

Tektronix[®]

組込みシステム設計における シリアル・バスのデバッグ

アプリケーション・ノート



概要：

強力なトリガ、デコード、サーチ機能を備えたテクトロニクスのオシロスコープを使用し、優れた効率で組込みシステム設計の問題を解決する方法を説明します。

はじめに

組込みシステムは、今日ではいたるところで目にすることができます。組込みシステムを簡単に定義すると、大型システム/機械をモニタ、制御するために、その一部として組込まれた、特殊用途のコンピュータ・システムと言えます。一般的な組込みシステムでは、電源を入れるとただちに特別なアプリケーションが実行され、電源がオフになるまで停止することはありません。今日では、ほとんどすべての電子デバイス、製品が組込みシステムであると言えます。組込みシステムの簡単な例を次に示します。

- 目覚まし時計
- ATM (現金自動預払機)
- 携帯電話
- コンピュータ用プリンタ
- アンチロック・ブレーキ・コントローラ
- 電子レンジ
- ミサイル慣性誘導システム
- DVD プレーヤ
- PDA (携帯情報端末)
- 工業用自動化・モニタ用プログラマブル・ロジック・コントローラ (PLC)
- 携帯音楽プレーヤ
- トースタもその可能性があります。

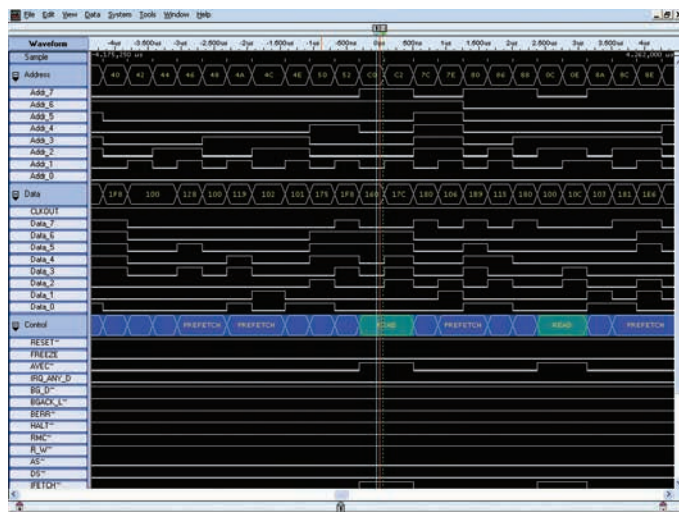


図1. マイクロコントローラのクロック、アドレス・バス、データ・バス、コントロールラインをロジック・アナライザで取込んだ例

組込みシステムには、マイクロプロセッサ、マイクロコントローラ、DSP、RAM、EPROM、FPGA、A/D、D/A、I/Oなどの異なったタイプのデバイスが使用されています。従来、これらのデバイスにおいては、デバイス間、外部との通信にパラレル・バスが使用されてきました。しかし、最近では、組込みシステム設計で使用される構造ブロックが、以下のような理由によりパラレル・バスからシリアル・バスに移行しつつあります。

- 配線数が少ないため、基板を小さくすることができる
- ロー・コスト
- 低電力
- パッケージのピン数が少ない
- エンベデッド・クロック
- 差動信号による優れた雑音耐性
- 標準規格のシリアル・インタフェースを使った豊富な部品群

シリアル・バスには優れた利点が数多くありますが、パラレルではなくシリアルで転送されるため、組込みシステムの設計エンジニアは解決すべき問題点も数多く持つことになります。このアプリケーション・ノートでは、組込みシステム設計エンジニアが直面する課題について説明し、5シリーズMSO、MSO/DPO70000シリーズ、DPO7000シリーズ、MSO/DPO5000シリーズ、MDO4000シリーズ、MDO3000シリーズ、MSO/DPO2000シリーズ・オシロスコープを使用して解決する方法について説明します。

