

高速 ADC SPI コントロール・ソフトウェア

著者：高速コンバータ事業部

特長

SPI 機能付きのすべての標準高速 ADC デバイスを対象とするシンプルなユーザー・インターフェース

高速 ADC の評価を簡素化

最終用途の C ソース・コードを自動生成

PIC12F629 のアセンブル・コードを自動生成

高速 ADC 評価ボードで使用

すべてのユーザー制御機能をアクセス可能

VisualAnalog、ADC Analyzer、ADC データ・キャプチャ・ボードと組み合わせて使用

標準 USB ポート・インターフェースを使用

ウェブ更新機能を内蔵

www.analog.com から設定ファイルを提供

必要な装置

Windows 98 (2nd Ed)、Windows 2000、Windows ME、または Windows XP が動作する PC

USB 2.0 ポートを推奨(USB 1.1 互換)

データ・キャプチャ・ボード(HSC-ADC-EVALB-SC、HSC-ADC-EVALB-DC、または HSC-ADC-EVALC)

互換性のある高速 ADC 評価ボード

概要

高速 ADC SPI プログラムのバージョン 2 とバージョン 3 (SPIController.exe) を使うと、ユーザーは SPI 機能付きの高速 A/D コンバータ(ADC)の高度な機能を制御することができます。この高度なコントローラ・プログラムは、データ・キャプチャ・ボード(HSC-ADC-EVALB または HSC-ADC-EVALC)およびデバイス固有の評価ボードと組み合わせて使い、SPI からアクセス可能な機能の操作と制御を行うことができます。使用可能な広範囲な機能の詳細については、デバイスのデータ・シートとアプリケーション・ノート AN-877 「*Interfacing to High Speed ADCs via SPI*」に記載されています。

この SPI Controller プログラムは、Windows®プラットフォーム上でスタンドアロン・アプリケーションとして、あるいは VisualAnalog™や ADC Analyzer™のような市販のデータ解析ツールと組み合わせて、動作させることができます。

このソフトウェアには、評価からプロトタイプと量産へスムーズに移行することを可能にする高度な機能が含まれています。所望の設定を行った後、ユーザーが開発したソフトウェアに組み込むことが可能な互換 C ソース・コードを提供する C コード・ジェネレータをこのプログラムに追加することができます。市販マイクロコントローラを使用しないユーザーの場合、ソフトウェアに超低価格 PIC12F629 マイクロコントローラをターゲットとするコードを生成するアセンブル・コード・ジェネレータを追加することもできます。

簡略化した機能ブロック図

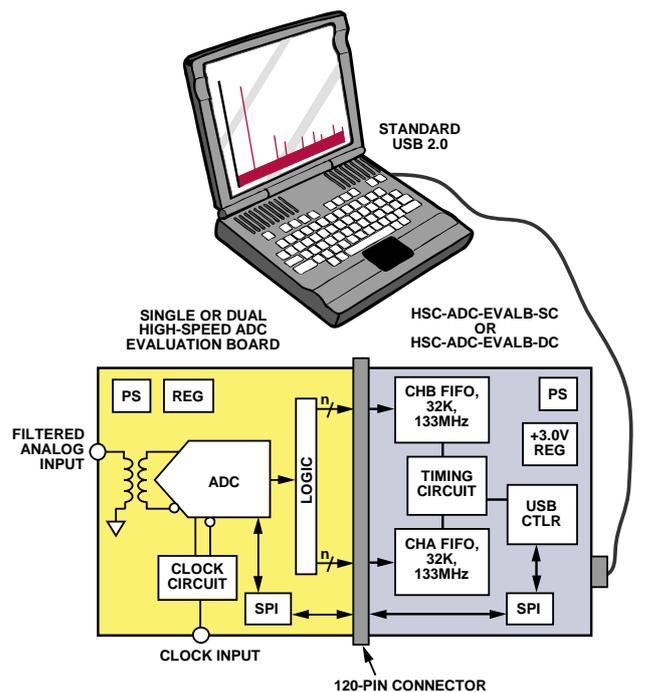


図 1.

このデバイスを使った SPI デバイス制御の詳細については、アプリケーション・ノート AN-812 「*Microcontroller-Based Serial Port Interface (SPI) Boot Circuit*」を参照してください。

製品のハイライト

1. グラフィック・インターフェースを採用。すべての内蔵レジスタを操作する直感的なユーザー・インターフェース。
2. シンプルなハードウェア・インターフェース。ADC データ・キャプチャ・ボードと互換 USB インターフェースとの容易なインターフェース
3. 組み込みコード・ジェネレータを採用。レジスタ設定用 C コードの自動生成。
4. PIC12F629 コード・ジェネレータ。PIC12F629 用コードの自動生成。
5. ウェブ更新機能。この機能により、最新プログラム・ファイルと設定ファイルの提供が常に保証されます。

