

高速ADC SPI控制软件

作者：高速转换器部

特性

- 简单的用户界面支持所有具备SPI功能的标准高速ADC器件
- 简化高速ADC的评估
- 自动生成最终使用C源代码
- 自动生成用于PIC12F629的汇编代码
- 配合高速ADC评估板使用
- 允许访问所有用户控制特性
- 可与VisualAnalog、ADC Analyzer和ADC数据捕捉板一起使用
- 标准USB端口接口
- 内置网络更新特性
- 配置文件可从www.analog.com获得

设备要求

- 运行Windows 98(第二版)、Windows 2000、Windows ME或Windows XP的PC
- 建议使用USB 2.0端口(兼容USB 1.)
- 数据捕捉板(HSC-ADC-EVALB-SC、HSC-ADC-EVALB-DC或HSC-ADC-EVALC)
- 兼容高速ADC评估板

概述

高速ADC SPI程序第二版和第三版(SPIController.exe)允许用户对具备SPI功能的高速模数转换器(ADC)的高级特性进行控制。该高级控制程序与HSC-ADC-EVALB或HSC-ADC-EVALC数据捕捉板和特定器件评估板一起使用,以操纵并控制那些可通过SPI接口访问的特性。有关可控特性的更多信息,请参阅特定器件的数据手册和应用笔记AN-877:“通过SPI与高速ADC接口”。

该SPI控制器程序可以作为独立应用程序运行于Windows®平台上,或者与任何现有数据分析工具一起使用,如VisualAnalog™或ADC Analyzer™等。

该软件所包括的高级特性可确保从评估到样机制造及量产的平稳过渡。获得所需的配置之后,可利用其中的C代码生成器生成兼容的C源代码,以便嵌入定制开发的软件当中。对于无微控制器可用的用户,该软件还包括一个汇编代码生成器,可产生适用于超低成本PIC12F629微控制器的代码。

简化功能框图

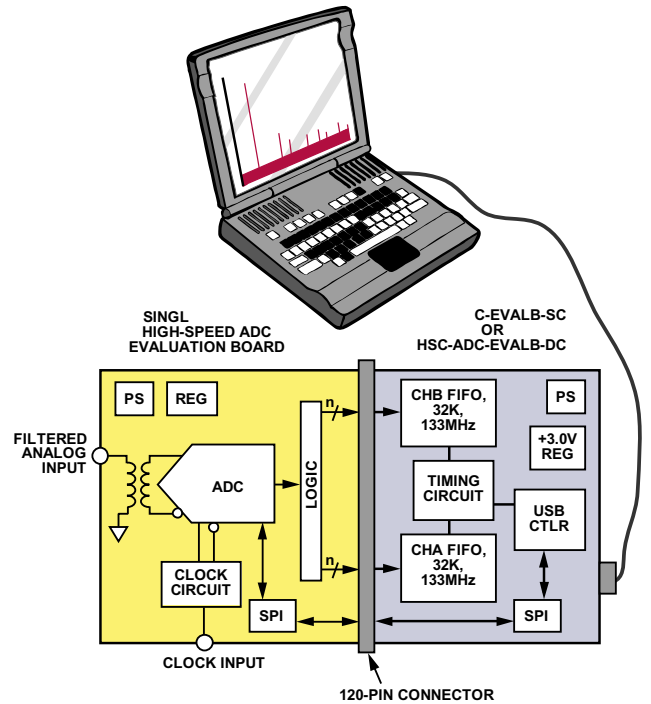


图1.

关于利用此器件控制SPI器件的更多信息,请参考应用笔记AN-812:“基于微控制器的串行端口接口(SPI)启动电路”。

产品聚焦

1. 图形界面。通过直观的用户界面操纵所有片内寄存器。
2. 简单的硬件接口。轻松连接ADC数据捕捉板和兼容USB接口。
3. 内置代码生成器。自动生成用于寄存器配置的C代码。
4. PIC12F629代码生成器。自动生成用于PIC12F629的代码。
5. 网络更新特性。此特性可确保用户随时获得最新的程序文件和配置文件。

有关使用和连接SPI器件的更多信息,请参考应用笔记AN-877和AN-812。

目录

特性.....	1	Enable Auto Channel Update (启用通道自动更新)	8
设备要求.....	1	Perform New DUT function during NewCal (加载新校准文件时执行新DUT功能)	8
概述.....	1	Ignore Transfer function during write (写操作期间忽略传输功能)	8
简化功能框图	1	Read Channel Index (读取通道索引)	8
产品聚焦.....	1	Error Messages (错误消息)	8
修订历史.....	2	USB error msg en (启用USB错误消息)	8
快速开始.....	3	Transfer error msg en (启用传输错误消息)	8
安装程序.....	3	Parallel Port (并行端口)	8
运行程序.....	3	Script Dialog (脚本对话框)	9
省电模式.....	3	生成的典型C源代码	9
开始使用.....	4	Launch Page ReadBack (启动页回读)	13
器件检查.....	5	Launch ASM Code Generator (启动汇编代码生成器)	13
SPI控制面板.....	6	典型PIC12F629汇编代码	14
File (文件) 菜单.....	7	Launch Cal Comparator (启动校准比较器)	17
Cfg Open (打开配置文件)	7	Help (帮助) 菜单.....	18
Cfg Save As (配置文件另存为)	7	Current Files (当前文件)	18
Cal Open (打开校准文件)	7	Device Cfg Filename (器件配置文件名)	18
Cal Save As (校准文件另存为)	7	Device Cal Filename (器件校准文件名)	18
Download Files from FTP Site (从FTP站点下载文件) ..	7	Script Filename (脚本文件名)	18
Exit (退出)	7	Script Log Filename (脚本日志文件名)	18
Configuration (配置) 菜单	8	ASM Filename (汇编文件名)	18
Controller Dialog (SPI Controller Cfg Dialog) [控制器对话框 (SPI控制器配置对话框)]	8	Rev History (修订历史)	18
FIFO Chip Select Number (USB FIFO Channel Number) [FIFO片选号码 (USB FIFO通道号码)]	8	USB Info (USB信息)	18
Program Control (程序控制)	8	Help Topics (帮助主题)	18
Enable Auto Update of DLL, Cfg., Cal files (启用DLL、Cfg、Cal文件的自动更新)	8	SPI Controller User Manual (SPI控制器用户手册)	18
Update Controller From DUT on New DUT (对新DUT应用DUT的控制信息)	8	ADI SPI Protocol User Manual (ADI SPI协议用户手册)	18
		工具栏	19

