

LTC PSM デバイスのEEPROM の概要

Nick Vergunst

はじめに

リニアテクノロジーは、あらゆるアプリケーション向けに柔軟な構成で大きな電力を供給する各種のデバイスを提供しています。これらのデバイスはEEPROM形式のオンボード不揮発性メモリ(NVM)も備えています。このNVMは構成パラメータの格納と呼び出しに使用され、一部のデバイスではフォルト・ログおよびユーザー・スクラッチ・パッドを提供します。

これらのパワーシステム・マネージメント(PSM)デバイスのアーキテクチャでは、電源投入時にこのNVMから希望の構成パラメータを自律的に読み込むことができます。I²Cやファームウェアとのやり取りは不要です。

お客様からよく聞かれるのは、「構成は決まりましたが、この構成をチップのオンボード不揮発性メモリ(NVM)にプログラムするにはどうすればいいですか」という質問です。

この資料では、利用可能な方法について、簡単なほうから順番に説明します。

NOTE: 量産前のプロトタイプ作成には**オプション1**、大量生産には**オプション2**、**オプション3**、または**オプション4**を使用することを強く推奨します。**オプション5**は、ボード組み立て前の少数のICをプログラムするのに便利な方法です。**オプション6**にはソフトウェア開発の高度な専門知識が必要です。

全オプション一覧

- オプション1A** LTpowerPlay™内蔵プログラミング・ユーティリティ(PC->NVM)による手動プログラミング
- オプション1B** 通常モード(PC->RAM->NVM)のLTpowerPlayによる手動プログラミング
- オプション1C** LTpowerPlay コマンド・ライン・インタフェースによる完全に自動化されたプログラミング
- オプション2A** 1個~499個の事前プログラム済みのデバイスをLinear Express®で購入
- オプション2B** 500個以上の事前プログラム済みのデバイスをリニアテクノロジー社から購入
- オプション3A** BPM Programmerによるデバイスのセルフプログラミング
- オプション3B** 事前プログラム済みのデバイスをArrowから購入
- オプション4A** スキャン・チェーン上のI/Oデバイスを使用した、JTAGによるカスタム・ボード上のデバイスのセルフプログラミング
- オプション4B** スキャン・チェーン上のI/Oデバイスを使用しない、JTAGによるカスタム・ボード上のデバイスのセルフプログラミング
- オプション5** LTC® PSMソケット付きプログラミング・ボードによるデバイスのセルフプログラミング
- オプション6** カスタム・ファームウェアのオーサリングとデバイスのセルフプログラミング

**非常に多くの選択肢があります。
どれを選んだらよいでしょう。**

LTC PSM 製品のプログラミングには多くの方法があります。強力なツールを思いどおりに利用して、1枚のボード上で複数の方法を実行することも可能です。

LT、LT、LTC、LTM、Linear Express、Linear TechnologyおよびLinearのロゴはリニアテクノロジー社の登録商標です。LTpowerPlayはリニアテクノロジー社の商標です。その他全ての商標の所有権は、それぞれの所有者に帰属します。

アプリケーションノート 145

PSMデバイスのV_{CC}ライン、I²Cクロック(SCL)、I²Cデータ(SDA)、およびグランドにアクセスできるように、ボード上にアクセス可能ヘッダを配置することを強く推奨します。このヘッダはLTpowerPlayにエントリポイントを提供します。これにより、ボードを簡単にプログラムできるだけでなく、遠隔測定値の読み出し、オンザフライでの変数の変更、デバッグの実行を可能にする非常に強力なツール・スイートを利用できるようになります。

この機能はデバッグ段階で非常に役に立ちます。絶えず進化する今日の電力環境においては、開発がかなり進むまで、電源およびシステム全体の最終的な要件が決まらないことがあります。したがって、設計プロセスの任意の段階でシステム要件に合わせて構成を調整または根本的に変更できる機能は、非常に有効です。

プロトタイプの実成または少〜中量生産の場合、DC1613Aなどの外付けUSB dongleを接続してPSMデバイスをプログラムするには、基本的なヘッダだけで十分です。これはオプション1A、オプション1B、オプション1C、オプション4Bのプログラミングの例です。

繰り返しのようになりますが、どのプログラミング手法を選択するかを問わず、ヘッダを配置することを強く推奨します。その理由は、dongleによって電力供給されるPSMデバイスをオンボードでプログラムすることができ、ボード上の電源レールに依存せずにPSMデバイスの通電状態を維持できるからです。このことはあらゆるデバッグ状況で極めて有効です。このデバイスとLTpowerPlayソフトウェアを組み合わせれば、電源システム全体の強力なデバッグ手法を提供します。遠隔測定値、フォルト状態レジスタおよびフォルト・ログ

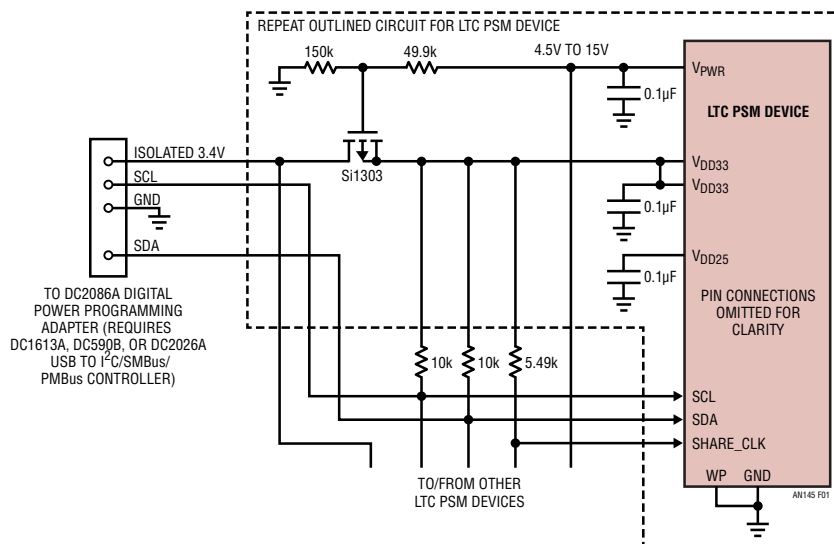


図1. PFETを使用してLTC PSMデバイスのV_{PWR}に電力を供給する場合のLTCコントローラの接続

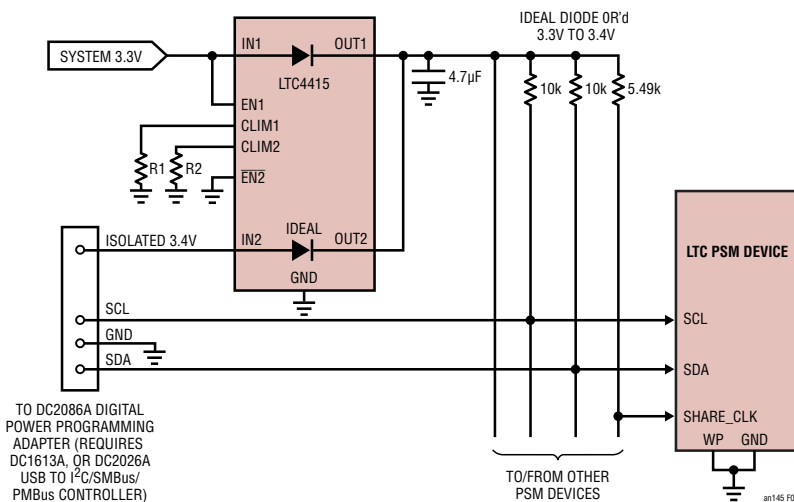


図2. システム電源またはdongle電源からLTC PSMデバイスに電力を供給する場合のLTCコントローラの接続

を使用して、短時間で故障を診断できます。このインタフェース1つを使用して、最終的なシステム構成を素早く開発し、PSM製品の各不揮発性メモリに格納できます。

これらの極めて重要な機能を最大限に利用し、DC1613AまたはDC2086AからLTC PSM製品に電力供給できるようにしたボードの設計例については、図1および図2を参照してください。

これらのデバイスをボード上でプログラムしたくない場合にも、複数のオプションが利用可能です。デフォルト状態のデバイスを発注し、ソケットを使用してプログラムしてからボードに半田付けしたい場合は、オプション3Aまたはオプション5を選択してください。2つ以上のデバイスを同時にプログラムする場合は、オプション3Aを選択し、BPMのマルチサイト・プログラマを使用できます。

事前プログラム済みのデバイスを発注する場合は、オプション2またはオプション3Bを選択してください。これらのオプションを選択した場合、デバイスは、選択した構成ファイルをデバイスにプログラムした状態で配送されます。このプロセスを選択した場合、前述の推奨事項に従ってボード上に基本的なヘッダを配置していないと、後で要件が変更された場合やデバッグを実行する必要がある場合にLTpowerPlayがデバイスを再プログラムできなくなることに**注意してください**。