SPI デバイスの使い方とインターフェースの詳細については、アプリケーション・ノート AN-877 と AN-812 を参照してください。

Rev. A

目次

特長.....	1	Enable Auto Channel Update.....	8
必要な装置.....	1	Perform New DUT function during NewCal.....	8
概要.....	1	Ignore Transfer function during write.....	8
簡略化した機能ブロック図.....	1	Read Channel Index.....	8
製品のハイライト.....	1	エラー・メッセージ.....	8
改訂履歴.....	2	USB error msg en.....	8
クイック・スタート.....	3	Transfer error msg en.....	8
プログラムのインストール.....	3	Parallel Port.....	8
プログラムの実行.....	3	Script ダイアログ.....	9
パワーダウン.....	3	Page ReadBack の起動.....	12
ケッティング・スターテイド.....	4	Launch ASM Code Generator.....	12
デバイスのチェック.....	5	代表的な PIC12F629 ASM コード.....	14
SPI コントロール・パネル.....	6	Launch Cal Comparator.....	17
File メニュー.....	7	Help メニュー.....	18
Cfg Open.....	7	Current Files.....	18
Cfg Save As.....	7	Device Cfg Filename.....	18
Cal Open.....	7	Device Cal Filename.....	18
Cal Save As.....	7	Script Filename.....	18
Download Files from FTP Site.....	7	Script Log Filename.....	18
Exit.....	7	ASM Filename.....	18
Configuration メニュー(設定メニュー).....	8	Rev History.....	18
Controller Dialog (SPI Controller の設定ダイアログ).....	8	USB Info.....	18
FIFO Chip Select Number (USB FIFO チャンネル番号).....	8	Help Topics.....	18
Program Control.....	8	SPI Controller User Manual.....	18
Enable Auto Update of DLL, Cfg., Cal files.....	8	ADI SPI Protocol User Manual.....	18
Update Controller From DUT on New DUT.....	8	Tool バー.....	19

改訂履歴

Updated Format.....	Universal
Changed Name of High Speed ADC Controller Quick Start Set Up (SPI) Section to Quick Start Section.....	3
Replaced Figure 2.....	4
Changes to Running the Program Section.....	3
Replaced the Device Check Section.....	5
Replaced Figure 6.....	6
Replaced Figure 7 and Figure 8.....	7
Replaced Figure 9.....	8

3/06—Revision 0: Initial Version

クイック・スタート

SPI Controller プログラム(SPIController.exe)は、SPI をイネーブルした高速コンバータへのアクセスを提供します。このプログラムを動作させると、コンバータの使用可能な機能を完全に制御できるウィンドウが開きます。使用可能な機能とプログラミングの詳細については、アプリケーション・ノート AN-877「*Interfacing to High Speed ADCs via SPI*」を参照してください。サポートされている機能はデバイスごとに異なるため、個々のデバイスのデータ・シートを参照してください。このアプリケーション・ノートでは SPI Controller プログラムの機能を説明しますが、SPI をドキュメント化するものではありません。

プログラムのインストール

このプログラムはハードウェアを装着しなくても動作しますが、フル機能を動作させるためには HSC-ADC-EVALB ボードまたは HSC-ADC-EVALC ボードが必要です。これらのデータ・キャプチャ・ボードには SPI Controller が含まれているため、SPI をイネーブルした高速コンバータ評価ボードに対するインターフェースが必要です。互換性については、各デバイスのデータ・シートを参照してください。

次のいずれかの方法で SPI Controller をインストールします。

- この SPI Controller プログラムは、ADC Analyzer のバージョン 4.9.0 以降の一部としてインストールすることができます。データ・キャプチャ・ボードと一緒に CD で提供されるこのプログラムは、www.analog.com/fifo からダウンロードすることもできます。ディスクを挿入するか、ダウンロードしたファイルを実行して、両アプリケーションをインストールします。
- www.analog.com/fifo から個別の SPI Controller インストレーション・プログラムも提供しています。このインストレーションには ADC Analyzer は含まれません。zip ファイルをダウンロードし、テンポラリ・ディレクトリへ解凍します。setup.exe アプリケーションを実行し、表示される指示に従ってください。
- 個別のインストーラ(最新バージョン)は、ftp.analog.com/pub/adispi/A2DComponents/Install から提供されています。

ソフトウェアをインストールした後、次のステップに従います。

- Windows にドライバをインストールする必要があります。データ・キャプチャ・ボード HSC-ADC-EVALB-xx または HSC-ADC-EVALC をセットアップします。ADC 評価ボードがない場合には、この時点で接続する必要はありませんが、ここで接続することができます。
- HSC-ADC-EVALB-xx または HSC-ADC-EVALC ボードと ADC 評価ボード(接続した場合)をターンオンします。
- コンピュータと HSC-ADC-EVALB-xx または HSC-ADC-EVALC ボードの間に USB ケーブルを接続します。Windows は評価ボードを認識して、ドライバを登録します。ドライバの登録が完了した後、Windows が新しいハードウェアの使用可能を表示したら、ソフトウェアは使用可能になります。

プログラムの実行

次の 2 つの方法のいずれかでプログラムを実行します。

- スタート・メニューで、**Analog Devices** から **SPIController** を選択します。
- SPIController** デスクトップ・ショートカットをダブル・クリックします。

ソフトウェアは、ハードウェアが存在し、SPI 互換デバイスが評価ボードに接続されていることを確認します。ハードウェアが存在しない、または正しいデバイスが接続されていない場合は、一連の警告メッセージが表示されます。警告が表示されても、プログラムは実行を続けます。

このプログラムを始めて実行したときは、幾つかの警告の後に設定ファイルのプロンプトが表示されます。ファイル・ブラウザが使用可能な設定を表示します。使用可能な設定が表示されない場合は、ブラウザに設定ファイルが格納されているハード・ディスクのロケーションを設定します。これは、通常、インストレーション・ディレクトリです。その他の設定ファイルは www.analog.com の製品ページからダウンロードするか、評価ボードに添付されています。最新の設定ファイルの場合、ウェブ更新も使用可能です。このアプリケーション・ノートの Download Files from FTP Site のセクションを参照してください。所望の設定ファイルを見つけることができない場合は、電子メールで、highspeed.converters@analog.com にデバイスの最新 SPI 設定ファイルを要求してください。

