

Application Brief

TI のプログラマブル ロジック デバイスで、ウェアラブル エレクトロニクスを最適化



機能ブロック図

TI のプログラマブル ロジック デバイス (TPLD) は、より小型でより安価なウェアラブル エレクトロニクスの実現を目指す設計者をサポートします。TPLD ファミリーは、複数のディスクリート コンポーネントを 1 つの小型フォーム ファクタのパッケージに統合しているため、設計者は部品表 (BOM) と基板の簡素化と、サイズ削減を実現できます。マイクロコントローラ (MCU) と異なり、TPLD はロジック デバイス並みにスタンバイ消費電力が低く、またアナログ コンポーネントの統合も可能です。

TPLD はウェアラブル エレクトロニクスで複数の機能を実行できます。以下のブロック図で例をいくつか示します。TPLD は単に MCU を置き換えるのではなく、1 つの MCU から機能の一部をオフロードでき、必要とされる MCU のサイズ、複雑さ、コストを削減できます。

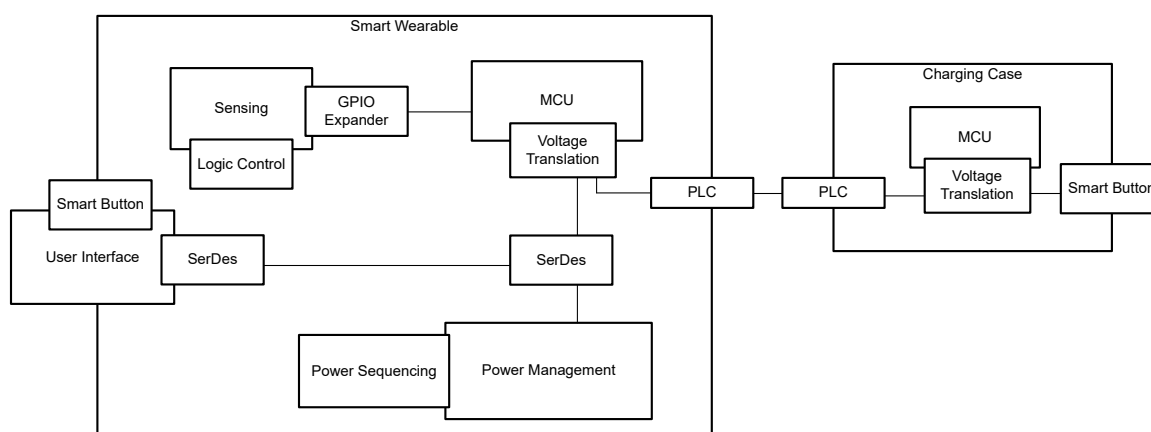


図 1. ウェアラブル エレクトロニクスの概略ブロック図

ウェアラブル エレクトロニクスにおける TPLD アプリケーション

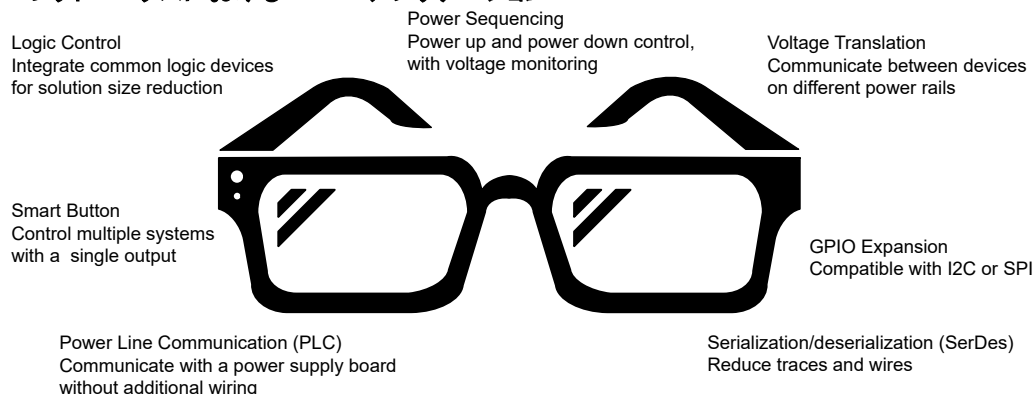


図 2. TPLD アプリケーションの例

電力線通信 (PLC)