ボード上に他のスマート・デバイス(FPGA、ASIC、MCUなど)を配置する場合は、オプション6を使用してカスタム・ファームウェアをオーサリングできます。使用するスマート・デバイスがJTAGに対応しており、I²Cラインがサウンダリ・スキャン・ノードに接続されている場合は、オプション4Aを使用して、LTC PSMデバイスとほかのJTAG対応デバイスを同時にプログラミングできます。この方法は、中～大量生産の場合に組み立て時間を短縮するのに有効です。

どのオプションを選択する場合も、初回製品を後で検証できるように、LTpowerPlayがプログラミング・ファイルから作成する*.zipファイルを保存しておくことを強く推奨します。

LTpowerPlay GUIを使用するオプション

ここで説明するオプションのほとんどは、リニアテクノロジーの強力なLTpowerPlay GUI (<http://www.linear-tech.co.jp/LTpowerPlay> から入手可能)を使用します。一般的には、LTpowerPlay GUIとターゲット・ハードウェアのデモボード(LTC2977用のDC1540Bなど)を使用して、ボード上の1つまたは複数のICのプロジェクト・ファイルを作成します。このデモンストレーション・システムを使用して、ボードの立ち上げの前に、この構成パラメータで目的の電源管理動作が実現されるかどうかを検証できます。プロジェクト・ファイル(.projファイル)に構成パラメータを保存した後、次のいずれかの方法で、この構成をボード上のICの不揮発性メモリ(NVM)に転送できます。

オプション1: LTpowerPlayを使用して顧客のボード上のLTC PSMデバイスをプログラムする

LTpowerPlay GUIは、リニアテクノロジーが設計したデモボードだけでなく、カスタム・ボード上の任意の数のLTCデバイスとも通信できるように設計されています。

NOTE: 量産前のプロトタイプ作成には、オプション1を強く推奨します。これは、オプション1にはI²Cプロトコルの知識が不要で、ファームウェアを開発する必要がないからです。また、オプション1では、デバイスをデソルダリングせずにボード上で構成を変更できます。

オプション1A: LTpowerPlay内蔵プログラミング・ユーティリティによる手動プログラミング

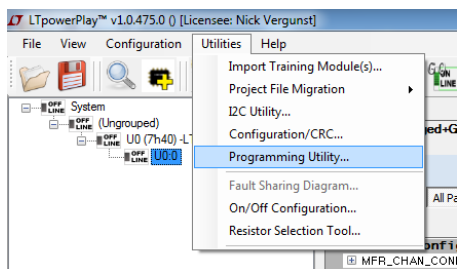
LTpowerPlayは、構成ファイルのプログラミングと検証を簡単に実行できる内蔵プログラミング・ユーティリティを備えています。LT USB-PMBusコントローラ(DC1613A)をPCのUSBポートとターゲット・ボードに接続します。書き込み禁止ピンが“L”になっており、ソフトウェア書き込み禁止が無効になっていることを確認します。このプロセスの詳細な回路図については、図1および図2を参照してください。

LTpowerPlay GUIで、以下の手順を実行します。

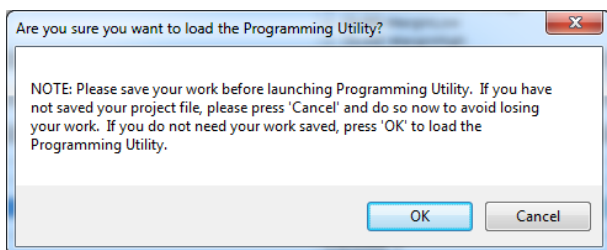
- 現在のプロジェクト・ファイルをまだ保存していない場合は、必ず保存してください。以下の手順を実行すると、現在開いているプロジェクト・ファイルは閉じられ、保存されていない変更は全て廃棄されて、プログラミング用のプロジェクト・ファイルが開かれます。

アプリケーションノート 145

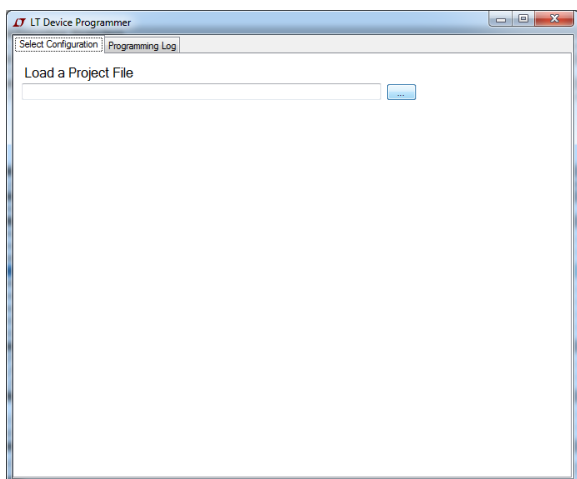
- [Utilities]->[Programming Utility]を選択します。



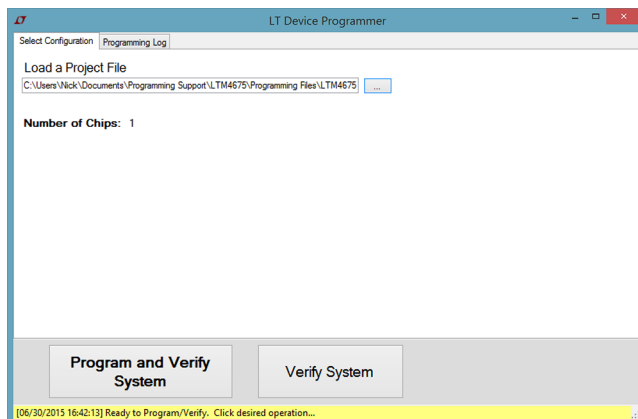
- LTpowerPlayは、現在開いているプロジェクト・ファイルを保存したかどうか確認してきます。現在開いているプロジェクトをまだ保存しておらず、ここで保存したい場合は、[Cancel]をクリックして、プロジェクトを保存してからプロセスを再開します。



- [LT Device Programmer]ウィンドウが表示されます。[Select Configuration]タブが開いていることを確認します。[...]ボタンをクリックし、デバイスのプログラミング用のソース・ファイルとして使用したいプロジェクト・ファイルを参照します。



- *.projファイルを参照して[Open]をクリックすると、[LT Device Programmer]ウィンドウは次のように変わります。



- 画面下部に2つの大きなボタンがあります。
 - [Program and Verify]ボタンをクリックすると、デバイスのNVMを適切な状態にプログラムしてから、その内容を検証します。
 - [Verify]ボタンをクリックすると、デバイスのNVMをプログラムせずに、プロジェクト・ファイルの内容に対してNVMの内容を検証します。詳細については、付録Bを参照してください。
- この演習では、[Program and Verify]をクリックします。画面最下部のステータス・メッセージが黄色に変わり、“Programming”、次に“Verifying”、最後に(プロセスが正常に終了した場合)“Successfully Programmed and Verified Device”と表示されます。



ここで次にプログラムするボードにUSB Dongleを接続し、[Program and Verify]ボタンをクリックします。新しいボードも最初のボードと同じようにプログラムされます。




後で必要に応じてプログラム済みデバイスの検証を実行できるように、プログラミングに使用したプロジェクト・ファイルは、バージョン管理されるリポジトリに別に保存しておくことを強く推奨します。

オプション1B:通常モード(PC->RAM->NVM)の LTpowerPlayによる手動プログラミング

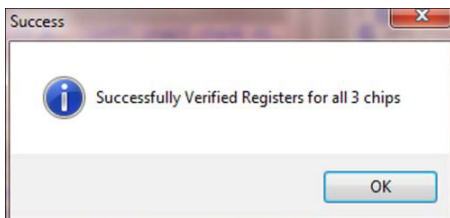
このプロセスは非常にシンプルです。

LT[®] USB-SMBusコントローラ(DC1613A)をPCのUSBポートとターゲット・ボードに接続します。このプロセスの詳細な回路図については、図1および図2を参照してください。

LTpowerPlay GUIで、以下の手順を実行します。

- ツールバーの[Open]ボタンをクリックして、プロジェクト(.proj)ファイルを読み込みます。
- ツールバーの[Go Online]をクリックします。
- ツールバーの[Write All]ボタンをクリックします。

LTpowerPlay GUIは、プロジェクト・ファイルに基づいて各オペレーション・レジスタのプログラミングと検証を実行します。このプロセスが正常に終了した場合、次のようなメッセージが表示されます。



- ツールバーの[Store User All]ボタンをクリックします。



ボード上の全てのチップの構成がNVMに格納されます。

オプション1C:LTpowerPlayコマンド・ライン・インタフェースによる完全に自動化されたプログラミング

GUIを使用してデバイスに直接プログラムするもう1つの方法は、シェルのコマンド・ライン・パラメータを渡すことです。LTpowerPlayは、人手による操作なしで、デバイスへの接続、プログラミング、検証を自動的に実行します。全てのプログラミングが完全に自動化された少量生産環境では、この機能が非常に便利です。

Microsoft Windows環境から、コマンド・シェル・ウィンドウを開きます。これを簡単に実行するには、XP/Vistaでは[Start]->[Run]に進んで“cmd.exe”と入力します。Windows7/Windows8ではスタート・メニューに“cmd”と入力します。