プログラムを前に実行したことがある場合には、そのとき選択した設定が使用されます。設定を変更するときは、File メニューで **Cfg Open** を選択します。選択したデバイスに対応するファイルを選択します。パワーアップ時、デバイスはレジスタ内のデフォルト値で動作します。これらは一般に、デバイスを動作させるための通常の状態です。各コンバータでサポートされている機能が異なるので、デバイスのデータ・シートを参照してください。すべてのデバイスはモード設定で少なくともパワーダウン機能をサポートしています。

パワーダウン

デバイスをパワーダウン・モードにするときは、次のステップに従います。

- 非同期転送をイネーブルした  オプションが選択されていることを確認します。これにより、制御が変更されるごとに、直ちに SPI を経由してデバイスへ送信されるようになります。
- ADCBase**、**ADCBase0**、または **ADCGlobal 0** タブを選択して、該当するページを表示し、モード・レジスタ(0x08h)を探します。次に **Int Pwr Dn** ウィンドウで、**Full Pwr Dn** ラジオボタンをクリックします。これにより対応する値がモード・レジスタに書き込まれて、デバイスはシャットダウンされます。電流計があれば、これを計測することができます。
- Chip Run** クリックして、デバイスを再起動します。

デバイスのその他の機能は、該当するボタン、チェックボックス、またはテキストボックスをクリックして調べることができます。コマンドは、非同期転送がイネーブルされている限り、直ちにデバイスへ送信されます。イネーブルされていない場合は、**Transfer** ボタンをクリックすると、コマンドが手動で発行されます。**Transfer** ボタンについては、表 2 を参照してください。

ケッティング・スターテイド

このアプリケーション・ノートのクイック・スタートのセクションで、SPI Contoller ソフトウェアのインストール/使用方法とデバイスのパワーダウン方法の短い説明を行いました。このアプリケーション・ノートの残りの部分では、SPI Contoller ソフトウェアの使い方について詳しく説明します。

SPI Contoller プログラムを始めて起動すると、必要なハードウェアと設定ファイルの検索を開始します。これらのファイルを見つけることができない場合は警告メッセージが表示されます。

1. プログラムが設定ファイルのプロンプトを表示するまで、各警告メッセージに対して **OK** をクリックして応答します。設定ファイルは、使用するデバイスに応じてユーザー・インターフェースの設定方法と選択方法を指定します。

2. ベンチ装置へ接続中のデバイス評価ボードに対応する設定ファイルを選択して、**Open** をクリックします。すべてのADCは同じSPIプロトコルとメモリ・マップを使うため、誤った設定を選択しても、デバイスに損傷を与えることはなく、実際にある程度の制御を行うことができますが、正しい設定を使って、デバイスのすべての使用可能な機能をアクセスできるようにしてください。

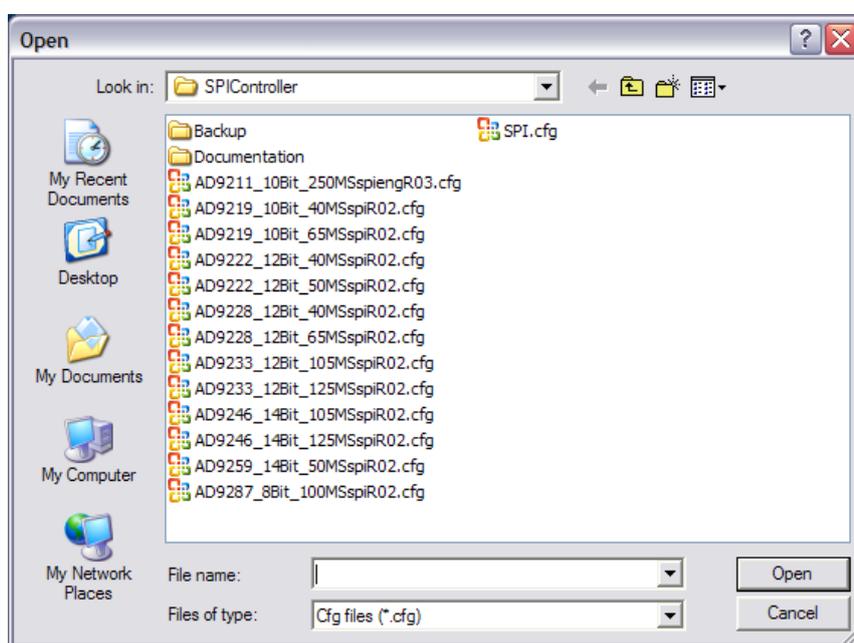


図 2. ファイルのブラウジング

デバイスのチェック

該当する設定ファイルを選択すると、評価ボードのステータスがテストされます。評価ボードにインストールされたデバイスもチェックされます。エラーが報告された場合、警告メッセージに対して **OK** をクリックすると、ソフトウェアは実行を続けます。

ソフトウェアは、先ず使用可能な HSC-ADC-EVALB-xx または HSC-ADC-EVALC データ・キャプチャ・ボードをチェックします。このボードが接続されていない場合、エラー・メッセージ (図 3 参照)が表示され、読み出しテスト・エラーが発生します。

ケーブルが正しく接続されている場合、**Ignore** (図 4 参照)をクリックして、スタートアップ・シーケンスを終わるか、あるいはスタートアップ・シーケンスをもう 1 回繰り返して、ケーブル接続を行います。エラーがチップ・セレクト割り当ての誤りから発生している場合、ソフトウェアがスタートアップ・シーケンスを終了した後チップ・セレクトを変更することができます。

データ・キャプチャ・ボードが見つかり、ソフトウェアはボードに接続されている ADC をチェックします。正しい設定ファイルが選択されていない場合、または正しくないボードが接続されている場合、エラー・メッセージが必要とされるデバイス名リストと一緒に表示されます。