更换图6.....	6
更换图7和图8	7
更换图9.....	8

2006年3月—版本0：初始版

修订历史

2007年4月—初始版至修订版A

格式更新.....	通篇
将“高速ADC控制器快速开始设置(SPI)”部分的名称更改为“快速开始”	3
更换图2	4
更改“运行程序”部分	3
更换“器件检查”部分	5

快速开始

SPI控制器程序(SPIController.exe)允许访问支持SPI的高速转换器。运行此程序将打开一个窗口，通过该窗口可以完全控制转换器的可用功能。有关可用功能和编程的详细信息，请参阅应用笔记AN-877：“通过SPI与高速ADC接口”。支持的特性随器件不同而异，请参考具体器件的数据手册。虽然本应用笔记介绍了SPI控制器程序的特性，但并未详细描述SPI。

安装程序

虽然本程序无需连接硬件即可运行，但要充分发挥作用，必须使用HSC-ADC-EVALB或HSC-ADC-EVALC板。这些数据捕捉板包括SPI控制器，并需要与支持SPI的高速转换器评估板连接。请参阅具体器件的数据手册以确定是否兼容。

安装SPI控制器程序的方法有以下几种：

- SPI控制器程序可以作为ADC Analyzer第4.9.0版或更高版本的一部分进行安装。此程序在数据捕捉板随附的CD上提供，也可以从www.analog.com/fifo下载。插入光盘或运行下载文件即可安装这两个应用程序。
- 可以从www.analog.com/fifo下载单独的SPI控制器安装程序。此安装程序不包括ADC Analyzer。请下载zip文件并解压缩到一个临时目录，然后运行setup.exe应用程序，按照屏幕说明操作。
- 单独的安装程序(最新版本)可从以下地址获得：
ftp.analog.com/pub/adispi/A2DComponents/Install。

完成软件安装后，请遵循以下步骤：

1. Windows必须注册驱动程序。为此，请设置HSC-ADC-EVALB-xx或HSC-ADC-EVALC数据评估板。虽然此时并无必要连接ADC评估板，但如果有评估板可用，也可在此将评估板安装好。
2. 接通HSC-ADC-EVALB-xx或HSC-ADC-EVALC板和ADC评估板(如有连接)的电源。
3. 将USB电缆连接在计算机与HSC-ADC-EVALB-xx或HSC-ADC-EVALC板之间。Windows将识别该评估板并注册驱动程序。驱动程序注册完成之后，一旦Windows指示新硬件已准备就绪，即可使用该软件。

运行程序

运行程序的方法有两种：

- 从开始菜单，选择Analog Devices下的SPIController。

- 双击桌面上的SPIController快捷方式。


软件将执行检查，确保硬件存在，并且有一个SPI兼容器件连接到评估板。如果硬件不存在和/或没有连接正确的器件，可能会显示一系列警告消息。虽然提供了警告，但程序将继续运行。

首次运行程序时，显示若干警告之后，它会提示用户要求配置文件。文件浏览器显示可用的配置文件。如果没有显示可用的配置文件，请将浏览器设置到配置文件所在硬盘上的存储位置。这通常是安装目录。更多配置文件可以从www.analog.com的产品页面下载，或者随同评估板提供。最新的配置文件也可以通过网络更新而获得。请参见本应用笔记的“Download Files from FTP Site(从FTP站点下载文件)”部分。如果您仍然无法找到所需的配置文件，请发送电子邮件至highspeed.converters@analog.com，索取特定器件的最新SPI配置文件。

如果程序不是首次运行，它将使用最后一次选择的配置文件。要更改配置文件，请打开File(文件)菜单，选择Cfg Open(打开配置文件)。选择与所选器件相对应的文件。上电时，器件以寄存器的默认值运行。通常有一个典型状态，确保器件正常工作。由于每种转换器都支持一组不同的特性，因此应参阅器件的数据手册。所有器件都至少支持模式设置中的省电特性。

省电模式

按以下步骤操作可将器件置于省电模式。

1. 确保选择“启用异步传输”  选项。这将确保每次更改控制时，相关更改将立即通过SPI发送到器件。
2. 选择ADCBase、ADCBase0或ADCGlobal 0选项卡，以显示适当的页面并找到模式寄存器(0x08h)。然后，在Int Pwr Dn窗口下，点击Full Pwr Dn单选按钮。这将把相应的值写入模式寄存器，使器件关断。如果有电流表，可以用它观察此时的器件状态。
3. 点击Chip Run(芯片运行)，器件将再次启动。

点击相应的按钮、复选框或文本框，可以探索器件的其它特性。只要选择了“启用异步传输”，命令就会被立即发送到器件。如果没有启用，可以通过点击Transfer(传输)按钮手动发送命令。有关Transfer按钮的说明，请参见表2。

开始使用

本应用笔记的“快速开始”部分简要说明了如何安装和运行SPIController软件以及如何使器件进入省电模式，后续部分将详细说明如何使用SPI控制器软件。

首次启动SPI控制器程序时，它会搜索所需的硬件和配置文件。如果无法找到这些文件，则会显示警告消息。

1. 对于每条警告消息，点击**OK**(确定)予以响应，直到程序提示要求配置文件。配置文件决定如何根据所用的器件配置和选择用户界面。

2. 选择与当前连接到基准设备的器件评估板相关的配置文件，然后点击**Open**(打开)。所有ADC都遵循同样的SPI协议和存储器映射，选择错误的配置文件不会损坏器件，实际上仍可能进行一定程度的控制。然而，使用正确的配置文件可确保给定器件的所有可用特性都能使用。