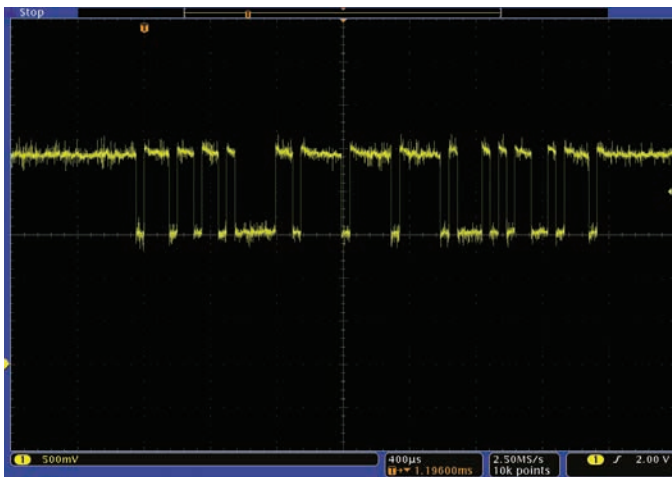


図2. CANバス信号を取込んだ例

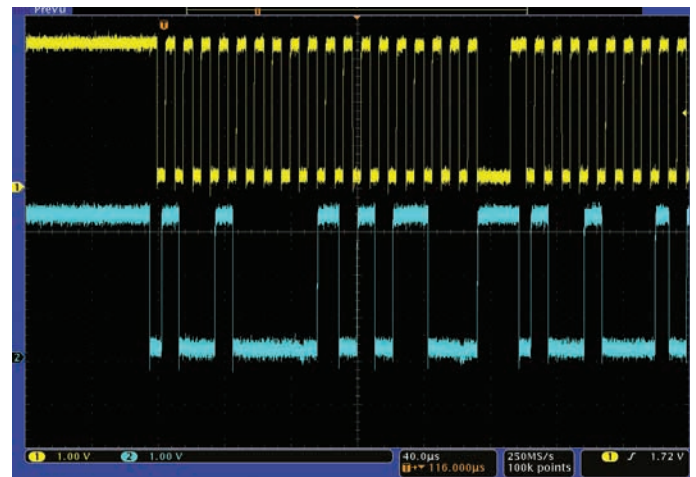


図3. PCバス信号を取込んだ例

パラレルとシリアル

パラレル構造では、バス上の各コンポーネントは、16本のアドレス・ライン、16本のデータ・ライン、クロック・ラインやその他のコントロール信号など、独自の信号パスを持ちます。バス上に送られるアドレスやデータは、すべてのパラレル・ラインに同時に転送されます。このため、オシロスコープやロジック・アナライザのステート・トリガまたはロジック・トリガを使用すれば、観測したいイベントに簡単にトリガすることができます。また、オシロスコープやロジック・アナライザに表示されるデータは、容易に理解できます。

例えば、図1は、ロジック・アナライザを使用して、マイクロコントローラのクロック、アドレス、データ、コントロール・ラインの信号を取込んだ例を示しています。ステート・トリガを使用することで、測定したいバス転送のみを取込むことができます。

バス上の動きは、各アドレス、データ、コントロール・ラインの論理状態を見るだけで理解できます。シリアル・バスでは、すべての情報は、数少ない（場合によっては1本の）ケーブルでシリアルに転送しなければなりません。これは、一本の信号に、アドレス、コントロール、データ、クロック信号が含まれることを意味します。CAN(Contoller Area Network)シリアル信号の例を、図2に示します。

このメッセージには、フレームの開始点、識別子(アドレス)、データ長コード、CRC、フレームの終了点、さらにコントロール・ビットも含まれています。さらに複雑なことに、クロックはデータに組込まれ、クロックを受信するデバイスがクロックにロックできるよう、適切なクロック・エッジのためのビット・スタッフィングが使用されています。このような信号に見慣れた人であっても、このメッセージの内容を理解することは容易ではありません。このメッセージに不良があり、一日に一回しか発生せず、これにトリガする必要があるとしたらどうでしょう。従来のオシロスコープやロジック・アナライザは、このような信号を取込むことには適していません。