OK をクリックして、プログラムをハードウェアなしで実行することを指示します。



図 3. 代表的な USB エラー・メッセージ

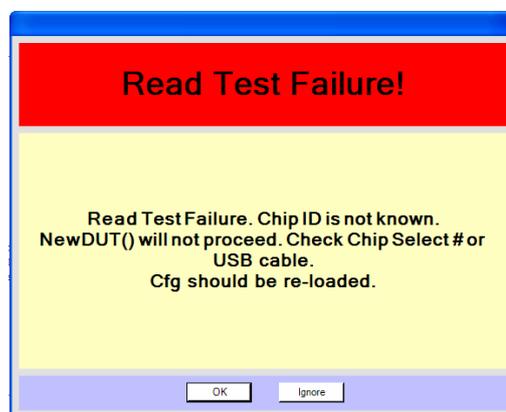


図 4. チップ ID エラー・メッセージ

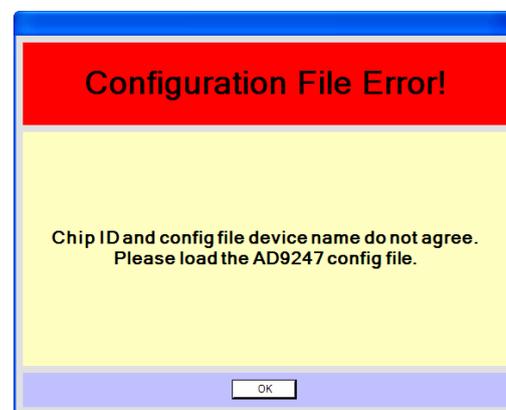


図 5. チップ ID エラー・メッセージ

SPIコントロール・パネル

SPI コントロール・パネルは、SPI インターフェースに接続されたデバイスと交信するグラフィカルな方法を提供します。各デバイスは異なるレジスタ・セットを持っているため、ユーザー・インターフェースはデバイスごとに異なります。各機能の詳細については、アプリケーション・ノート AN-877「*Interfacing to High Speed ADCs via SPI*」を参照してください。各デバイスで使用可能な機能については、デバイスのデータ・シートを参照してください。このアプリケーション・ノートは、ソフトウェア機能の概要を説明するもので、使用可能な各デバイス機能を説明するものではありません。

SPI Controller ユーザー・インターフェースの各タブには様々なセットの制御が含まれています。**Global** タブには、SPI デバイスにグローバルに適用される機能が含まれ、SPI インターフェース、LSB ファースト、チップ・ソフト・リセットのような機能に対する物理インターフェースの設定に限定されています。すべての制御の設定は、チップにプログラムされる状態を指示しています。

共通の ADC 機能は、**ADCBase 0** タブを選択するとアクセスできる **ADCBase 0** ページにあります。

デバイスが複数の ADC を持っている場合、各 ADC チャンネルにローカルな機能は、**ADC A~ADC Z** のタブに表示されます。ADC が 1 個の場合は、**ADCBase 0** タブのみがあります。すべてのコンバータがすべての機能をサポートしているわけではないため、使用可能な個別機能とプログラムされた値に対するデバイスの応答方法については個々のデバイスのデータ・シートを参照してください。

補助 ADC 機能は、ラベル表示されたタブを選択してアクセスできるその他のページに含まれています。複数のページが必要な場合には、複数のページが使用されています。これらはデバイスに固有な機能であるため、個々のデバイスのデータ・シートにドキュメント化されています。

コンバータによっては、ここでは追加ページが省略されている場合があります。この場合も、これらの機能については、個々のデバイスのデータ・シートを参照してください。一般に、これらの機能はここに記載する機能と矛盾することはありません。