あるいは、Windows/System32フォルダを参照してcmd.exeを探し、ほかのプログラムと同じように開くことができます。

NOTE:ほかのプログラミング言語を使用して、GUIによる自動プログラミングを更に自動化できます。最も簡単な方法は、コマンド・ライン・コマンドの単なる集合体であるWindowsバッチ・ファイル(*.bat)を作成することです。ループ処理、エラー・チェックやその他の機能もこの方法で実装できます。必要なのはテキスト・エディタだけです。

C#, C++, C, VB, Delphiなど、現在の多くのプログラミング言語では、プログラムの実装と保守が容易に行えるので、プログラムからWindowsのコマンド・ライン・オプションを呼び出せるようになっています。これにより、LTpowerPlay GUIの自動プログラミング機能を、任意のシステムに簡単に組み込むことができます。

PATH環境変数のリストにLTpowerPlayのインストール・フォルダが追加されていない場合は、インストール・ディレクトリに移動するか、コマンドに完全なディレクトリ名を入力する必要があります。LTpowerPlayをインストールしたディレクトリに移動するのが最も簡単です。

“cd ”(スペースに注意)と入力し、LTpowerPlayがインストールされているディレクトリの絶対パスを入力します。x64ベースのシステムではデフォルトが“C:\Program Files (x86)\Linear Technology\LTpowerPlay”なので、次のように入力します。

```
cd "C:\Program Files (x86)\Linear Technology\LTpowerPlay"
```

Windowsのバージョンによっては、Windows Explorerのアドレス・バーからディレクトリの文字列をコピーし、コマンド・シェルに貼り付けることで、入力ミスを回避することもできます。x86ベースのマシンにはProgram Filesフォルダが1つしかないため、パスは次のようになります。

```
cd "C:\Program Files\Linear Technology\LTpowerPlay"
```

このディレクトリ内に、ユーザーが入力できるコマンド・プロンプトが表示されます。このコマンド・ラインから“C:\myProgrammingProject.proj”に格納されたプロジェクト・ファイルを使用してデバイスのプログラミングと検証を実行するには、次のように入力します。

```
LTpowerPlay.Shell.exe -Program -Verify -ProjectFile="C:\myProgrammingProject.proj"
```

これでGUIシェルが自動的に起動し、デバイスのプログラミングと検証が実行され、結果によって成功メッセージまたはエラーが返されます。GUIを使用した手動プログラミングの場合と同様に、 dongleが接続されている必要があることに

アプリケーションノート 145

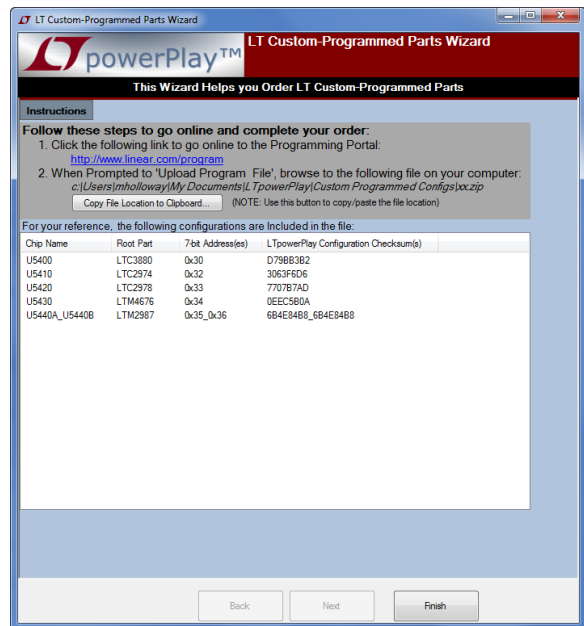
注意してください。後で必要に応じてプログラム済みデバイスの検証を実行できるように、プログラミングに使用したプロジェクト・ファイルは、バージョン管理されるリポジトリに別に保存しておくことを強く推奨します。

オプション 2A: Linear Express を使用して事前プログラム済みのデバイスをリニアテクノロジー社に発注する (サンプル数量 500 個未満)

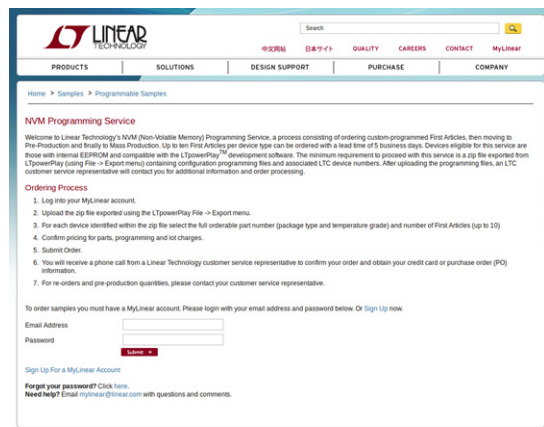


Linear Express First Article and Pre-Production Programming (LTEXpress) は、事前プログラム済みのデバイスをリニアテクノロジー社から直接入手できる迅速で信頼性の高い方法です。LTEXpress は LTpowerPlay と緊密に統合されているため、簡単に利用できます。GUI の 1 クリック操作でプロジェクトをパッケージングした後、linear.com の Web サイトにそのファイルをアップロードしてプロセスを完了すると、プログラム済みのデバイスが 2、3 日以内に配送されます。

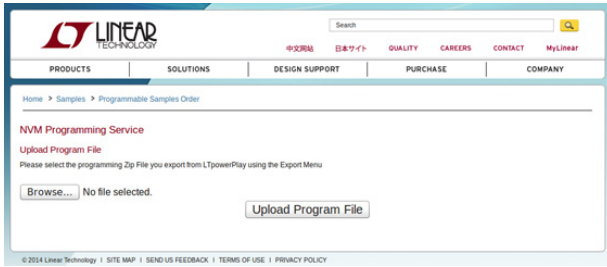
- ボードのシステム全体を含むプロジェクト・ファイルを開きます。プログラミングしたくないデバイスがある場合は、後で個々のアイテムを選択解除できます。
- LTpowerPlay のトップ・メニューから、[File]->[Order Custom LT-Programmed Parts] を選択します。
- [LT Pre-Programmed Parts Programming Wizard] メニューが表示されます。[Next] をクリックして続行します。
- ウィザードは、全てのデバイスを 1 つのアーカイブ・ファイルに自動的にパッケージして、[Documents] フォルダ (Windows XP/Vista では [My Documents] フォルダ) 内の “My LTpowerPlay Files\LT Programming Orders” の下に置きます。*.zip アーカイブには、プロジェクト・ファイル名の末尾に日付と時間を追加した名前が自動的に付けられます。LTpowerPlay に表示されているウィンドウには、作成されたアーカイブへの完全なパスとアーカイブの完全な名前が表示されます。また、アーカイブに含まれる各デバイスと製品番号、デバイスのアドレス、その特定の構成のチェックサムを示す表も表示されます。



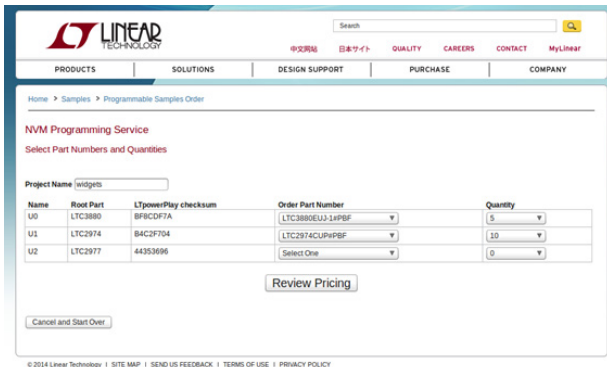
- このウィンドウには、オンライン・プログラミング・ポータル (<http://www.linear-tech.co.jp/program>) に接続するリンクもあります。このリンクが機能しない場合は、Web ブラウザを開き、<http://www.linear-tech.co.jp/program> に移動します。このページには、上記の手順を解説したチュートリアル・ビデオもあります。



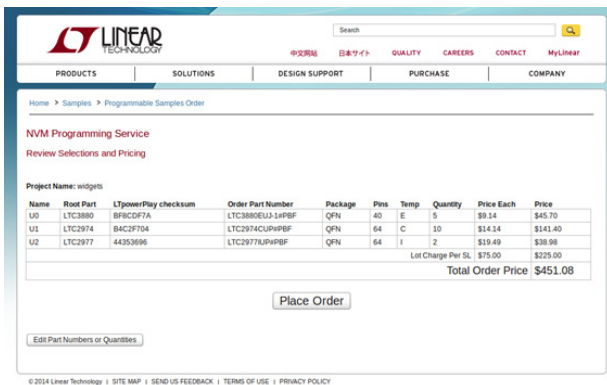
- MyLinear アカウントにログインします。
- [File]->[Order LT Custom-Programmed Parts...] メニュー・アイテムを使用して、LTpowerPlay からエクスポートされた *.zip ファイルをアップロードします。Parts Wizard の [Copy File Location to Clipboard] ボタンをクリックすると、ファイルの正確な場所をアップロード・ポータルに貼り付けることができます。