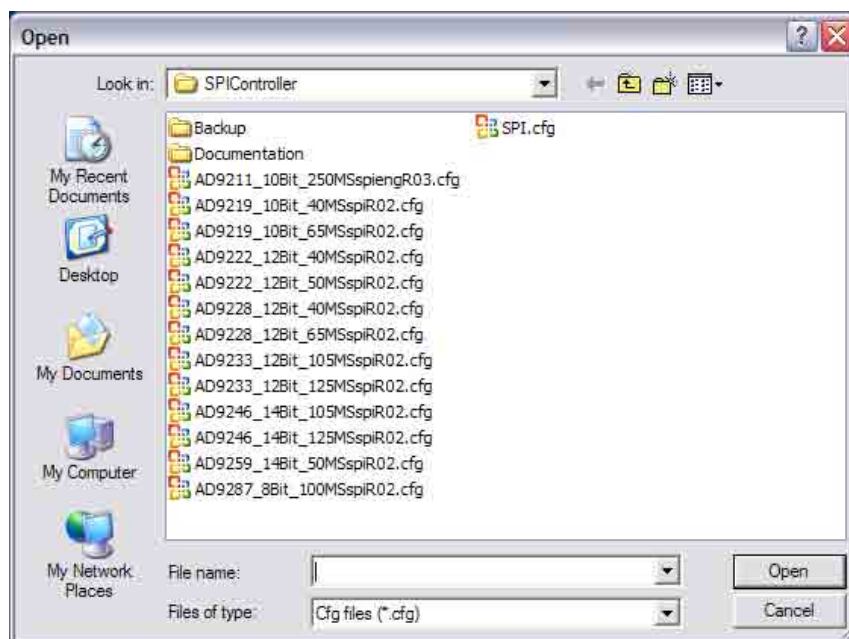


图2. 文件浏览

器件检查

选择适当的配置文件后，即开始测试评估板状态。同时还会检查评估板上安装的器件。报告错误时，点击**OK**(确定)响应警告消息之后，软件继续运行。

软件首先检查是否有HSC-ADC-EVALB-xx或HSC-ADC-EVALC数据评估板。如果此板未连接，将显示图3所示的错误消息，表明发生读取测试故障。

如果电缆连接正确，用户可以点击**Ignore**(忽略，见图4)以结束启动过程，否则启动过程将重复一次，以使用户连接电缆。如果错误是因片选分配错误而引起，可以在软件完成启动过程之后更改芯片选择。

找到数据捕捉板之后，软件将进而确定哪一个ADC与板相连。如果没有选择正确的配置文件，或者连接的板不正确，则将显示一条错误消息，其中会列出要求的设备名称。

点击**OK**(确定)表示程序应在无硬件的情况下继续运行。



图3. 典型USB错误消息



图4. 芯片ID错误消息



图5. 芯片ID错误消息

SPI控制面板

SPI控制面板提供图形界面，以使用户与连接到SPI接口的器件交互。由于各个器件可能具有不同的一组寄存器，因此用户界面因器件而异。有关各种特性的详细信息，请参阅应用笔记AN-877：“通过SPI与高速ADC接口”。欲确定各器件有哪些特性可用，请参考器件数据手册。本应用笔记旨在概要介绍该软件的功能，而无意阐述器件各种可用功能的意义。

SPI控制器用户界面上的每个选项卡都含有一组不同的控件。**Global**(全局)选项卡包含的特性适用于所有SPI器件，并且局限于物理接口配置的特性，如SPI接口、LSB优先或芯片软复位等。所有控件的设置表示写入芯片的状态。

常见ADC特性位于**ADCBase 0**页面，可通过选择**ADCBase 0**选项卡进行访问。

如果器件具有多个ADC，则各ADC通道的局部特性显示在**ADC A至ADC Z**选项卡上。对于单通道ADC，只有**ADCBase 0**选项卡可用。并非所有转换器都支持所有特性，因此，应参阅特定器件的数据手册以确定哪些特性可用，以及器件将如何响应所写入的值。

辅助ADC功能位于附加页面上，可通过选择具有相应标签的选项卡进行访问。如果需要不只一个页面，则将包括多个页面。这些是器件特定的功能，相关说明请参考具体器件的数据手册。