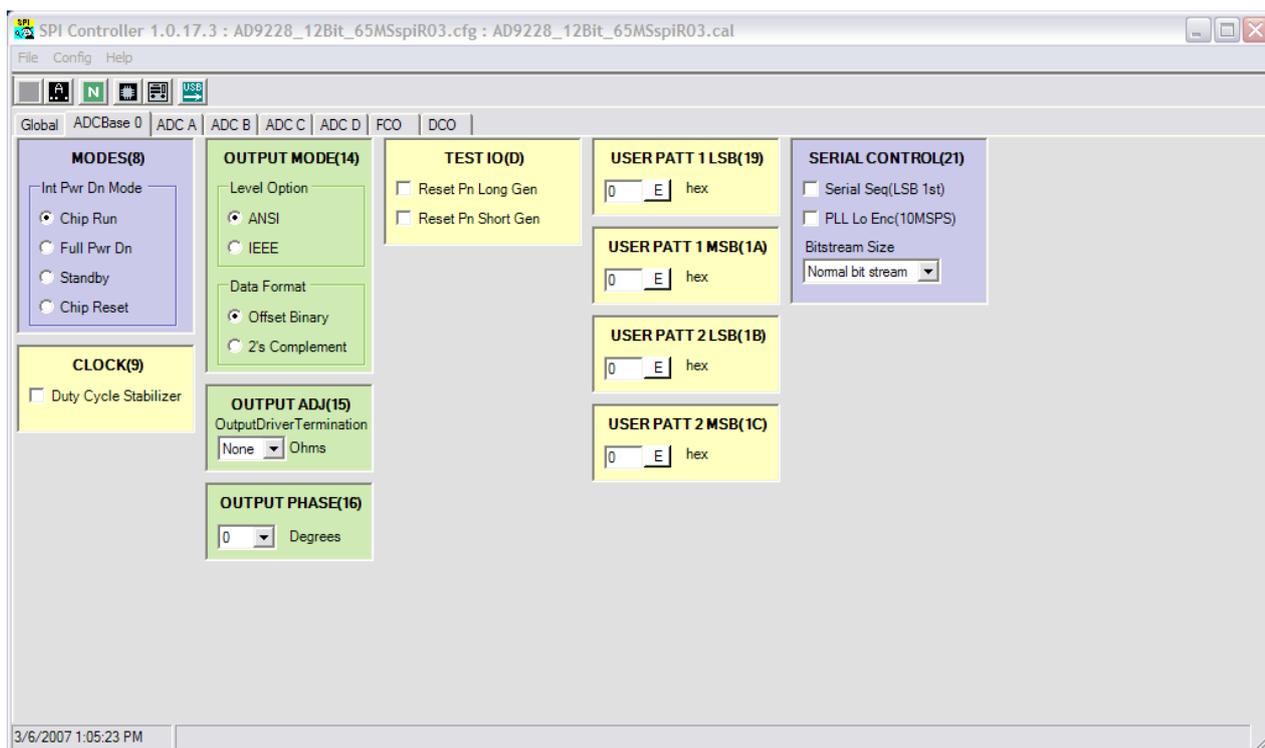


図 6. 代表的なユーザー・インターフェース—表示はデバイスごとに異なります

Fileメニュー

File メニューを使うと、**図 7** に示すように、種々の設定ファイルのオープンと保存を行うことができます。

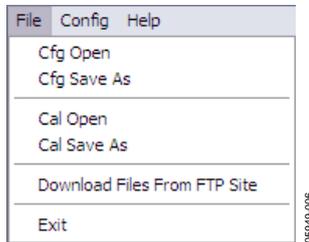


図 7. File メニュー

Cfg Open

SPI Controller は SPI ポートをサポートする大部分の AD92xx、AD94xx、AD66xx データ・コンバータと組み合わせて動作します (詳細については、デバイスのデータ・シート参照)。該当するデバイスのコントローラを設定するときは、**Cfg Open** を選択して、そのデバイスの設定ファイルを選択します。

設定ファイルはこのソフトウェアまたは評価ボードに添付されており、www.analog.com から提供しています。ウェブ更新も、このアプリケーション・ノートの **Download Files from FTP Site** のセクションで説明するように、提供されています。必要な設定ファイルをこれらの方法で見つけることができない場合は、highspeed.converters@analog.com へ電子メールでご要求ください。

Cfg Save As

設定ファイルを変更した場合にこのオプションをクリックすると、新しい名前でも保存することができます。プログラムは新しいファイル名を尋ねてきます。

Cal Open

Cal Open を使うと、セットアップ情報を取り出すことができます。前に保存したレジスタ設定を取り出すオプションを使うと、デバイスに所望の状態を能率良く設定できるようになります。

Cal Save As

デバイスのレジスタが望ましい状態にある場合、これらをファイルに保存することができます。これを行うときは、**Cal Save As** を選択して、プロンプトに対して新しいファイル名を入力します。

Download Files from FTP Site

このオプションを使うと、アナログ・デバイセズの公開 FTP サイトから新しい設定を直接ダウンロードすることができます。これは、使用する設定ファイルと実行可能ファイルを最新のものに維持する最も迅速かつ容易な方法です。同様に、設定ファイルのない評価ボードを持っている場合には、ここを最初に探してください。

このオプションをクリックすると、アナログ・デバイセズの公開 FTP サイトからデータがコピーされることが通知され、これをキャンセルするオプションも同時に表示されます。

Yes を選択すると、プログラムは現在の設定を保存して、プログラムを終了し、更新プログラムを実行し、プログラムに戻り、直前の設定の再ロードを行います。あるいは、設定を製品ウェブサイト www.analog.com から手動でダウンロードするか、highspeed.converters@analog.com にデバイスの最新 SPI 設定ファイルを要求してください。

Exit

SPI Controller プログラムを終了するときは、**Exit** をクリックします。

Configurationメニュー(設定メニュー)

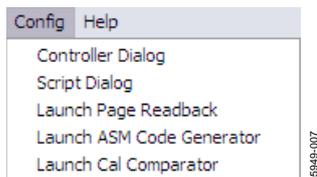


図 8.設定メニュー

Controller Dialog (SPI Controller の設定ダイアログ)

このオプションを使うと、デバイスのチップ・セレクト、種々のプログラム・フロー制御オプション、エラー処理(図 9 参照)、代わりにインターフェース・オプションの選択が可能になります。一般に、これらの選択はデバイスの設定プロセスで設定されるため、変更する必要はありませんが、選択を変更する機能は、高度なユーザーにとって便利ことがあります。

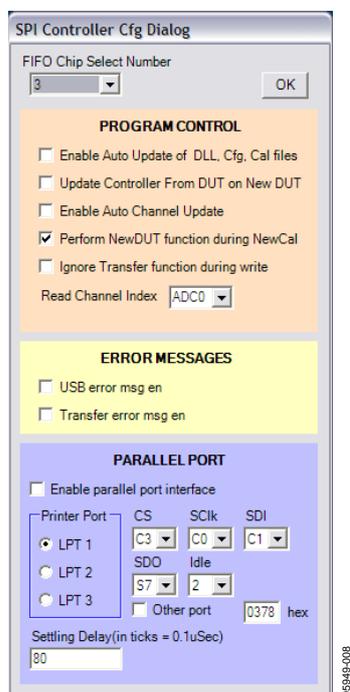


図 9.SPI Controller の設定ダイアログ・ボックス

FIFO Chip Select Number (USB FIFO チャンネル番号)