- zip ファイルに含まれる各デバイスについて、完全な発注製品番号(パッケージ・タイプおよび温度グレード)と品数(最大10個まで)を選択します。



- デバイスの単価、プログラミング料金およびロット価格を確認します。



- 発注します
- Linear Expressの顧客サービス担当者からメールが送信され、ご注文の確認と、クレジット・カードまたは発注書(PO)情報をお伝えいただけます

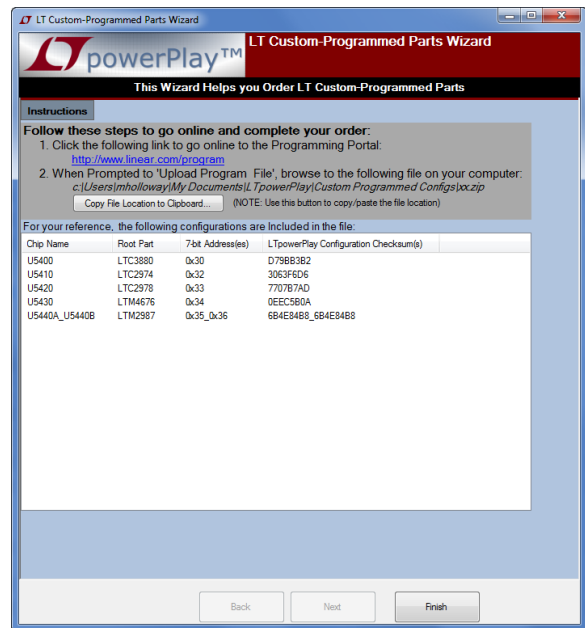
再注文および先行量産数量については、顧客サービス担当者にお問い合わせください。

後で必要に応じてプログラム済みデバイスの検証を実行できるように、このアーカイブは、バージョン管理されるリポジトリに別に保存しておくことを強く推奨します。

オプション2B:LTpowerPlayを使用して事前プログラム済みのデバイスをリニアテクノロジー社に発注する(量産数量)

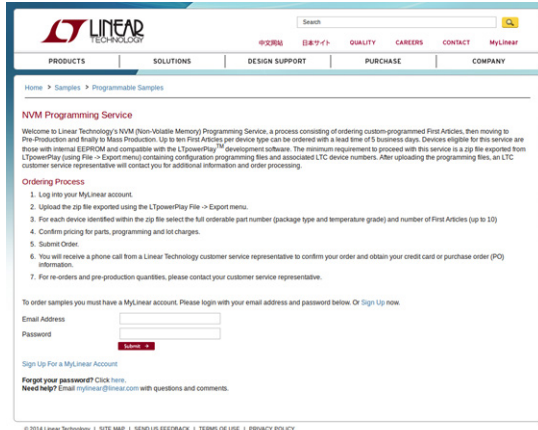
大量生産に適したオプション2Bでは、事前プログラム済みのデバイスをリニアテクノロジー社に発注します。最小発注数量については、リニアテクノロジーまでお問い合わせください。LTpowerPlayを使用して、希望するプロジェクト・ファイル(.proj)を保存します。以下の簡単な手順に従います。

- ボードのシステム全体を含むプロジェクト・ファイルを開きます。プログラミングしたくないデバイスがある場合は、後で個々のアイテムを選択解除できます。
- LTpowerPlayのトップ・メニューから、[File]->[Order Custom LT-Programmed Parts]を選択します。
- [LT Pre-Programmed Parts Programming Wizard]メニューが表示されます。[Next]をクリックして続行します。
- ウィザードは、全てのデバイスを1つのアーカイブ・ファイルに自動的にパッケージして、[Documents]フォルダ(Windows XP/Vistaでは[My Documents]フォルダ)内の“My LTpowerPlay Files\LT Programming Orders”の下に置きます。*.zipアーカイブには、プロジェクト・ファイル名の末尾に日付と時間を追加した名前が自動的に付けられます。LTpowerPlayに表示されているウィンドウには、作成されたアーカイブへの完全なパスとアーカイブの完全な名前が表示されます。また、アーカイブに含まれる各デバイスと製品番号、デバイスのアドレス、その特定の構成のチェックサムを示す表も表示されます。

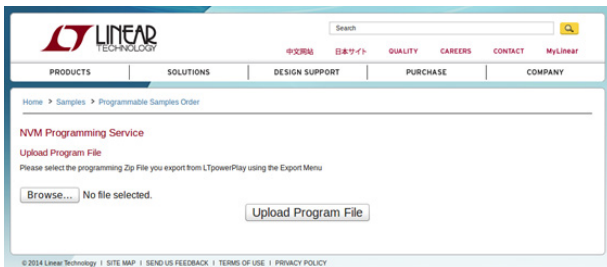


アプリケーションノート 145

- このウィンドウには、オンライン・プログラミング・ポータル (<http://www.linear-tech.co.jp/program>) に接続するリンクもあります。このリンクが機能しない場合は、Webブラウザを開き、<http://www.linear-tech.co.jp/program> に移動します。このページには、上記の手順を解説したチュートリアル・ビデオもあります。

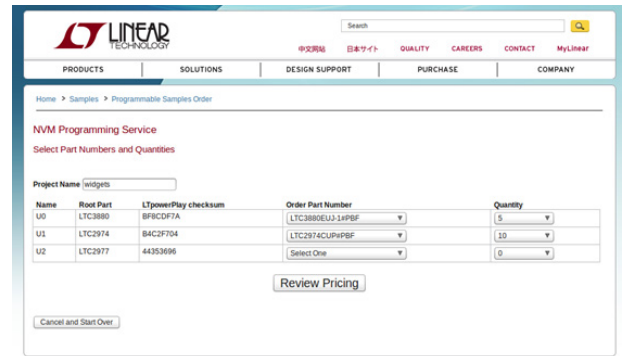


- MyLinear アカウントにログインします。
- [File] -> [Order LT Custom-Programmed Parts...] メニュー・アイテムを使用して、LTpowerPlay からエクスポートされた *.zip ファイルをアップロードします。

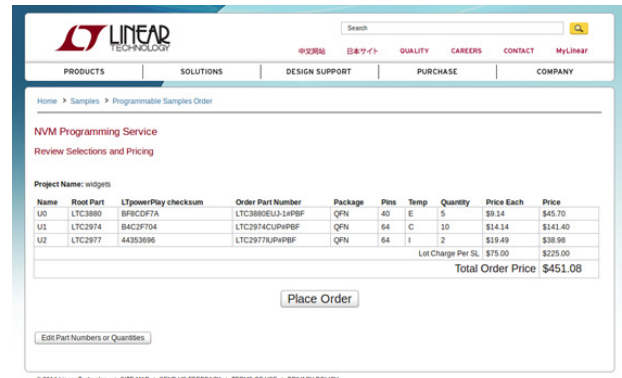


NOTE 1: 2つ以上の個別の構成を発注する場合は、それぞれの EEPROM CRC ID ごとに必要な個数を明確に指定する必要があります。
*.zip ファイルに含まれている手順ファイル (OrderingInstructions.txt) を参照してください。このファイルには、GUI プロジェクト・ファイル内の各 PC アドレス (または U ナンバ) と、固有の EEPROM CRC ID との対応関係が記載されています。

- zip ファイルに含まれる各デバイスについて、完全な発注製品番号 (パッケージ・タイプおよび温度グレード) と品数 (最大 499 個まで) を選択します。



- デバイスの単価、プログラミング料金およびロット価格を確認します。同意する場合は、[Place Order] ボタンをクリックします。



後で必要に応じてプログラム済みデバイスの検証を実行できるように、このアーカイブは、バージョン管理されるリポジトリに別に保存しておくことを強く推奨します。

.zip ファイルをリニアテクノロジーの担当者へ送信するだけで、事前プログラム済みのデバイスを発注できます。^{1,2}

NOTE 2: 事前プログラム済みのデバイスを LTC から購入しない場合、契約条件が適用されます。委託製造業者またはサードパーティ・ソフトウェア・ハウスにデバイスのプログラミングを依頼する場合は、デバイスの保証について LTC の担当者にご確認ください。

オプション3A: LTpowerPlayとBPM Micro Programmerを使用してプログラム済みのカスタム・デバイスを作成する

希望するプロジェクト・ファイル(.proj)を作成したら、以下の簡単な手順に従って、BPM Micro Programmerを使用する委託製造業者またはサードパーティ・ソフトウェア・ハウスに、事前プログラム済みのデバイスを発注します。

LTpowerPlay GUIで、以下の手順を実行します。

- ツールバーの[Open]ボタンをクリックして、プロジェクト(.proj)ファイルを読み込みます。
- メニューから、[File]->[Export]->[To Programming File]->[Package Project for BPM Micro Programmer...]を選択します(図3を参照)。
- 格納する.zipファイルの名前を参照します。

GUIは、上の手順で指定した場所に.zipファイルを保存します。.zipファイルには、プロジェクト・ファイル内の各チップ用のサブフォルダが含まれます。この*.zipファイルは、各ICに必要な個々の.oem hexファイルを1つのファイルに格納するために作成されます。後で必要に応じてプログラム済みデバイスの検証を実行できるように、このアーカイブは、バージョン管理されるリポジトリに別に保存しておくことを強く推奨します。