有些转换器含有本应用笔记未说明的额外页面。同样，欲了解有关这些特性的信息，请参考具体器件的数据手册。一般而言，这些特性的功能与这里所述的特性是一致的。

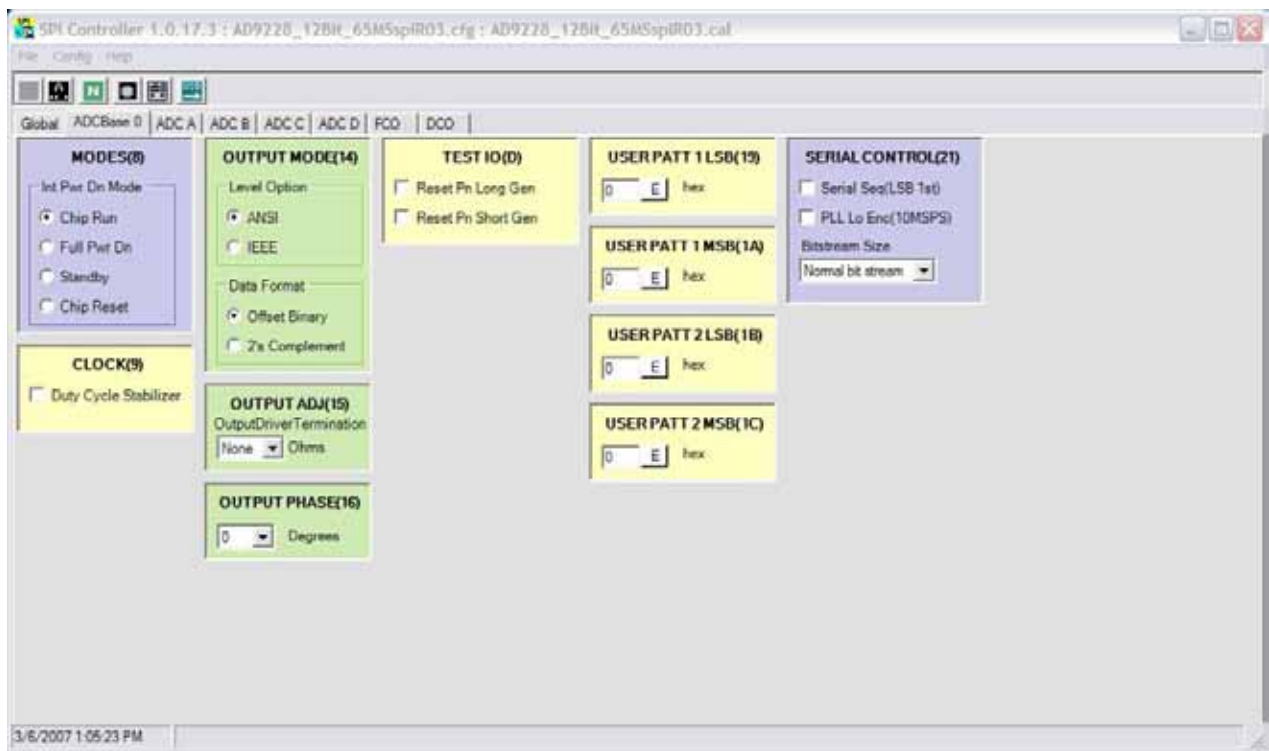


图6. 典型用户界面，外观可能因器件而异

File(文件)菜单

文件菜单允许加载和保存各种配置文件，如图7所示。

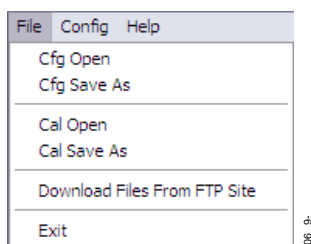


图7. 文件菜单

Cfg Open(打开配置文件)

SPI控制器可以配合大多数支持SPI端口的AD92xx、AD94xx和AD66xx数据转换器工作(详情见器件数据手册)。要针对特定器件配置控制程序，请选择**Cfg Open**，然后选择该器件的配置文件。

配置文件随同本软件或评估板提供，也可以从网站(www.analog.com)下载。还可以通过网络获得更新，请参见本应用笔记的“Download Files from FTP Site(从FTP站点下载文件)部分”所述。如果通过以上途径均无法找到所需的配置文件，请联系highspeed.converters@analog.com索取。

Cfg Save As(配置文件另存为)

配置文件发生更改时，可以通过选择此选项以新名称予以保存，以供将来使用。程序会提示输入新的文件名。

Cal Open(打开校准文件)

Cal Open允许用户检索设置信息。通过此选项可检索先前保存的寄存器设置，以便将器件迅速设置为所需的状态。

Cal Save As(校准文件另存为)

当器件的寄存器处于所需状态时，可以将寄存器设置保存为一个文件，以供将来使用。选择**Cal Save As**并在提示时提供新的文件名，可以完成该操作。

Download Files from FTP Site(从FTP站点下载文件)

此选项允许从ADI公司的公共FTP站点直接下载新的配置文件。这是确保所用的配置文件和可执行文件为最新版本的最便捷、最简单方法。同样，如果您的评估板没有配置文件，这是您第一个应该查找的地方。

点击此选项时，程序会提示您将从ADI公司公共FTP站点复制数据，需要时您可以取消复制。

如果您选择**Yes(是)**，程序将保存当前配置，退出程序，执行更新程序，然后返回程序并重新加载上一次配置。另一方面，您也可以从www.analog.com的产品站点手动下载配置文件，或者联系highspeed.converters@analog.com以索取特定器件的最新SPI配置文件。

Exit(退出)

点击Exit可关闭SPI控制器程序。

Configuration(配置)菜单

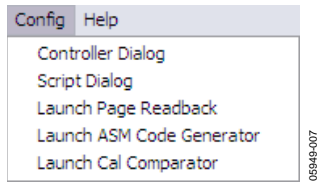


图8. Config菜单

Controller Dialog (SPI Controller Cfg Dialog) [控制器对话框(SPI控制器配置对话框)]

此选项允许选择器件片选设置、各种程序流程控制选项、错误处理方法(见图9)及其它接口选项。一般不需要更改这些选择,因为它们是在配置器件的过程中设置的。不过,对于高级用户,更改选择的功能是有用的。



图9. SPI控制器配置对话框

FIFO Chip Select Number (USB FIFO Channel Number) [FIFO片选号码(USB FIFO通道号码)]

FIFO Chip Select Number菜单允许您配置SPI控制器以激活所需转换器的片选线。虽然大多数评估板使用Chip Select Number 1,但此选项允许您根据需要选择其它片选号码(Chip Select Number)。程序运行时默认选择适当的值,不需要更改。

Program Control(程序控制)

Program Control提供了多种可选特性。要启用一项特性,请选择该特性左边的复选框。

Enable Auto Update of DLL, Cfg., Cal files(启用DLL、Cfg、Cal文件的自动更新)

选择此特性时,程序将在每次运行时搜索ADI公司公共FTP站点有无最新的程序和配置文件。此选项要求接入Internet,只有当Internet连接可用时才能选择。选择File菜单下的Download Files from FTP Site,可以手动检索更新。

Update Controller From DUT on New DUT(对新DUT应用DUT的控制信息)

选择此选项时,一旦按下Update Controller From DUT on New DUT按钮,片内寄存器设置就被传输到控制程序的图形用户界面(GUI)。如果未勾选此复选框,按下Update Controller From DUT on New DUT按钮时,程序将用GUI中的值对DUT编程。DUT指待测器件。

Enable Auto Channel Update(启用通道自动更新)

对于多通过ADC,勾选此复选框将使多个通道应用类似的特性更新。

Perform New DUT function during NewCal (加载新校准文件时执行新DUT功能)

如果勾选此复选框,则当加载新配置文件时,GUI将执行新DUT功能。

Ignore Transfer function during write(写操作期间忽略传输功能)