FIFO Chip Select Number メニューを使うと、SPI Controller が所望コンバータのチップ・セレクト・ラインをアクティブにするように設定することができます。大部分の評価ボードは **Chip Select Number 1** を使いますが、このオプションを使うと、必要に応じて別の**チップ・セレクト番号**を使うことができます。デフォルトでは、プログラム実行時に該当する値が選択されるので変更する必要はありません。

Program Control

Program Control を使うと、多くのオプション機能をイネーブルすることができます。機能をイネーブルするときは、機能の左にあるチェックボックスを選択します。

Enable Auto Update of DLL, Cfg., Cal files

この機能を選択すると、プログラムが実行されるごとに、プログラムがアナログ・デバイセス公開 FTP サイトを調べて最新プログラムと設定ファイルを探すように指示することができます。このオプションを選択する場合には、インターネット・アクセスが必要であるため、アクセス可能か否かチェックする必要があります。File メニューで **Download Files from FTP Site** を選択すると、手動で更新を取得することができます。

Update Controller From DUT on New DUT

このオプションを選択すると、**Update Controller From DUT on New DUT** ボタンを押したとき、内蔵レジスタがコントローラのグラフィカル・ユーザー・インターフェース(GUI)へ転送されます。このチェックボックスをチェックしない場合、**Update Controller From DUT on New DUT** ボタンは、GUI 内の値を DUT に書き込みます。DUT は被テスト・デバイスを意味します。

Enable Auto Channel Update

マルチチャンネル ADC でこのチェックボックスをチェックすると、複数のチャンネル間で同じ機能の更新が可能になります。

Perform New DUT function during NewCal

チェックすると、新しいキャリブレーション・ファイルがロードされたとき、GUI は新しい DUT 機能を実行します。

Ignore Transfer function during write

すべての ADC で、各コマンドまたはコマンドのグループの後に転送コマンドを送信する必要があるとはかぎりません。このボックスをチェックすると、このプログラムにより生成された自動転送は実行されなくなります。

Read Channel Index

Index Read Channel Index ドロップダウン・ボックス内の値により、リードバック・サイクルでアクセスする ADC チャンネルを指定します。

エラー・メッセージ

USB error msg en

エラーが発生したとき、USB エラー・メッセージの表示を可能にします。チェックしない場合、メッセージは表示されません。

Transfer error msg en

エラーが発生したとき、転送メッセージの表示を可能にします。チェックしない場合、メッセージは表示されません。

Parallel Port

USB SPI Controller によりサポートされていないデバイスに対してパラレル・ポートを提供します。現在、このインターフェースを使用するデバイスはありません

Script ダイアログ

このオプションを使うと、ログの生成、疑似コード(Cコード)ファイル、アセンブル(PIC12F629)ファイルをイネーブルすることができます(図 10 参照)。

ログ・ファイルは、この機能がイネーブルされている間に SPI ポートへ送信されたコマンドのレコードであり、レコード・マクロ・セッションのように機能します。このオプションをイネーブルまたはディスエーブルするときは、ボックスをそれぞれチェックまたは非チェックします。履歴とドキュメント化のために、この機能は送信されたコマンドを調べるときに便利です。**Clear** がクリックされるまで、このファイルの追加とこのファイルへの追加が続きます。

スクリプト・ファイルには C コード・ファイルが含まれ、デバイスの SPI ポートへ渡されたデータのシーケンスを表しています。このファイルはプログラミング・リファレンスとして使うことができ、あるいは最終アプリケーションのソース・コードに含めることもできます。フォーマットは標準 C/C++構文に準拠しています。

ソフトウェアが最終アプリケーションに含まれる場合、ユーザーはハードウェア固有の読み出しおよび書き込みユーティリティを用意する必要があります。この機能は、ユーザーがデバイスのプログラム方法を理解するのに役立ちます。次に示すコード例は、スクリプト・ファイルとして書かれた代表的なプログラミング・シーケンスです。これらのファイルには注釈が付いており、設定ファイルのシーケンスが説明されています。

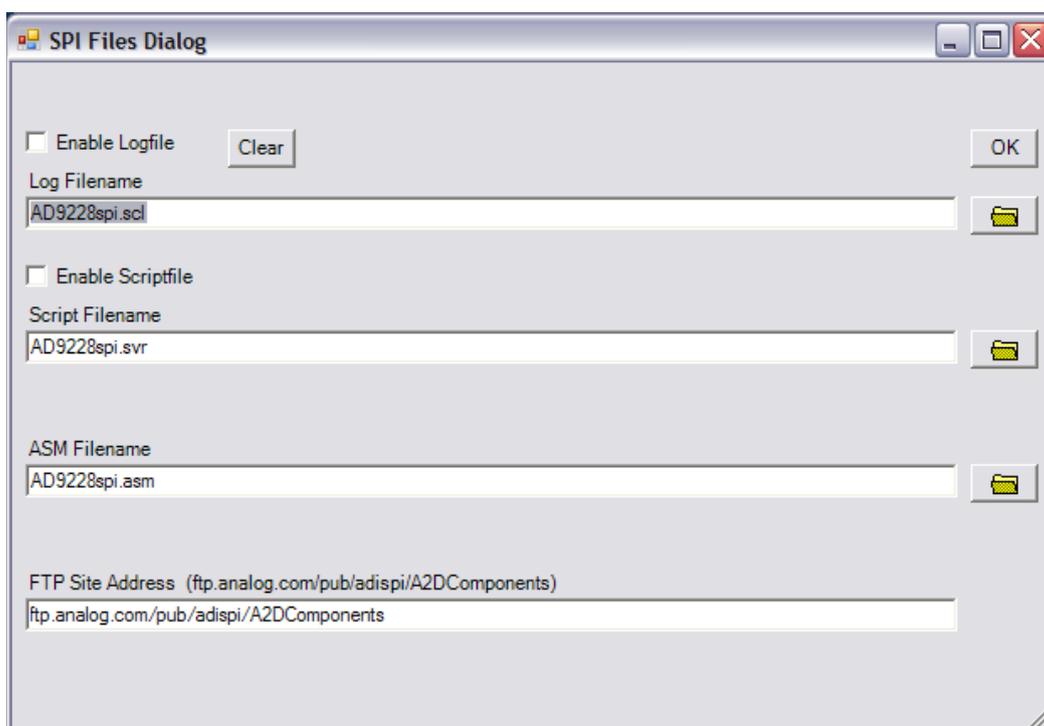


図 10.生成する代表的な C ソース・コードの SPI ファイル・ダイアログ・ボックス

```
//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)
// {
```

```

//      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
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
```