充電ユニットとスマート デバイス間の通信は通常、個別の電力ラインと通信ラインを必要とします。TPLD を採用した PLC により、通信ラインが不要になり、複雑さとサイズを低減できます。既存の専用 PLC 集積回路 (IC) と比較して、TPLD はカスタム システムに統合できる構成可能な設計を実現しており、MCU も不要です。

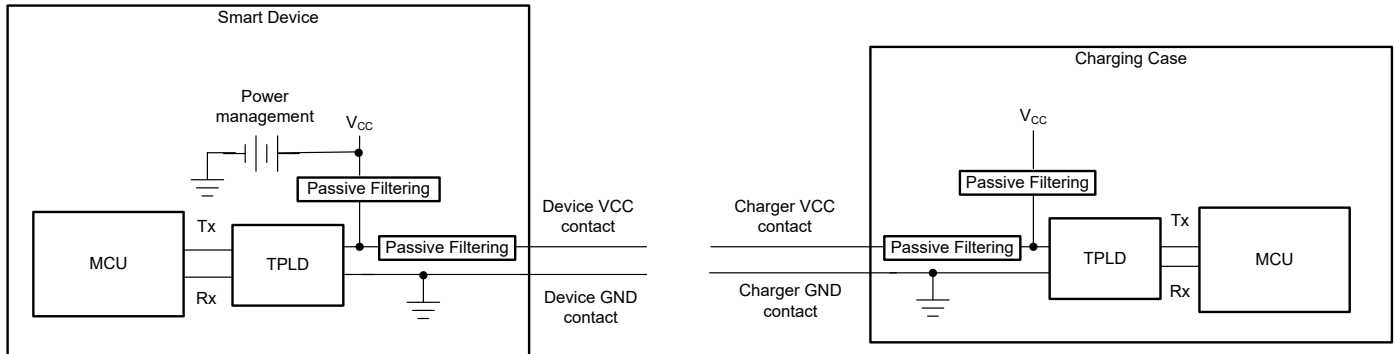


図 3. PLC のブロック図

TPLD を使用する PLC の詳細については、TI のプログラマブル ロジック デバイス (TPLD) の半二重電力線通信を参照してください。

シリアライゼーション / デシリアライゼーション (SerDes)

スマートグラスのように広がった基板を持つデバイスでは、システムの一側の側から他方の側に複数の配線を接続すると、設計全体のサイズが増大します。TPLD を使用すると、最大 8 つの平行入力信号を単一の出力にシリアル化して、1 本の配線にて送信し、基板の反対側にある別の TPLD がデシリアライズします。つまり TPLD を使用すると、設計サイズと配線の複雑さを低減できます。

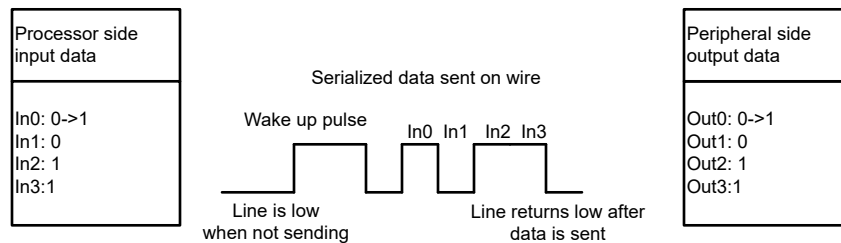


図 4. SerDes 機能

TPLD を採用すると、SerDes システムの両側でカスタマイズ可能な入出力 (IO) 数を 0 ~ 8 個の入力の範囲で実現でき、MCU や他の高度なデバイスを必要とせず、入力信号の変化に自動的に対応します。

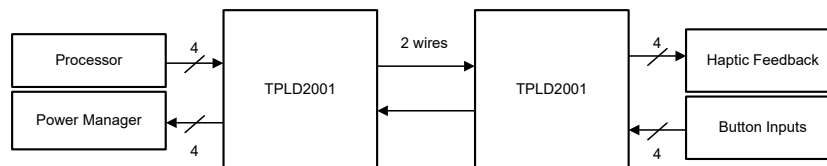


図 5. SerDes のブロック図

電源シーケンスと電源監視

多くの設計では、電力への逆流や意図しないデバイス動作を防止する、電源シーケンスが必要となります。TPLD は、エラー検出機能付きのシングルチップ電源オン / オフ制御機能を備えています。TPLD は整合性が高く、抵抗コンデンサ (RC) ネットワーク設計よりも高度な制御が可能で、MCU 設計よりも高速で起動時の消費電力を低減できます。TPLD の