并非所有ADC都要求在执行一个或一组命令后执行一个传输命令。如果勾选此复选框,则将禁止程序自动产生传输命令。

Read Channel Index(读取通道索引)

Read Channel Index下拉框中的值决定回读周期中访问哪一个ADC通道。

Error Messages(错误消息)

USB error msg en(启用USB错误消息)

出现USB错误消息时,允许其显示。如果未勾选此复选框,则将禁止消息显示。

Transfer error msg en(启用传输错误消息)

出现传输错误消息时,允许其显示。如果未勾选此复选框,则将禁止消息显示。

Parallel Port(并行端口)

并行端口用于USB SPI控制器可能不支持的器件。目前没有器件使用此接口。

Script Dialog (脚本对话框)

此选项允许启用日志、伪代码（C代码）和汇编（PIC12F629）文件的创建功能（见图10）。

日志文件是此特性启用期间向SPI端口发送的命令记录，与记录宏会话非常相似。勾选或不勾选此复选框将启用或禁用此选项。日志文件是一种历史记录，通过它可以回顾发送了哪些命令。在点击Clear（清除）之前，记录将持续加载到此文件中。

脚本文件包括C代码文件，代表传输至器件SPI端口的数据序列。此文件可以用作编程参考，或者包括在最终应用程序源代码中。其格式遵循标准C/C++语法。

如果该软件是最终应用程序的一部分，则用户必须提供硬件特定的读取和写入工具。此特性有助于用户了解如何对器件编程。下面的示例代码显示了写入脚本文件的典型编程操作序列。这些文件带有注释，描述了设置文件的顺序。

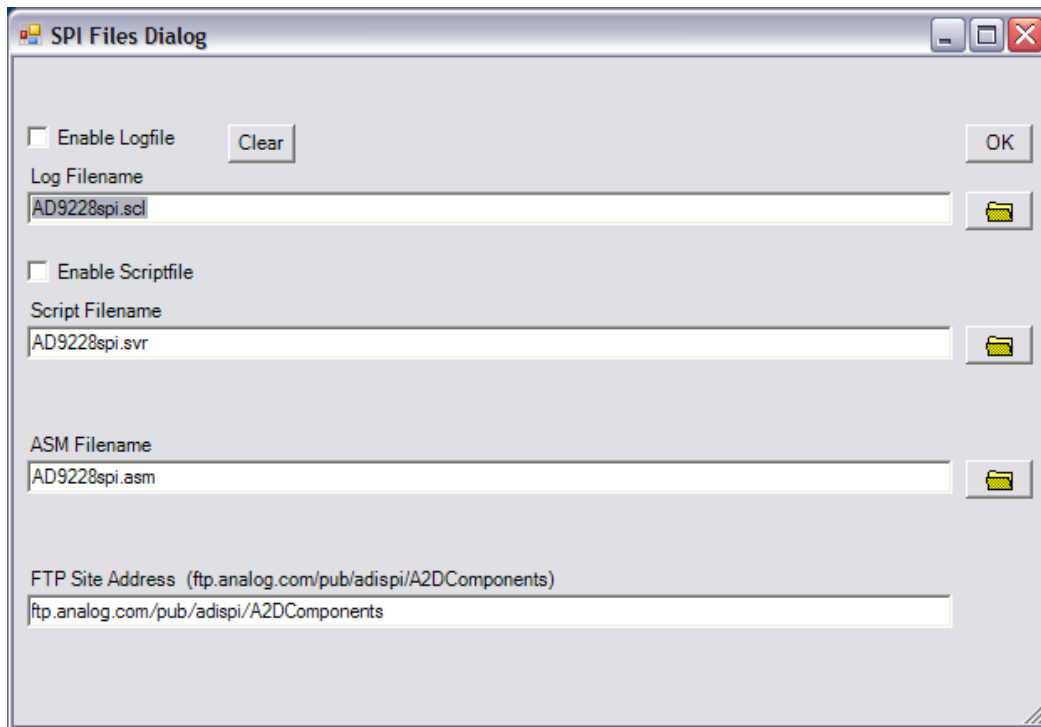


图10. SPI文件对话框

生成的典型C源代码

```
//The following code exemplifies usage of the generated script or log file.
// declarations for wrapper functions *****
// void SPIwrite(int address, int value);
// int SPIread(int address);
// definitions of wrapper functions *****
// void SPIwrite(int address, int value)
// {
//     place compiler specific 'write' function here, i.e.
//     void write(int address, int value);
// }

// int SPIread(int address)
```

