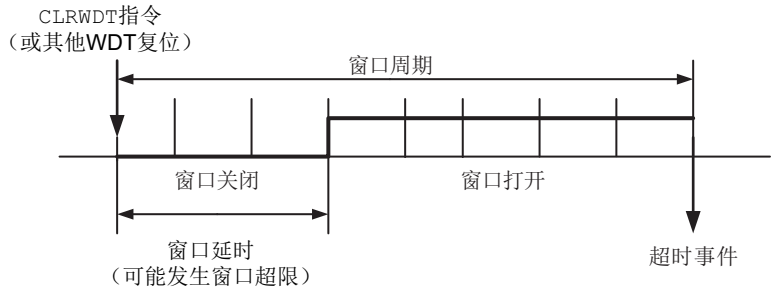
附录A： 窗口化看门狗定时器

窗口化看门狗定时器是PIC16F161X系列产品新增的模块，与典型看门狗相比，它更适合时隙监视。窗口化看门狗定时器可检测过慢和过快的时钟速率。如果程序在窗口之前或者未触发CLRWDT指令的情况下尝试将定时器清零，则会发生复位。为使用此功能，需要精确的时序以确保过程正常运行。CRC外设便是这样一种可提供精确时序的模块（见**第2.4节“CRC外设时序”**）。

必须针对参考时钟（本例中为LFINTOSC）的容差调整窗口时序。LFINTOSC随温度和电压变化的精度规定为+-15%。这意味着窗口必须增加保护频带以覆盖此振荡器频率范围。

例如，在8 MHz频率下，要测试PIC16F1613的整个存储器范围（2K字），CRC外设将在突发模式下花费4 ms的时间。这意味着，应向看门狗定时器添加一个窗口周期，以便在CRC模块提前结束、错误调用或者程序计数器出现问题时报错。这可通过将WDT周期设置为8 ms加上窗口的87.5%来实现。扩展窗口将针对LFINTOSC的容差提供足够的保护频带，同时仍能提供额外的错误覆盖率。

图A-1： 看门狗定时器窗口



附录B： SMT时序

可使用信号测量定时器（SMT）获取PIC16F161X系列器件上的外设或其他软件事件的准确时序。SMT外设对于速率较慢的应用特别有用，因为它是一个24位定时器，而不是16位定时器Timer 1。

有关如何设置SMT以使用Fosc/4时钟进行时序操作的信息，请参见例B-1。

例B-1： SMT初始化

```
SMT2CON0 = 0b10100000;     //使能SMT，上升沿，预分频比1:1，
                           //计数器将在PR处暂停
SMT2CLKbits.CSEL = 0b001; //Fosc/4
SMTxPR = 0xFFFFFF;
```

将PR寄存器的值设置为最大值，确保可将SMT的整个24位范围用于时序用途。如果将STP位（bit 5）置1，则计数器将在SMTxPR值处暂停。这样无需使用溢出即可将事件的时序限制在SMT的24位范围内。

按上述方式初始化SMT后（见例B-1），可使用SMTXCON1寄存器的SMTXGO位来使能和禁止SMT。要获取任意事件的时序信息，只需在事件之前将SMTXGO位置1，然后在事件之后将SMTXGO位立即清零。时间将存储在SMTXTMRU:H:L寄存器中。该值将以指令周期（Fosc/4）数为单位。具体过程如例B-2中的CRC示例代码所示。

例B-2： SMT时序

```
SMT2CON1bits.SMT2GO = 1; //开始SMT时序
//此时启动硬件CRC校验。
HW_CRC(LASTCRCADDRESS);
SMT2CON1bits.SMT2GO = 0; //完成SMT时序
```


© 2015 Microchip Technology Inc.

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 愿与那些注重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。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

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。在 Microchip 知识产权保护下，不得暗中或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、 dsPIC、 FlashFlex、 flexPWR、 JukeBlox、 KEELOQ、 KEELOQ 徽标、 Kleer、 LANCheck、 MediaLB、 MOST、 MOST 徽标、 MPLAB、 OptoLyzer、 PIC、 PICSTART、 PIC32 徽标、 RightTouch、 SpyNIC、 SST、 SST 徽标、 SuperFlash 及 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

The Embedded Control Solutions Company 和 mTouch 为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、BodyCom、chipKIT、chipKIT 徽标、 CodeGuard、 dsPICDEM、 dsPICDEM.net、 ECAN、 In-Circuit Serial Programming、 ICSP、 Inter-Chip Connectivity、 KleerNet、 KleerNet 徽标、 MiWi、 MPASM、 MPF、 MPLAB Certified 徽标、 MPLIB、 MPLINK、 MultiTRAK、 NetDetach、 Omniscient Code Generation、 PICDEM、 PICDEM.net、 PICkit、 PICtail、 RightTouch 徽标、 REAL ICE、 SQI、 Serial Quad I/O、 Total Endurance、 TSHARC、 USBCheck、 VariSense、 ViewSpan、 WiperLock、 Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 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 为 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。

© 2015, Microchip Technology Inc. 版权所有。

ISBN：978-1-63276-931-2

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

═ 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 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和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。

全球销售及服务网点

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技术支持：
http://www.microchip.com/support
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110

加拿大多伦多 Toronto
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港特别行政区
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-3019-1500

日本 Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

日本 Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Dusseldorf
Tel: 49-2129-3766400

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Pforzheim
Tel: 49-7231-424750

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Venice
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

波兰 Poland - Warsaw
Tel: 48-22-3325737

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14

© 2015 Microchip Technology Inc.