ASMFilename は、PIC12F629 アセンブル・コードの生成に使われるファイル名です。これらのファイル生成の詳細については、Launch ASM Code Generator のセクションを参照してください。その他の情報は、アプリケーション・ノート [AN-812](#) 「Microcontroller-Based Serial Port Interface (SPI®) Boot Circuit application note」に記載されています。

FTP Site Address は、最新の実行可能形式と設定をロードする更新ソフトウェアによって使われるアドレスです。これは一般に変更されることはありませんが、たとえば、サイトが将来移動する場合などで、このオプションを使うことができます。

Page ReadBack の起動

Page ReadBack 機能を使うと、接続されたデバイスの内蔵メモリを読み出すことができます(図 11 参照)。

Read ボタンをクリックすると、リードバックと比較プロセスが起動されます。ウィンドウが 1 つ開かれて、メイン・コントローラ・ウィンドウは ReadBack ページとは独立に動作します。このウィンドウにより値の変更が容易になり、デバイスがデータを正しく受信したことを確認することができます。

ReadBack ページには、レジスタが存在するタブ、レジスタの説明、アドレス、書き込み/読み出し値が表示されます。分かりやすくするため、pass/fail フラグが書き込まれています。この機能はデバイスのデバッグ時に便利です。

この機能とメイン・インターフェースを組み合わせると、ユーザー・メモリ・マップ全体を書き込むことができ、これを調べることが可能になります。

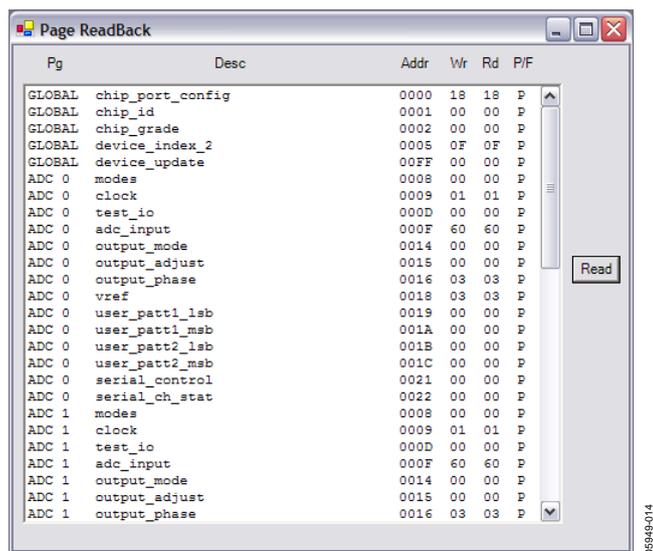


図 11 .Page ReadBack ダイアログ・ボックス

Launch ASM Code Generator

ASM Code Generator は評価ボード上の Microchip® PIC12F629 と組み合わせて使います(図 12 参照)。この回路は、非常に低価格な SPI Controller 用にエンドユーザー・ボードにコピーして配置することができます。コード・ジェネレータは、設定データを集める簡単な方法です。データを PIC コンパイラによりアセンブルして、デバイスに書き込んで、所望の状態を初期化します。詳細については、アプリケーション・ノート [AN-812](#) を参照してください。

ASM Code Generator ダイアログ・ボックスの 3 個のボタン(表 1)は、設定データの収集に役立ちます。

表 1.ASM Code Generator のボタン

Button	Description
	The Record button begins recording key strokes. While recording, as controls change, their values are recorded in the ASM Code Generator until recording stops. When this button is pressed again, it changes into the Stop Record button.
	The Erase button erases recorded commands. Use this button to correct commands or to erase commands so that new commands can be entered.
	The Write button writes the completed macro to a file. The file used is specified in the SPI Files Dialog (see Figure 10).

ASM Code Generator ダイアログ・ボックス(図 12 参照)には、インデックス・レジスタ 5 内の 4 個の ADC をすべてイネーブルするコマンド・シーケンス、デューティ・サイクル・スタビライザのディスエーブル、チップの起動が表示されています。