AN-878

```
// {
//     place compiler specific 'read' function here, i.e.
//     int read(int address);
// }

// sample pgm *****
// void main()
//     {
//         int x;
//         SPIwrite(22,4);
//         x = SPIread(22);
//     }

// *****
SPIwrite(0000,18); //chip_port_config 2/7/2006 10:40:11 PM
SPIwrite(0005,0F); //device_index_2 2/7/2006 10:40:11 PM
SPIwrite(00FF,00); //device_update 2/7/2006 10:40:11 PM
//ADC A
SPIwrite(0008,00); //modes 2/7/2006 10:40:11 PM
SPIwrite(0009,01); //clock 2/7/2006 10:40:11 PM
SPIwrite(000D,00); //test_io 2/7/2006 10:40:11 PM
SPIwrite(000F,60); //adc_input 2/7/2006 10:40:11 PM
SPIwrite(0014,00); //output_mode 2/7/2006 10:40:11 PM
SPIwrite(0015,00); //output_adjust 2/7/2006 10:40:11 PM
SPIwrite(0016,03); //output_phase 2/7/2006 10:40:11 PM
SPIwrite(0018,03); //vref 2/7/2006 10:40:11 PM
SPIwrite(0019,00); //user_patt1_lsb 2/7/2006 10:40:11 PM
SPIwrite(001A,00); //user_patt1_msb 2/7/2006 10:40:11 PM
SPIwrite(001B,00); //user_patt2_lsb 2/7/2006 10:40:11 PM
SPIwrite(001C,00); //user_patt2_msb 2/7/2006 10:40:11 PM
SPIwrite(0021,00); //serial_control 2/7/2006 10:40:11 PM
SPIwrite(0022,00); //serial_ch_stat 2/7/2006 10:40:11 PM
//ADC B
SPIwrite(0008,00); //modes 2/7/2006 10:40:11 PM
SPIwrite(0009,01); //clock 2/7/2006 10:40:11 PM
SPIwrite(000D,00); //test_io 2/7/2006 10:40:11 PM
SPIwrite(000F,60); //adc_input 2/7/2006 10:40:11 PM
SPIwrite(0014,00); //output_mode 2/7/2006 10:40:11 PM
```

```
SPIwrite(0015,00); //output_adjust 2/7/2006 10:40:11 PM
SPIwrite(0016,03); //output_phase 2/7/2006 10:40:11 PM
SPIwrite(0018,03); //vref 2/7/2006 10:40:11 PM
SPIwrite(0019,00); //user_patt1_lsb 2/7/2006 10:40:11 PM
SPIwrite(001A,00); //user_patt1_msb 2/7/2006 10:40:11 PM
SPIwrite(001B,00); //user_patt2_lsb 2/7/2006 10:40:11 PM
SPIwrite(001C,00); //user_patt2_msb 2/7/2006 10:40:11 PM
SPIwrite(0021,00); //serial_control 2/7/2006 10:40:11 PM
SPIwrite(0022,00); //serial_ch_stat 2/7/2006 10:40:11 PM
//ADC C
SPIwrite(0008,00); //modes 2/7/2006 10:40:11 PM
SPIwrite(0009,01); //clock 2/7/2006 10:40:11 PM
SPIwrite(000D,00); //test_io 2/7/2006 10:40:11 PM
SPIwrite(000F,60); //adc_input 2/7/2006 10:40:11 PM
SPIwrite(0014,00); //output_mode 2/7/2006 10:40:11 PM
SPIwrite(0015,00); //output_adjust 2/7/2006 10:40:11 PM
SPIwrite(0016,03); //output_phase 2/7/2006 10:40:11 PM
SPIwrite(0018,03); //vref 2/7/2006 10:40:11 PM
SPIwrite(0019,00); //user_patt1_lsb 2/7/2006 10:40:11 PM
SPIwrite(001A,00); //user_patt1_msb 2/7/2006 10:40:11 PM
SPIwrite(001B,00); //user_patt2_lsb 2/7/2006 10:40:11 PM
SPIwrite(001C,00); //user_patt2_msb 2/7/2006 10:40:11 PM
SPIwrite(0021,00); //serial_control 2/7/2006 10:40:11 PM
SPIwrite(0022,00); //serial_ch_stat 2/7/2006 10:40:11 PM
//ADC D
SPIwrite(0008,00); //modes 2/7/2006 10:40:11 PM
SPIwrite(0009,01); //clock 2/7/2006 10:40:11 PM
SPIwrite(000D,00); //test_io 2/7/2006 10:40:11 PM
SPIwrite(000F,60); //adc_input 2/7/2006 10:40:11 PM
SPIwrite(0014,00); //output_mode 2/7/2006 10:40:11 PM
SPIwrite(0015,00); //output_adjust 2/7/2006 10:40:11 PM
SPIwrite(0016,03); //output_phase 2/7/2006 10:40:11 PM
SPIwrite(0018,03); //vref 2/7/2006 10:40:11 PM
SPIwrite(0019,00); //user_patt1_lsb 2/7/2006 10:40:11 PM
SPIwrite(001A,00); //user_patt1_msb 2/7/2006 10:40:11 PM
SPIwrite(001B,00); //user_patt2_lsb 2/7/2006 10:40:11 PM
SPIwrite(001C,00); //user_patt2_msb 2/7/2006 10:40:11 PM
SPIwrite(0021,00); //serial_control 2/7/2006 10:40:11 PM
```

AN-878

```
SPIwrite(0022,00); //serial_ch_stat 2/7/2006 10:40:11 PM
//AUX 0
SPIwrite(0022,00); //serial_ch_stat 2/7/2006 10:40:11 PM
//AUX 1
SPIwrite(0022,00); //serial_ch_stat 2/7/2006 10:40:11 PM
SPIwrite(00FF,01); // write transfer bit
0 = SPIread(00FF); // read transfer bit
```

ASM Filename(ASM文件名)设置用于创建PIC12F629汇编代码的文件名称。有关如何生成这些文件的详情,请参见“Launch ASM Code Generator(启动汇编代码生成器)”部分。更多信息请参考应用笔记AN-812:“基于微控制器的串行端口接口(SPI®)启动电路”。

FTP Site Address(FTP站点地址)是更新软件用以加载最新可执行文件和配置文件的地址。虽然一般不应更改此地址,但此选项允许您进行必要的更改,例如将来站点移动时。

Launch Page ReadBack(启动页回读)

Page ReadBack(页回读)特性允许读取相连器件的片内存储器(见图11)。

点击**Read(读取)**按钮将启动回读和比较过程。显示器将在单独的窗口中显示结果,主控窗口可继续独立工作。这样就很容易更改值,然后验证是否正确接收到数据。

回读页显示寄存器所在的选项卡、寄存器描述、地址以及写入和读取值。为清楚起见,还会写入一个通过/失败标志,此特性有利于调试器件。

此特性与主界面相结合,允许用户写入并探索完整的用户存储器映射。

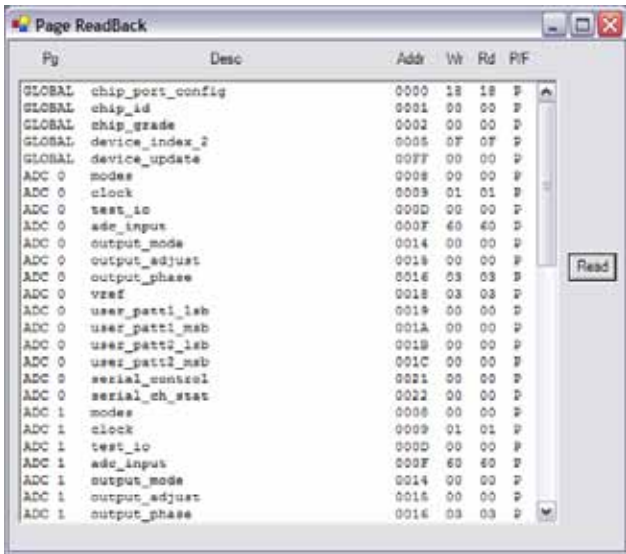





图11. 页回读对话框

Launch ASM Code Generator(启动汇编代码生成器)