内部発振器を使用すると、外部クロック信号を使用せずにタイミング制御ができます。また内蔵のアナログ コンパレータにより、電圧の監視や電源シーケンスのゲーティングを実行できます。

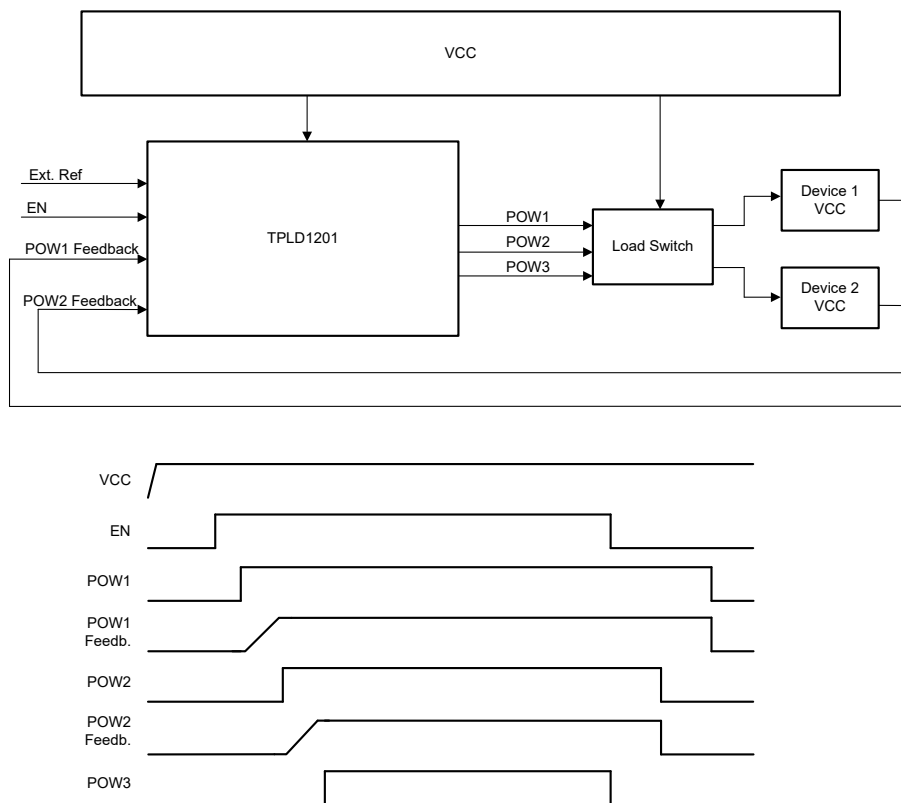


図 6. 電源シーケンス機能

TPLD による電源シーケンスの詳細については、TI のプログラマブル ロジック デバイスによるフィードバック付き電源シーケンスのアプリケーション概要を参照してください。

汎用入出力 (GPIO) の拡張

MCU からペリフェラルを制御するのに複数のピンを専用で使用すると、必要な MCU のコストと複雑さが増大します。TPLD GPIO エクスパンダは、シリアル ペリフェラル インターフェイス (SPI) または MCU からの I2C (inter-integrated circuit) を使用して、GPIO 出力を制御できます。SPI または I2C を使用して、最大 8 入力と 8 出力を読み書きできます。I2C または SPI ブロックは、他のリソースと完全に独立して未使用ピンの読み書きに使用できるため、GPIO の拡張を TPLD 内の他の機能と組み合わせることができます。このため専用の GPIO エクスパンダ チップが不要になり、BOM サイズと設計サイズを削減できます。