PCなどの簡単なシリアル規格であっても、パラレル・プロトコルに比べると、バス上で転送されている信号の意味を理解することは容易なことではありません。

PCでは、クロックとデータ・ラインは別になっていますので、この例では、クロックを基準ポイントとすることができます。しかし、メッセージの開始点(クロックがハイのときに、データはローになる)を見つけ、各クロックの立上りエッジにおけるデータを書き留め、メッセージ構造になるように各ビットをまとめる必要があります。

ロング・メモリに取込んだ膨大なデータの中から一つのメッセージをデコードするだけでも数分は必要であり、このメッセージが本当に探していたものかどうかもわかりません。探していたものでない場合は、次のメッセージに対して、再び面倒で間違いやすい作業を行わなければなりません。探しているメッセージのみにトリガできればよいのですが、従来のオシロスコープやロジック・アナライザのステート・トリガ、パターン・トリガではトリガをかけることはできません。これらのトリガ機能は、複数のチャンネルで同時に起こっているパターンを検出するために設計されています。シリアル・バスで機能させるためには、何千というステート(1ビットに1ステート)を持つハードウェアのトリガが必要になります。たとえこのようなトリガ機能があったとしても、これらすべてのビットをステートごとにプログラムするのは容易な作業ではありません。もっと簡単な方法はないのでしょうか。

当社のオシロスコープ*1を使用することで、この問題を解決します。以下では、組込みシステム設計で使用される、一般的なロー・スピード・シリアル規格に対するMSO/DPOシリーズのソリューションを説明します。

*1 オシロスコープによってサポート可能なシリアル・バス規格は異なります。詳細については、付録Aまたは当社ウェブ・サイト(www.tektronix.com/ja)をご覧ください。

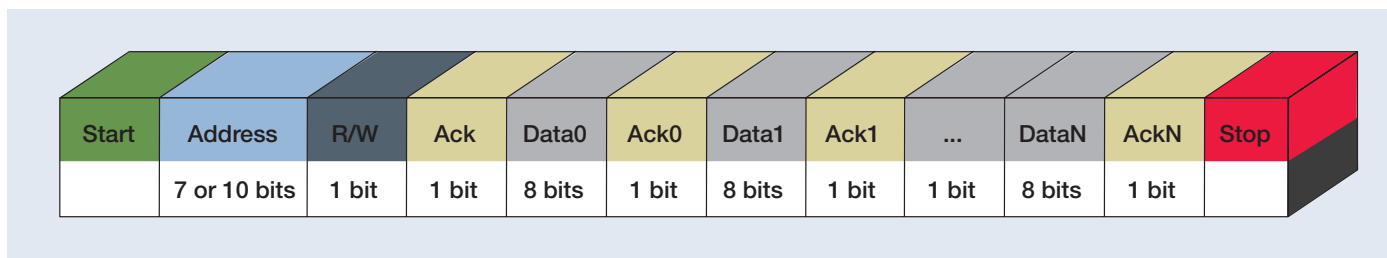


図4. I2Cのメッセージ構成

I2C

概要

I2C (アイスクエアシー) は、Inter Integrated Circuitの略です。1980年代初期にPhilips社により、テレビのコントローラと周辺機器を接続するための低価格な方法として開発されました。今では、組込みシステムのデバイス間の通信における世界標準規格として発展しています。単純な2線式であり、Analog Devices社、Atmel社、Infineon社、Cyprus社、Intel社、Maxim社、Philips社 Silicon Laboratories社、ST Microelectronics社、Texas Instruments社、Xicor社などの主要半導体メーカーによって、I/O、A/D、D/A、温度センサ、マイクロコントローラ、マイクロプロセッサなどの広範囲なチップに採用されています。