ASM Code Generator(ASM代码生成器)配合评估板上的**Microchip® PIC12F629**使用(见图12)。可以将此电路复制并放置于最终用户板上,以提供极低成本的**SPI**控制器。代码生成器是一种用于收集配置数据的简单方法。所收集的数据可以用**PIC**编译器进行汇编,并写入器件中,以初始化所需的条件。更多信息请参考应用笔记AN-812。

ASM Code Generator对话框上提供了三个按钮来帮助收集配置数据,如表1所示。

表1. ASM代码生成器按钮

按钮	描述
	按下Record(记录)按钮即开始记录键击动作。记录期间,当控件改变时,其值记录在ASM Code Generator中,直到记录停止。再次按下此按钮时,它变为Stop Record(停止记录)按钮。
	按下Record(记录)按钮即开始记录键击动作。记录期间,当控件改变时,其值记录在ASM Code Generator中,直到记录停止。再次按下此按钮时,它变为Stop Record(停止记录)按钮。
	按下Record(记录)按钮即开始记录键击动作。记录期间,当控件改变时,其值记录在ASM Code Generator中,直到记录停止。再次按下此按钮时,它变为Stop Record(停止记录)按钮。

ASM Code Generator对话框(见图12)显示了使能索引寄存器5中的所有四个ADC、禁用占空比稳定器以及启动芯片的命令序列。

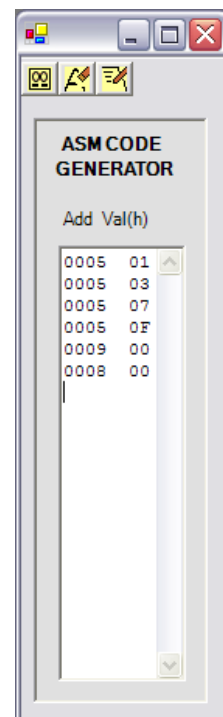


图12. ASM Code Generator对话框

AN-878

典型PIC12F629汇编代码

```
; This is the SPI_control ASM file used to configure the PIC12F629 for the SPI boot circuit
; Code for the PIC12F629 SPI boot circuit
#define _version '0.10'
;Update History:
;Application Description/Comments
;Author: Steve Ranta and Alfredo Barriga
;Hardware Notes
    list p=12F629 ;Device Specification
    #include <p12F629.inc> ;Include files/registers
;Variable/registers declarations
;Macros (Configuration parameters>      Internal RC oscillator and Watchdog timer disabled
    __CONFIG _CP_OFF & _CPD_OFF & _BODEN_OFF & _WDT_OFF & _MCLRE_ON & _WDT_OFF &
    _PWRTE_ON & _INTRC_OSC_NOCLKOUT ;Parameters for the configuration register
;Define Port Variables and pin assignment
#define SCLK GPIO,0 ;defines clock pin in GPIO,0 (SPI clock)
#define CSB GPIO,2 ;defines CSB pin in GPIO,2 (Chip enable signal)
#define SDO GPIO,1 ;defines SDO pin in GPIO,1 (SPI data in)
#define LED GPIO,4 ;defines LED pin in GPIO,4 (LED indicator for busy state)
;*****Variable Definitions*****
;Variables are stored in these general purpose registers
upperaddress      EQU 0x20
loweraddress      EQU 0x21
SPIData           EQU 0x22
bitcounter        EQU 0x23
byte              EQU 0x24
ORG 0x000
    ;calibrate internal oscillator with factory settings
;Initialize all internal peripherals and ports for SPI functions
bcf STATUS,RP0 ;Select Bank 0
clrf GPIO ;Initializes GPIO register
movlw 07h ;Setting up value for comparator module <CM2,CM1,CM0>=<111>
movwf CMCON ;comparator module off(digital out)
bsf STATUS,RP0 ;select Bank 1
movlw b'00001000' ;literal to set all port pins as output except for GPIO.3 (reset)
movwf TRISIO ;direction bits moved to TRISIO
bcf STATUS,RP0 ;clear register bank select
```

```

;*****
;Initializes outputs
INIT
    bsf CSB; sets CSB
    bcf SCLK;clears SCLK
    bcf SDO; clears SDO
    bcf LED;clears LED (LED is off)
    clrf bitcounter ;clear bitcounter register
    clrf byte; clear byte register

;Since we are performing writing operations, and the SPI mode is MSB first, 16-bit
addressing
;the first register value, which is the instruction register will always be h'00'
;-----
-----
    movlw h'00'
    movwf upperaddress
;*****
;*****
;This section of the program is generated by SPIController, and will contain the register
;address and data values the user has specified.
movlw h'0005'
movwf loweraddress
movlw h'01'
movwf SPIdata
call write_SPI
movlw h'0005'
movwf loweraddress
movlw h'03'
movwf SPIdata
call write_SPI
movlw h'0005'
movwf loweraddress
movlw h'07'
movwf SPIdata
call write_SPI
movlw h'0005'
movwf loweraddress
movlw h'0F'
movwf SPIdata