.zipファイルの各サブフォルダには、ボード上の対応する各IC用のOEM Programming Hex(.oem)ファイルがあります。

- この.oem hexファイルを委託製造業者に送信します。
- 委託製造業者は、この.oemファイルを使用して、希望どおりの構成で多数のICをプログラムします。

- プロジェクト内の残りの各チップについても、.zipファイルに含まれる個々の.oemファイルに対してこの手順を繰り返します。

注記: 請負製造業者がデバイスをプログラムするには、次の2つのハードウェアが必要です。

- BPM Micro Programmer
- デバイス用のソケット・モジュール

サポートされるデバイスのリストと詳細については、<http://www.bpmicro.com>を参照してください。

NOTE: BPM Micro Programmerは、BPWinソフトウェアを使用してデータ・パターンを読み込み、デバイスをプログラムします。顧客および請負製造業者は、通常は計算されたチェックサムを使用して個別の構成を特定します。LTpowerPlayはユーザーの代わりにチェックサムを事前に計算し、BPWinはこのチェックサムを使用して構成を計算し、特定します。詳細は、.zipファイル内のProgrammingInstructions.txtファイルに記載されています。

上記の手順で生成した.zipファイルを開き、メイン・フォルダでテキスト・ファイルProgrammingInstructions.txtを探します。このテキスト・ファイルには、プロジェクト・ファイル内の各チップの詳細な発注手順が記載されています。請負製造業者との通信に使用できるチェックサムは“Data Pattern CRC-32”という名前です。このチェックサムは、プロジェクト・ファイル内の各チップについて個別に計算され、図4に示すようにProgrammingInstructions.txtファイルに出力されます。

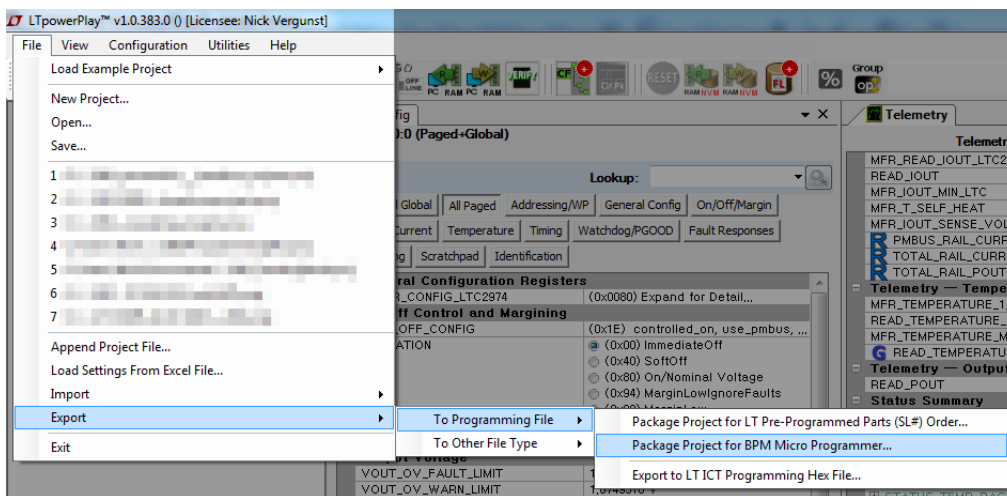


図3. [File]->[Export]->[To Programming File]->[Package Project for BPM Micro Programmer]の順に選択

アプリケーションノート 145

```
ProgrammingInstructions.txt - Notepad
File Edit Format View Help
This file describes the contents of the archive .zip file "ExampleProgrammingFile.zip". This .zip file
contains unique LT OEM Programming Hex (.hex) files, which can be used
individually with BPM Micro programming hardware to program that specific configuration,
or instructions on which files to send to the BPM Micro programmer
see the instructions below in the "Programming Instructions" section.

-----
Internal Tracking Information:
-----
The following identifying information is here for the purpose of internal tracking.

Tracking Information:
Order Package Archive File: 'fsdfs.zip'
Original Archive Location: 'C:\Users\Nick\Desktop\fsdfs.zip'
Date Created: '05/05/2014 12:41:19'
GUI Tracking Information
GUI Version: 1.0.383.0
Company: 'Linear Technology Corporation'
User Name: Nick vergunst
User Email: nvergunst@linear.com
Computer Name: Nickworkslaptop
Hardware-ID: 616A-6E7D-797E-388E-0039

Model Number: LTC2974
Silicon Rev:
Number of Chips in this Project Archive: 1
Number of unique configurations: 1
Project Structure:
Chip ID (0x3C)
LTPowerPlay Computed Configuration Checksum = 0x9420CCC2
User Config Comment: 'N/A'

-----
Page Names From the Project File:
-----
U0:0 - 'U0:0'
U0:1 - 'U0:1'
U0:2 - 'U0:2'
U0:3 - 'U0:3'

-----
Programming Instructions:
-----
This project contains 1 unique configurations.
The table below shows the individual .oem file
that corresponds to each chip in your project file:

To program an individual configuration, simply locate
the specific .oem file for the configuration you want to program
in this .zip archive and send this file to the programming house with the following instructions:

NOTE: At this time only BPM Micro programming hardware is supported.
Notes for BPM Micro Hardware/BPWin Software:
-----
If you are ordering pre-programmed parts from a programming
house, send these instructions to the programming house...
To enable simple serialization, enter Serial Number
Buffer Address "0x0C" and set size to 2 bytes under
Tools serialization in the BPWin software.

Please send the programming house the 'Data Pattern CRC-32'
value shown below for the configuration you want to order.
Instructions for programming house: In the BPWin software,
select the 'CRC-32' as the checksum method, and
'Data Pattern' for the checksum type. The BPWin computed checksum
should match the checksum provided ('Data Pattern CRC-32').

-----
Configuration Table:
-----
Chip LT Configuration Checksum Hex File Checksums OEM (.oem) File
U0 (0x3C) 0x9420CCC2 Data Pattern CRC-32: 0x856e4331 Hex File MD5: 0x08333334E0A3749FC28C0776FB842607 <ZipFile>\(0x
```

図4. .zipファイルに含まれるプログラミング手順

図4でハイライトされている手順は、ProgrammingInstructions.txtからの抜粋で、以下の方法を説明しています。

- “Data Pattern CRC-32”チェックサムを計算するようにBPWinソフトウェアを構成する
- オプションにより、デバイスの簡易なシリアル化を行うようにBPWinソフトウェアを構成する(シリアル・ナンバ・プログラミング)

オプション3B: Arrow プログラミング・サービスの利用

Arrowは、BPMプログラミング・ハードウェアおよびソフトウェアを利用したプログラミング・サービスを提供しています。³Arrowをプログラミング・サービスとして選択する場合は、以下の手順に従って、事前プログラム済みのデバイスを発注してください。

NOTE 3: Arrowの事前プログラム・デバイス・サービスについてご不明な点などありましたら、Arrow Global Programming Services (1-(775)-334-1000 または p4FirstArt@arrow.com) までお問い合わせください。

最初に、オプション3Aの手順に従って、Arrowに発注する各デバイスの構成について個別の*.oem hexファイルを作成します。オプション3Aで説明したように、このプロセスでは、発注を完了するのに必要な個々の*.oem hexファイルとProgrammingInstructions.txtファイルを含む*.zipファイルが生成されます。各デバイスの構成について、以下の手順に従って*.zipアーカイブ内のファイルを使用します。後で必要に応じてプログラム済みデバイスの検証を実行できるように、このアーカイブは、バージョン管理されるリポジトリに別々に保存しておくことを強く推奨します。

初回品発注用のArrowの事前プログラミング・ワークシートに記入します (<http://www.arrow.com/globalprogramming>)。

このWebフォームで、以下の項目に入力します。

- この構成に対応する固有の[Customer Part Number]を入力します。
- [Device Type]は[OTHER]を選択します。
- この構成用の*.oem hexファイルに対応する固有の“Data Pattern CRC-32”値を入力します(各*.oem hexファイルのData Pattern CRCが記載されている、*.zipアーカイブ内のProgrammingInstructions.txtを参照)。
- チェックサムの計算方法は[32-CRC]を選択します。
- フォームのその他の情報を入力します。ご不明な点などありましたら、Arrow Global Programming Services (1-(775)-334-1000、p4FirstArt@arrow.com) までお問い合わせください。

必要な情報の入力を終えたら、Webフォーム上の[SUBMIT]ボタンをクリックします。

この発注に関する確認メールが送信されます。電子メールに記載されたリンクを使用して、要求した構成に対応する*.oem hexファイルを添付/アップロードします。

オプション4: JTAGプログラミング

今日の複雑なシステム・ボードの多くは、少なくとも1つのJTAGプログラマブル・デバイスを搭載しています。1つのエントリポイントから複数のデバイスのプログラミングとテストが行えるように、これらのJTAGデバイスをデジチェーン接続して「スキャンチェーン」を構成することがあります。