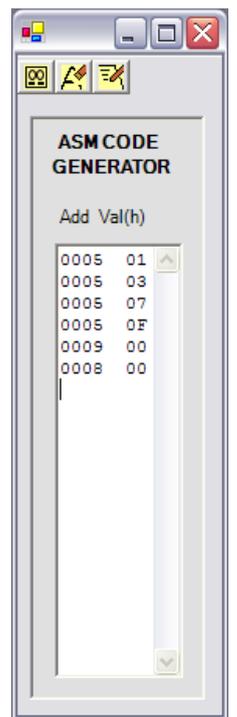


図 12.ASM Code Generator のダイアログ・ボックス

代表的な PIC12F629 ASM コード

```

; 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 comperator 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
    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

この機能は、2種類のキャリブレーションの比較を可能にするウインドウを表示します。DUT Default to Current State、DUT Default to a Specified File、Current State to Specified Fileのキャリブレーションが含まれています。高度なユーザーにとって、この機能は2つの設定の違いを比較する際に役立ちます。

Helpメニュー

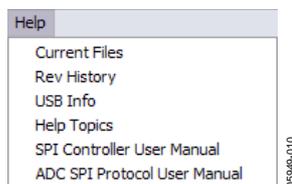


図 13.Help メニュー

Current Files

Current Files メニューは、プログラムの種々の部分で使われている重要なファイル名の一覧を表示します(図 14 参照)。これらのファイル名は **Config** メニューを使って変更することができます(**Script ダイアログ**のセクション参照)。Cfg ファイル、Cal ファイル、スクリプト・ファイル、ダイアログ・ファイルの変更を行うことができます。

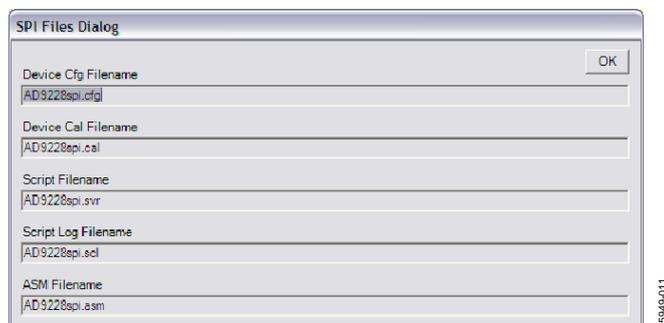


図 14.SPI ファイル・ダイアログ・ボックス

Device Cfg Filename

このファイルはデバイスに対してプログラムを設定する際に使われ、デバイスのレジスタとビットの情報が格納されています。レジスタとビットの情報はこのソフトウェア・パッケージに添付されています。あるいは www.analog.com から、または FTP ダウンロード機能を使って入手することができます(Download Files from FTP Site のセクション参照)。インストール CD にコンバータの設定が含まれていない場合には、highspeed.converters@analog.com へご連絡ください。

Device Cal Filename

このファイルは、各内蔵レジスタのデフォルト値を格納しています。ソフトウェアに添付されている初期ファイルにデフォルトのチップ値が含まれています。デフォルト値の変更を可能にする新しい Cal ファイルを生成することができます。

Script Filename

C コード・ジェネレータがこのファイルに書き込みます。このファイルは USB SPI Controller に送信されるデータと同じデータで生成されるため、評価ボードからデバイスをプログラムする際に使われる同じ情報を表しています。このコードには、読み出し機能を強化するドキュメントが組み込まれています。コードは完全に使用可能ですが、最適化されていません。コードの速度とサイズを最適化する様々なプログラミング技術が存在しています。このコードは、デバイス設定に役立つリファレンス・コードを確保するための開始点として提供するものです。

Script Log Filename

命令がこのファイルに記録されます。この機能をイネーブルすると、デバイスへ渡されたすべての命令もこのファイルに書き込まれるため、デバイスへ書き込まれたすべてのコマンドのログが生成されます。このファイルは、デバイスのプログラム履歴(ファイルがクリアされるまで)を含む追記ファイルです。このファイルは、デバイスに対して発行されたコマンドを調べ、動作の有無を確認するデバッグで便利です。

ASM Filename

PIC12F629 アセンブル・コード生成オプションを選択すると、このファイルへ書き込まれます。このファイルの詳細については、[AN-812](#) アプリケーション・ノートを参照してください。

Rev History

このオプションは、このソフトウェアの履歴の詳細を提供します。

USB Info

このオプションは USB ドライバの詳細を提供します。

Help Topics

このオプションは、プログラム機能の限定されたセットの説明を提供します。

SPI Controller User Manual

このオプションは、PC にインストールされたこのアプリケーション・ノートの PDF バージョンを表示します。

ADI SPI Protocol User Manual

このオプションは、アプリケーション・ノート [AN-877](#) 「*Interfacing to High Speed ADCs via SPI*」を表示します。

Toolバー

表 2 に、SPI Controller ツール・バー(図 15 参照)にあるボタンの定義を示します。すべてのデバイスのすべてのモードで、すべての機能が使用可能であるわけではありません。



図 15.Tool バー

表 2.SPI Controller の Tool バー

Tool	Symbol	Description
Transfer		The Transfer button programs the on-chip registers with the current values in the program. If the script file is enabled (C code generator), this file is written to at the same time, but log files are inhibited. When the program is launched, this button is disabled and grayed out indicating that asynchronous transfers are enabled. If asynchronous transfers are disabled, this control is enabled and all registers on-chip are simultaneously updated when the  button is pressed.
En Async Xfer		This button toggles between asynchronous transfer and synchronous transfer. The default mode for this control is asynchronous transfer, which means that as a button, text box, or other control is changed, the information is immediately passed to the DUT. When in synchronous transfer mode, changes are not made until the Transfer button is pressed. When asynchronous mode is enabled, the Transfer Button is disabled. If script files are enabled, they are inhibited. If log files are enabled, they are updated.
New DUT		This button reinitializes the DUT interface and issues a device soft reset to Register 00h. In addition, the Chip ID is read to ensure that the correct device is recognized. If Use defaults on New DUT is checked on the SPI Controller Cfg Dialog box, then the chip defaults are used instead of the values currently shown in the software.
Update DUT		Updates the registers on the DUT to match those in the controller software. This feature is useful if the device is powered down without exiting the software.
Update Controller		Updates the SPI Controller program to match those in the DUT. This feature is useful if the program was exited and restarted, but the device was not powered down.
USB Refresh		This button reestablishes communications with the USB interface if it has been broken due to power down or a cable being disconnected.