```

AN-878

```
call write_SPI
movlw h'0009'
movwf loweraddress
movlw h'00'
movwf SPIdata
call write_SPI
movlw h'0008'
movwf loweraddress
movlw h'00'
movwf SPIdata
call write_SPI
;*****
;*****
;This is the last part of the program. Here, an UPDATE DEVICE command is executed to
transfer
;data from the buffers to the SPI port.
;Sleep command is executed after all registers have been updated, and the device goes to
sleep
;until reset or power on.
;Update register: FFh, value:h'01'
    movlw h'FF'
    movwf loweraddress
    movlw h'01'
    movwf SPIdata
    call write_SPI
    sleep
;*****
write_SPI ;routine that writes data to the SPI port
    bsf LED ;LED turns ON indicating writing process has started
    bcf CSB; CSB pin goes low to start a writing command

;Write_upperaddress
    movf upperaddress,W ;transfer upperaddress to W register
    movwf byte
    call send_byte ;call send_byte subroutine
;Write_loweraddress
    movf loweraddress,W
    movwf byte
    call send_byte; write lower address into SPI
```



```

;write_data
    movf SPIdata,W
    movwf byte
    call send_byte
    bsf CSB;CSB is set to end SPI writing operation
    bcf LED; Led turns off
    bcf SDO; clears sdo
    retlw 0 ;Write_spi subroutine ends.
;*****
send_byte ;subroutine to send byte to spi port
    movlw 8; w=8
    movwf bitcounter;bitcounter=8
loop
    bcf SCLK; clk set low
    RLF byte,F ;rotate each bit to the left through carry in STATUS register
    btfsc STATUS,C
    goto SDO_HIGH
SDO_LOW
    bcf SDO ;Set SDO low
    goto clk_high
SDO_HIGH
    bsf SDO;Set SDO high
clk_high
    bsf SCLK
    decfsz bitcounter,F
    goto loop
    retlw 0 ;send_byte subroutine ends
;End of program
    sleep
END

```

Launch Cal Comparator(启动校准比较器)

此特性将启动一个窗口，允许对两个不同的校准文件进行比较。这包括以下校准文件：DUT默认文件与当前状态、DUT默认文件与指定文件、当前状态与指定文件。高级用户会发现这项特性十分有用，可比较任何两种设置以查找不同之处。

Help(帮助)菜单

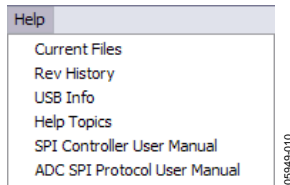


图13. Help菜单

Current Files(当前文件)

Current Files菜单提供程序各部分所用重要文件名称的汇总(见图14)。使用Config菜单可以更改这些文件名(见“Script Dialog(脚本对话框)”部分)。可以更改的文件包括配置文件、校准文件、脚本文件和对话框文件。



图14. SPI文件对话框

Device Cfg Filename(器件配置文件名)

此文件用于配置程序以支持器件，包含器件的寄存器和位信息。寄存器和位信息可以从该软件包获得、从ADI网站(www.analog.com)下载或者通过FTP下载特性获得(见“Download Files from FTP Site(从FTP站点下载文件)”部分)。如果安装CD没有提供转换器的配置文件，请联系highspeed.converters@analog.com。

Device Cal Filename(器件校准文件名)

此文件包含各片内寄存器的默认值。随同软件提供的初始文件包含默认的芯片值。可以创建新校准文件(New Cal)，以便设置不同的值。

Script Filename(脚本文件名)

C代码生成器写入此文件。生成此文件所用的数据即为发送至USB SPI控制器的数据，因此它所代表的信息与用于从评估板对器件进行编程的信息相同。此代码包括嵌入式文档以增强可读能力。虽然代码完全可用，但未经优化。可以利用各种编程技术优化代码，以改进速度并缩减代码。此代码提供一个出发点，确保您在配置器件时有参考代码可资借鉴。

Script Log Filename(脚本日志文件名)

指令记录在此文件中。启用此特性时，传给器件的所有指令同时被写入此文件，形成一个命令日志。它是一个附加文件，包含器件的编程历史(直到文件被清除)。调试时，可通过此文件确定发送给器件的哪些命令是否发挥作用。

ASM Filename(汇编文件名)

选择PIC12F629汇编代码生成选项后，汇编代码写入此文件。有关此文件的更多信息，请参阅应用笔记AN-812。

Rev History(修订历史)

此选项提供该软件的详细历史信息。

USB Info(USB信息)

此选项提供USB驱动程序的详细信息。

Help Topics(帮助主题)

此选项提供有关部分程序特性的指南。

SPI Controller User Manual(SPI控制器用户手册)

此选项打开安装在PC上的此应用笔记的PDF版本。

ADI SPI Protocol User Manual(ADI SPI协议用户手册)

此选项打开应用笔记AN-877：“通过SPI与高速ADC接口”。








工具栏

表2说明了SPI控制器工具栏上的按钮（见图15）。并非所有器件在所有模式下都可以使用所有特性。



图15. 工具栏

表2. SPI控制器工具栏

工具	符号	描述
Transfer(传输)		此按钮将程序中的当前值写入片内寄存器。如果启用了脚本文件(C代码生成器)，则同时将写入此文件，但禁止写入日志文件。启动程序时，此按钮禁用并呈灰色，表示异步传输已启用。如果异步传输禁用，则此控件启用，按下  按钮将同时更新所有片内寄存器。
En Async Xfer (启用异步传输)		此按钮切换异步传输和同步传输。此按钮的默认模式为异步传输，意味着当按钮、文本框或其它控件改变时，信息立即传递至DUT。在同步传输模式下，只有按下Transfer按钮后才会做出更改。启用异步模式时，Transfer按钮禁用。如果启用了脚本文件，则禁止写入脚本文件。如果启用了日志文件，则将更新日志文件。
New DUT(新DUT)		此按钮重新初始化DUT接口并向寄存器00h发出器件软启动命令。此外还会读取芯片ID，确保识别正确的器件。如果勾选了 SPI Controller Cfg Dialog (SPI控制器配置对话框)上的Use defaults on New DUT(对新DUT使用默认值)复选框，则使用芯片默认值，而不是软件当前显示的值。
Update DUT (更新DUT)		更新DUT的寄存器，使之与控制软件中的设置相匹配。如果器件进入省电模式而未退出软件，此特性会发挥作用。
Update Controller (更新控制器)		更新SPI控制器程序，使之与DUT中的设置相匹配。如果退出后再启动程序，但器件未进入省电模式，此特性会发挥作用。
USB Refresh (USB刷新)		当USB接口由于掉电或电缆断开连接而中断时，此按钮可重新建立与USB接口的通信。

注释