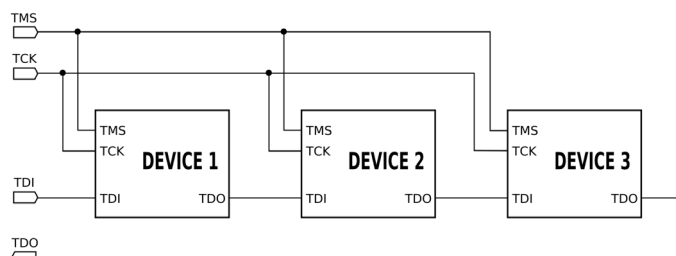


図5. JTAG スキャンチェーンの例

リニアテクノロジーは、JTAG TechnologiesおよびAsset Intertechと緊密に協力して、標準のJTAGプログラミング・インタフェースまたは同様のJTAGプログラミング・ハードウェアを使用したLTC PSM製品のプログラミング手法を開発しました。

オプション4A: I/OビットがPSM製品のPMBusに接続されているスキャンチェーン内のデバイスを使用するJTAGプログラミング

最も簡単で便利な方法は、バウンダリ・スキャン機能がピン上で利用できるスキャンチェーン上のデバイスに、PMbusを接続することです。JTAGプログラミングが可能なCPLD、ASIC、FPGA、およびMCUのほとんどは、PMbusの接続に対応しています。

PMbusがJTAGスキャンチェーン内のバウンダリ・スキャン対応ピンに接続されると、スキャンチェーン全体を通してデータを移動させることにより、これらのピン上でロジック“H”および“L”信号を強制し、測定できます。これにより、簡単なビットバンギング手法でバスにインタフェースできますが、信号がビットバンギングされるため、動作周波数が問題に

なります。クロックが遷移するたびに、スキャンチェーン全体を2回ループする必要があります。PMbusの動作周波数が10KHzより低いと、タイムアウトになってデータが無効になります。したがって、スキャンチェーン全体の最低クロック周波数は、 $10\text{KHz} \times \text{スキャンチェーン内のJTAGノード数} \times 2$ 以上が必要です。スキャンチェーンの動作周波数は数十MHzなので、通常はこの問題は発生しません。ただし、多数のデバイスまたは大型のデバイスを組み合わせて使用する場合、スキャンチェーンが大きくなりすぎて、PMbusの最低速度の条件を満たせない可能性があります。

この場合、JTAGチェーンを分割してノードの数を減らす、スキャンチェーンの動作周波数を上げる、オプション4Bを試すといった対策があります。

オプション4B: PSM製品のPMBusに接続された外付けデバイスを使用するJTAGプログラミング

JTAGを使用してプログラムしたいが、a) システム・ボード上にJTAGデバイスがない、b) PMbusピンがスキャンチェーン内のバウンダリ・スキャン対応ハードウェアに接続されていない、および/またはc) スキャンチェーン全体の速度が遅すぎてPMbusの10KHz以上の動作を保証できない場合は、JTAGと外付けデバイスを使用してプログラムすることができます。

Asset Intertechのソリューションは、同社のJTAGプログラマと同社の汎用JTAG-GPIOボードを組み合わせ使用します。このソリューションは、ボード上でPMbusに直接接続できるJTAGノードとして機能します。汎用JTAG-GPIOボードは、バウンダリ・スキャン機能を備えたスキャンチェーン上の単一デバイスとして機能し、同じビットバンギング手法を利用します。

これは複数のデバイス・メーカー用に複数のプログラマを使用する代わりに、1つのJTAGプログラマを使用してボードをプログラムできるので、よりコスト効率の高い手法です。

アプリケーションノート 145

オプション5:LTpowerPlayとソケット付きプログラミング・ボードを使用して組み立て前のICにプログラムする

LTpowerPlay GUIは、デバイス専用のDC####ソケット付きプログラミング・ボードを使用して、個々の組み立て前のICにプログラムすることもできます。たとえば、LTC2978にはDC1508を使用します。この方法は、(オプション1のように)LTpowerPlay GUIをターゲット・ボードに物理的に接続できない場合に有効です。この場合、個々の組み立て前のICを、ボード上に半田付けされる前にプログラムできます。

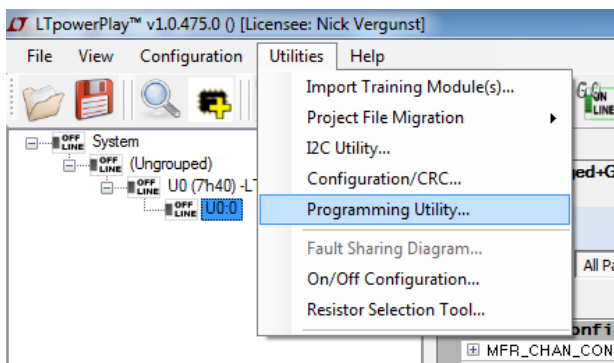
NOTE: このオプションよりも**オプション1**を強く推奨します。オプション1では、デバイスをデソルダリングせずに思いどおりに構成を変更できます。

このオプションはオプション1A「LTpowerPlay内蔵プログラミング・ユーティリティによる手動プログラミング」と同じですが、エンドユーザー・ボードの代わりに、デバイスに対応するソケット付きプログラミング・ボードを使用する点が異なります。

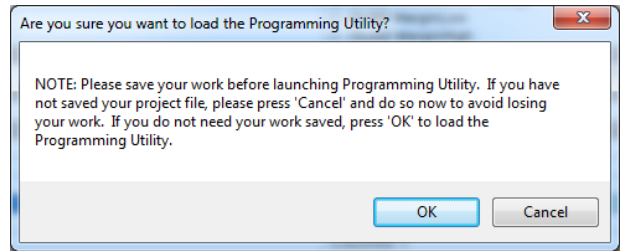
LT USB-PMBusコントローラ(DC1613A)を、ドングルからPCのUSBポートとターゲット・ボードの12ピン・ヘッダに接続します。

LTpowerPlay GUIで、以下の手順を実行します。

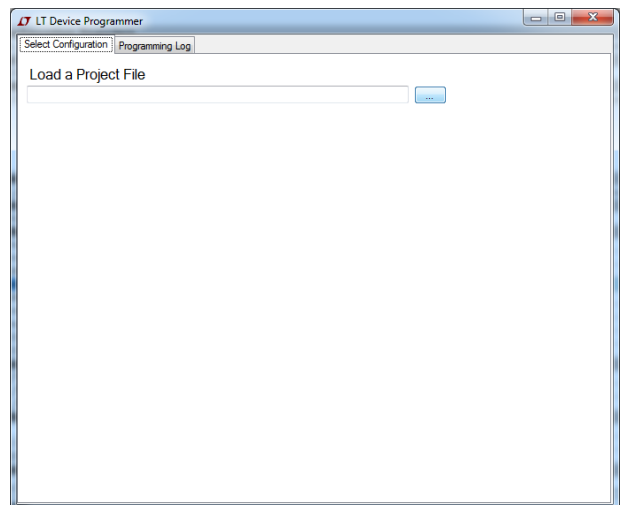
- 現在のプロジェクト・ファイルをまだ保存していない場合は、必ず保存してください。以下の手順を実行すると、現在開いているプロジェクト・ファイルは閉じられ、保存されていない変更は全て廃棄されて、プログラミング用のプロジェクト・ファイルが開かれます。
- [Utilities]->[Programming Utility]を選択します。



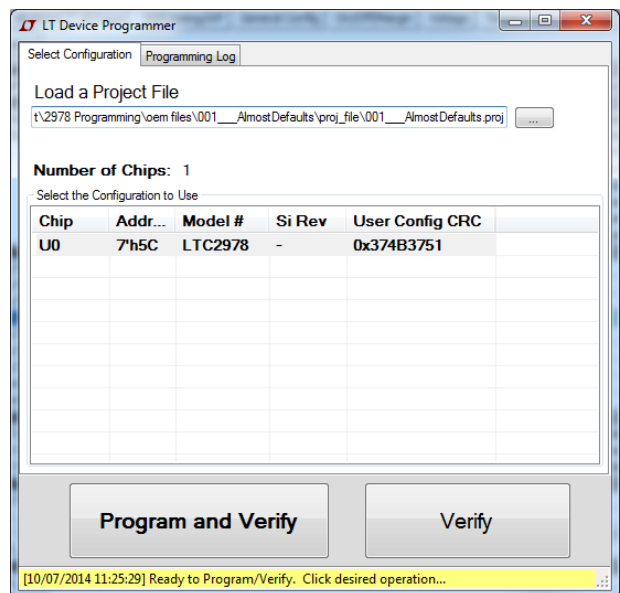
- LTpowerPlayは、現在開いているプロジェクト・ファイルを保存したかどうか確認してきます。現在開いているプロジェクトをまだ保存しておらず、ここで保存したい場合は、[Cancel]をクリックして、プロジェクトを保存してからプロセスを再開します。