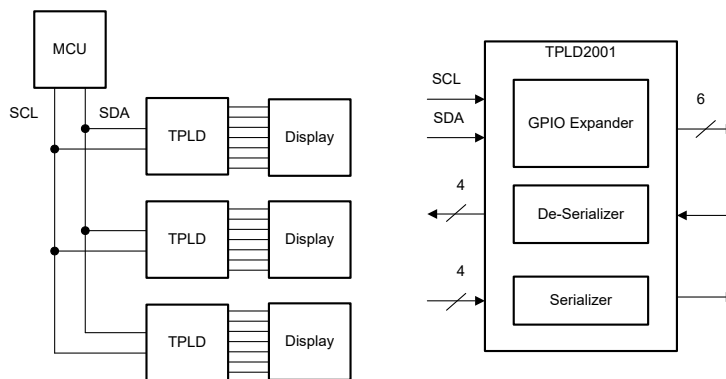


図 7. GPIO の拡張の例

TPLD を使用した GPIO 拡張の例については、TI のプログラマブル ロジック デバイスの I2C GPIO 拡張とステート マシンを使用した 7 セグメント ディスプレイの制御のアプリケーション 概要を参照してください。

スマート ボタン

ウェアラブル デバイスはサイズの小型化が一層進んでおり、ボタンなどのユーザー用スペースが減少しています。TPLD を使用すると、単一の入力信号を使用してボタンの押下パターンを解釈し、さまざまな動作を実行できるため、デバイスの制御性を損なわずに必要なユーザー側入力の数を削減できます。TPLD はボタンのデバウンスや電圧変換もでき、多様な構成が可能な一連の制御方式にも対応しています。

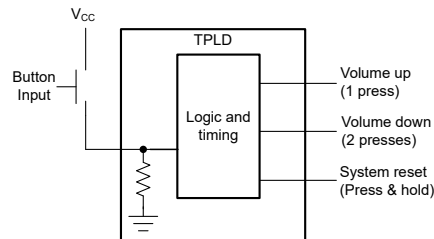


図 8. スマート ボタンの例

スマート ボタン制御方式の例は、プログラマブル ロジックでのステート マシンの使用のアプリケーション ノートで紹介されています。

ロジック制御

小型のワンオフ ロジック エLEMENTは、多くの場合、パワーグッド通信や信号リドライビングなどの機能に必要です。これらのロジック エLEMENTを、他の機能を実行する 1 個の TPLD に統合して、複数の別部品の必要性を排除できます。すべての TPLD ブロックは互いに独立して動作するため、複数の機能を並列に実行できます。ロジック コンポーネントを単一のデバイスに統合して、設計全体のサイズと消費電力を削減します。

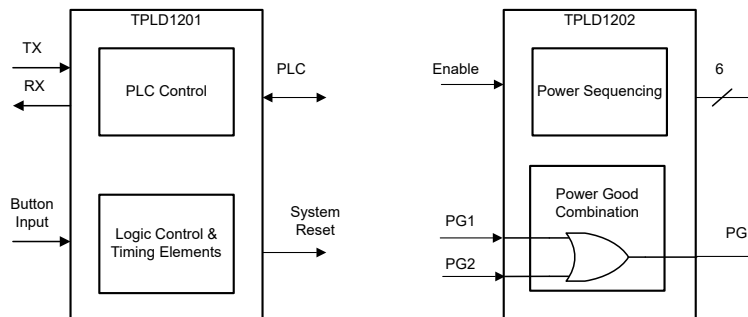


図 9. ロジック制御の例

電圧変換

MCU は多くの場合、周辺 デバイスより低い電圧レベルで動作し、通信を実現するために他の IC を必要とします。TPLD はオプションの低電圧レベル入力とオープン ドレイン出力を備えており、昇圧や降圧の変換が可能です。GPIO 拡張やロジック制御などの別の目的のために、MCU から TPLD を介してパターンを配線すると、専用の電圧変換 IC を使用せずに電圧変換が可能になり、設計サイズと消費電力を低減できます。

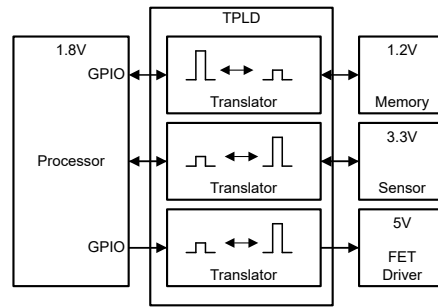


図 10. 電圧変換の例

昇圧や降圧電圧変換に TPLD を使用する方法の詳細については、[TI のプログラマブル ロジック デバイス \(TPLD\) の昇圧および降圧変換のアプリケーション概要](#)を参照してください。

商標

すべての商標は、それぞれの所有者に帰属します。

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日 : 2025 年 10 月