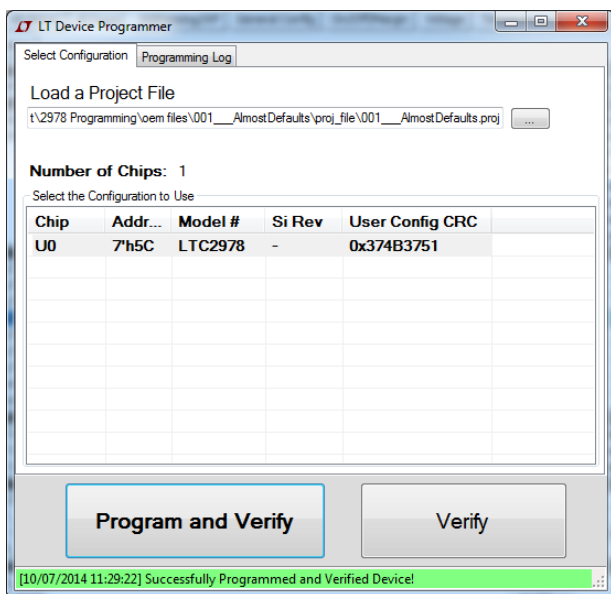
- [LT Device Programmer]ウィンドウが表示されます。[Select Configuration]タブが開いていることを確認します。[...]ボタンをクリックし、デバイスのプログラミング用のソース・ファイルとして使用したいプロジェクト・ファイルを参照します。



- *.projファイルを参照して[Open]をクリックすると、[LT Device Programmer]ウィンドウは次のように変わります。



- 画面下部に2つの大きなボタンがあります。
 - [Program and Verify] ボタンをクリックすると、デバイスのNVMを適切な状態にプログラムしてから、その内容を検証します。
 - [Verify] ボタンをクリックすると、デバイスのNVMをプログラムせずに、プロジェクト・ファイルの内容に対してNVMの内容を検証します。詳細については、付録Bを参照してください。
- ソケット内の組み立て前のデバイスをプログラムするには、[Program and Verify] をクリックします。画面最下部のステータス・メッセージが黄色になり、“Programming”、次に“Verifying”、最後に(プロセスが正常に終了した場合)“Successfully Programmed and Verified Device”と表示されます。



ここで、組み立て前のICを一度に1つずつソケットに挿入して、何も再ロードせずに[Program and Verify] ボタンを繰り返しクリックし、全ての組み立て前のデバイスをプログラムします。ICの損傷を防ぐため、ソケット内の組み立て前のICの挿入、取り外し、調整を行う前に、PCからUSB Dongleを切り離すことを推奨します。プログラミング・ボードに電源スイッチがある場合は、デバイスの取り外しや挿入を行うときに電源スイッチをオフにすることを推奨します。Dongleが接続されている間、ソケットは常に電力供給され通電しています。ソケットを確実にパワーダウンする方法は、全ての電源(一般的にはUSB Dongle)を取り外す以外にありません。

後で必要に応じてプログラム済みデバイスの検証を実行できるように、プログラミングに使用したプロジェクト・ファイルは、バージョン管理されるリポジトリに別に保存しておくことを強く推奨します。

オプション6: カスタム・ファームウェアのオーサリング

付録Aに示すように、不揮発性メモリ(NVM)に構成を格納するには複数の方法があります。1つ目の方法では、適切なI²C書き込みコマンドを使用して個々のレジスタへの書き込みを実行し、最後にSTORE_USER_ALLコマンドを発行して、この構成をNVMに格納します。全てのPMBusデバイスはこの方法をサポートしています。2つ目の方法では、バルク・プログラミング・モードでI²CからNVMへ直接書き込みを実行します。このモードでは、システムはプログラミング中も動作を続行し、(リセットまたはRESTORE_USER_ALLコマンドの後)NVMを次回読み込んだときに更新されます。バルク・プログラミング・モードの詳細とサポートについては、デバイスのデータシートを確認してください。

顧客はI²Cバスを介したPSMデバイスとの通信用の独自のファームウェアを自由に実装できます。リニアテクノロジーはサンプル・コードを提供しています。カスタム・ファームウェアは、ファームウェアのオーサリングへの投資が必要になるため、最も難しいオプションと考えられています。しかし、一度投資を行えば、そのファームウェアはデバイスおよび製品ファミリ全体にわたって再利用することができ、ボードの立ち上げ、デバッグ、通常動作の段階でデバイスとの高度な統合が可能となります。

この手法に挑戦したい方は、システム・プログラミングおよび構成のインフラ更新用にリニアテクノロジーが用意したいくつかのリソースをご検討ください。

次のWebページには、LTC Programming Hex ファイルの解析方法の詳細が掲載されています。<http://www.linear-tech.co.jp/solutions/5710>

LTC Programming Hex ファイルとオープンソースのLinduino (www.linear-tech.co.jp/Linduino) を使用して実際のインフラ・システム更新を実行する方法について詳しく説明したアプリケーション・ノートもあります。詳細については、弊社にお問い合わせください。

付録A: メモリ・アーキテクチャ

NVM互換デバイスは、構成メモリの標準PMBusモデルに従ってオペレーション・レジスタ(RAM)と不揮発性メモリの両方を実装しています。以下の図は、直接NVMアクセスなしの構成メモリ・アーキテクチャ(図6)とバルク・プログラミングによる直接NVMアクセスありの構成メモリ・アーキテクチャ(図7)を示しています。

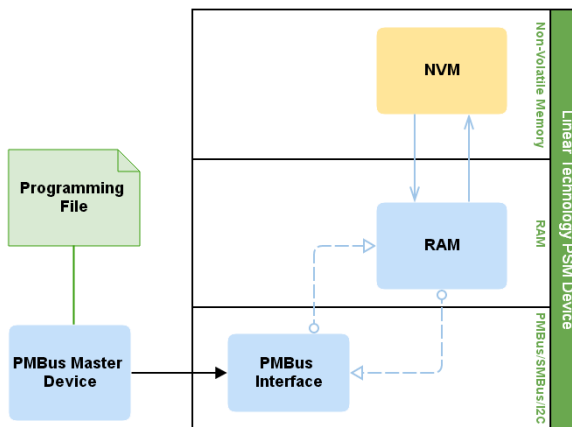


図6. LTC PSMデバイスのメモリ・アーキテクチャ (NVMアクセスなし)

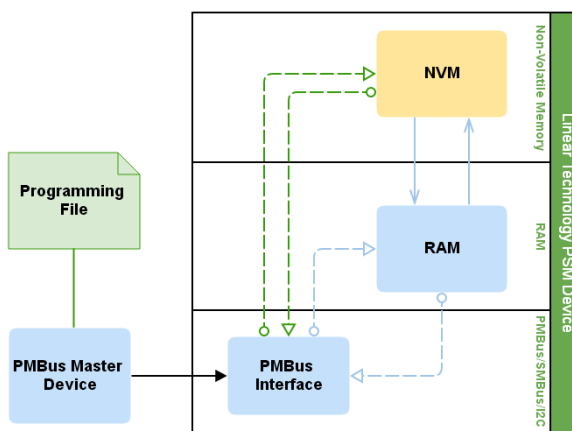


図7. LTC PSMデバイスのメモリ・アーキテクチャ (NVMアクセスあり)

オペレーションRAMには、チップの動作を制御する複数のレジスタが含まれます。これらのオペレーション・レジスタの読み出しと変更には、簡単なI²C/SMBus/PMBusの読み出し/書き込みコマンドを使用できます。これらのコマンド全てのリストは、データシートの「レジスタ・コマンド・セット」の表を参照してください。

NVM互換デバイスは、電源を一度切って入れ直すときにこれらのオペレーション・レジスタの希望の値を格納する、オンボード不揮発性メモリ(NVM)を搭載しています。

希望の構成がRAMにプログラムされたら、STORE_USER_ALL (コマンド・コード0x15)という1つのコマンドを使用して、その構成をNVMに格納できます。チップがSTORE_USER_ALLコマンドを受信すると、NVMに格納される全てのレジスタがオペレーションRAMからチップのオンボードNVMにコピーされ、永続的に格納されます。これらのレジスタ全てのリストは、データシートの「レジスタ・コマンド・セット」の表を参照してください。

デバイスの電源投入のたびに(またRESTORE_USER_ALLコマンドの受信時にも)、オンボードNVMに格納された構成値がオペレーション・レジスタに転送されます。

このアーキテクチャでは、デバイスは電源投入時に希望の構成を自律的に読み込むことができ、I²C/ファームウェアとのやり取りは不要です。

また、一部のPSMデバイスは、RAMを省略してNVMの内容を直接プログラムするバルク・プログラミング手法をサポートしています。全てのオペレーション・レジスタはRAM内にあるため、NVMに直接書き込む手法では、「舞台裏で」プログラミングが行われます。リセット/電源の入れ直しまたはRESTORE_USER_ALLコマンドによってRAMがNVMから再ロードされるまで、変更は有効になりません。システムをダウンさせずに新しい構成を読み込むインフライト更新には、この手法が便利です。

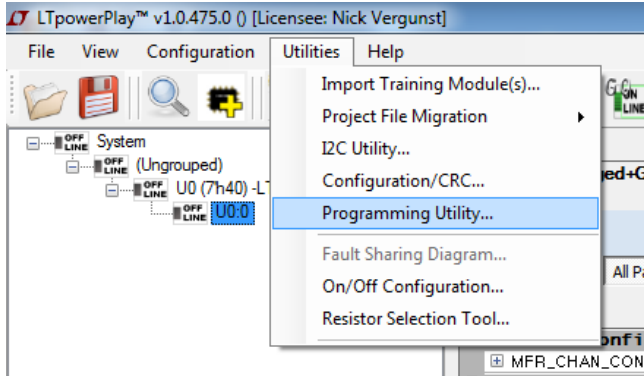
RAMまたはNVMの実際のプログラミング手法は、プログラミング・ファイル内のレシピによって決まります。詳細については、http://www.LTpowerPlay.com/in_circuit_programming/を参照してください。

付録B:LTpowerPlayを使用したプログラム済みデバイスの検証

上記のいずれかの方法によるデバイスのプログラミング後、LTpowerPlayを使用してNVMの内容を検証できます。これにはオプション1Aおよびオプション5がプログラミングに使用すると同じ機能を使用しますが、大きな違いは、[Program and Verify]ボタンではなく、大きな[Verify]ボタンを使用することです。

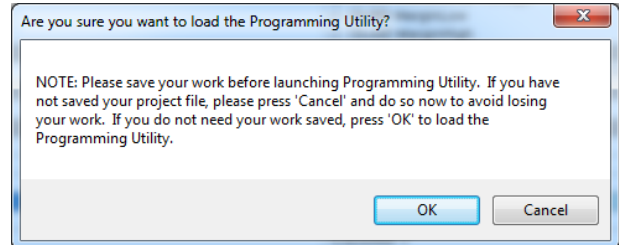
LT USB-PMBusコントローラ(DC1613A)をPCのUSBポートとターゲット・ボードに接続する必要があります。このプロセスの詳細な回路図については、図1および図2を参照してください。リニアテクノロジーでは、全てのボードにヘッダを配置することを強く推奨しています。ボード上にヘッダがない場合は、この機会にヘッダを配置してください。組み立て前のデバイスを検証する場合は、そのデバイス専用のソケット付きプログラミング・ボードを使用してください。

- 選択したバージョン管理リポジトリからプロジェクト・ファイルまたはプログラミング・アーカイブ・ファイルを取り出し、LTpowerPlayからアクセスできるように、そのファイルをローカルにコピーします。
- LTpowerPlayを開き、メニューから[Utilities]->[Programming Utility]に進みます。

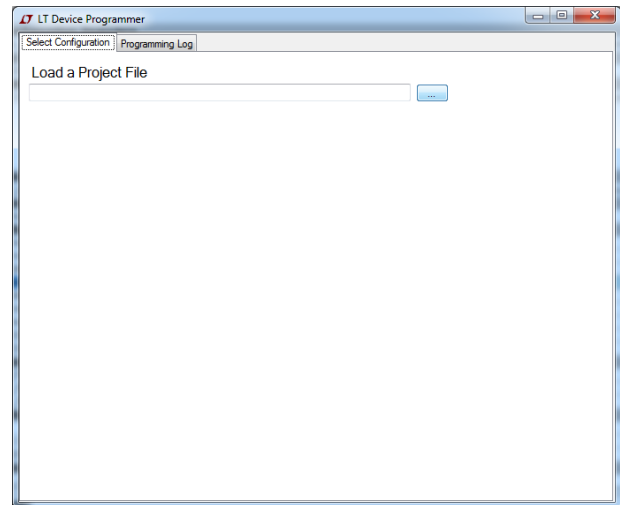


- 現在開いているプロジェクト・ファイルがまだ保存されていない場合、LTpowerPlayは、現在開いているプロジェクト・ファイルを保存したかどうか確認します。以下の手順を実行すると、現在開いているプロジェクト・ファイル

は閉じられ、保存されていない変更は全て廃棄されて、プログラミング用のプロジェクト・ファイルが開かれます。LTpowerPlayを起動したばかりであれば、この警告は無視してかまいません。現在開いているプロジェクトをまだ保存しておらず、ここで保存したい場合は、[Cancel]をクリックして、プロジェクトを保存してからプロセスを再開します。

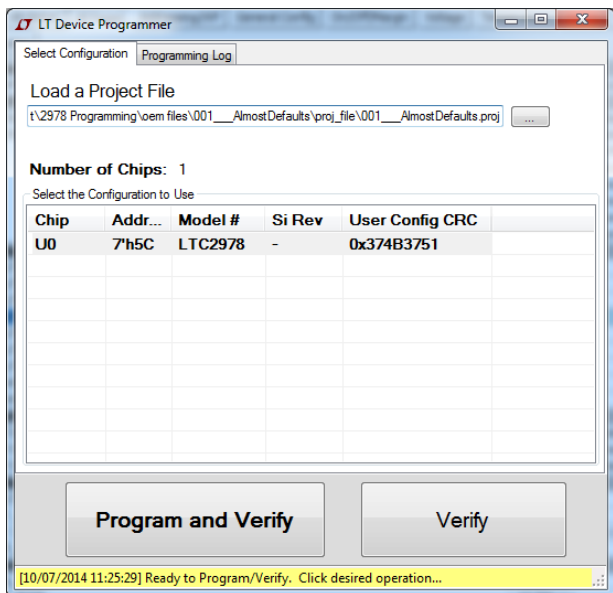


- [LT Device Programmer]ウィンドウが表示されます。[Select Configuration]タブが開いていることを確認します。[...]ボタンをクリックし、デバイスのプログラミング用のソース・ファイルとして使用したいプロジェクト・ファイルを参照します。プログラミング・ファイルの*.zipアーカイブを保存していた場合は、そのアーカイブをフォルダに展開する必要があります。展開されたフォルダには、そのアーカイブに対応するプロジェクト・ファイルがあります。プロンプトが表示されたらこのファイルを使用します。

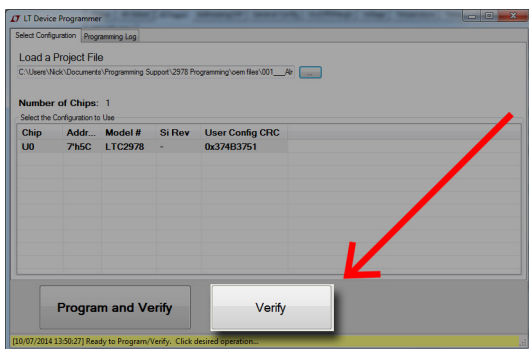


アプリケーションノート 145

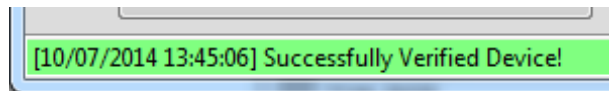
- *.proj ファイルを参照して[Open]をクリックすると、[LT Device Programmer]ウィンドウは次のように変わります。



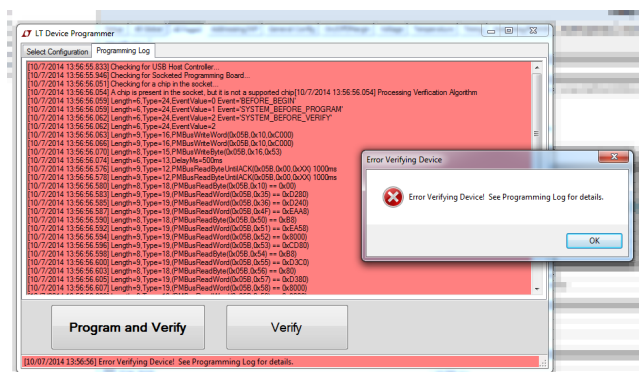
- 画面下部に2つの大きなボタンがあります。
 - [Program and Verify]ボタンをクリックすると、デバイスのNVMを適切な状態にプログラムしてから、その内容を検証します。
 - [Verify]ボタンをクリックすると、デバイスのNVMをプログラムせずに、プロジェクト・ファイルの内容に対してNVMの内容を検証します。
- ここでは不揮発性メモリの内容を検証するだけなので、[Verify]ボタンのみを使用します。



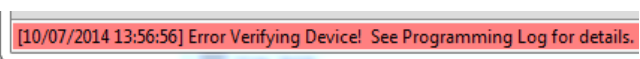
- [Verify]をクリックすると、LTpowerPlayは自動的に、プロジェクト・ファイル内の希望の値に基づいてデバイス内の全ての不揮発性メモリを徹底的に検証します。
- 検証が正常に終了した場合、LTpowerPlayは“Successfully Verified Device!”メッセージを表示します。



- 検証に失敗した場合、ポップアップ・ウィンドウにエラー・メッセージが表示され、エラーの詳細を示す赤いウィンドウが表示されます。このテキストを外部のファイルにコピーして、ログとして保存できます。



- ウィンドウの下部には別のステータス・メッセージが表示されます。



どの手法でプログラミングした場合でも、この方法を使えば、素早く簡単にデバイスのプログラミングを検証できます。リニアテクノロジーが承認している全ての手法には、プログラミング・プロセスそれ自体に不可欠な手順として、この検証チェックが含まれています。

このプロセスは、開発技術者の不安を解消し、サードパーティ・ソフトウェア・ハウスや製造請負業者が発注したとおりの製品を納品したこと、(更に重要なことに)デバイスを取り違えていないことを検証するのに便利です。