動作原理

I2Cは、物理的には2線式のインタフェースで、双方向シリアル・クロック (SCL) とデータ (SDA) で構成されています。バス上の複数のマスタとスレーブをサポートしますが、同時には一つのマスタのみがアクティブになります。任意のI2Cデバイスがバスに接続でき、任意のマスタ・デバイスとスレーブ・デバイスで情報を交換できます。各デバイスはユニークなアドレスで認識され、デバイスの機能によって送信、受信のいずれかとして機能します。当初は7ビットのアドレスでしたが、10ビットのアドレスも使用できるようになりました。100kbps (標準モード)、400kbps (ファースト・モード)、3.4Mbps (高速モード) のビット・レートがサポートされています。

最大デバイス数は、システムのバス負荷容量の上限値400pFから、おおよそ20~30のデバイスが接続できます。

図4のI2C規格のフォーマットを以下に記します。

- Start — 通信の最初を示す信号で、この後にメッセージが続きます。
- Address — 7ビットまたは10ビットによるデバイスのアドレスで、読取りまたは書込みで使用されます。
- R/Wビット — 1ビット。データがデバイスから読取られるのか、デバイスに書込まれるのかを示します。
- Ack — 1ビット。マスタのアクションをスレーブ・デバイスが正しく受信したことを示します。通常、各アドレス、データ・バイトはAckを持ちますが、常に持つとは限りません。
- Data — 整数のバイトでデバイスの読取り、書込みを行います。
- Stop — メッセージの終了を示し、マスタはバスを解放します。

I2Cのアドレスをデコードする方法には、7ビット+リードまたはライト (R/W) ビット、またはR/Wビットがアドレスに含まれる8ビットの方法があります。7ビット・アドレスの方法は、ファームウェアおよびソフトウェアの設計エンジニアが従うべきI2C規格で規定されています。しかし、その他多くのエンジニアは8ビット・アドレス法を使用しています。MSO/DPOシリーズ・オシロスコープは、どちらの方法でもデータをデコードすることができます。

I²C デバッグのポイント

DPOxEMBD型シリアル・トリガ/解析アプリケーション・モジュールまたはOpt. SR-EMBDを装備したテクトロニクスのおシロスコープは、I²Cバスに携わる組込みシステム設計エンジニアの強力なツールになります。前面パネルにはB1とB2の2つのバス・ボタンがあり、入力が定義できます。I²Cバスの設定メニューを図5に示します。

クロックとデータのチャンネル、ロジック1と0のスレッシュホールド電圧を設定するだけで、バスで転送されるプロトコルなどをオシロスコープで観測することができます。簡単な設定をするだけで、特定のメッセージ・レベルでの情報でトリガをかけ、取込んだ結果をデコードすることで、結果が容易に理解できます。エッジでトリガし、測定したいイベントであることを期待しながらメッセージを手作業でデコードすることは、もはや必要ありません。

図6に、組込みシステムの例を示します。I²Cバスには、CPU、EEPROM、ファン・スピード・コントローラ、D/Aコンバータ、温度センサなどが接続されています。

この機器は、常に高温になり、電源が切れてしまうことにより、故障解析のために設計部門に送られてきました。最初に行うことは、ファン・コントローラとファンのチェックです。しかし、どちらも正常に動作しているようです。次に、温度センサに異常がないかチェックします。ファン・スピード・コントローラは、機器内の異なる場所にある2つの温度センサから定期的に情報を集め、内部温度を一定に保つようにファン・スピードを調整します。どちらか、または両方の温度センサが正しく機能していないのではないかと疑っています。センサとファン・スピード・コントローラの通信を確認するために、オシロスコープのプローブをI²Cのクロックとデータ・ラインに接続し、バス解析機能の設定をします。2つの温度センサのアドレスは、I²Cバスの18と19とわかっていますので、アドレス18への書き込み（ファン・スピード・コントローラは、現

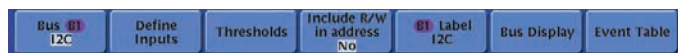


図5. I²Cバス・サポートの設定メニュー

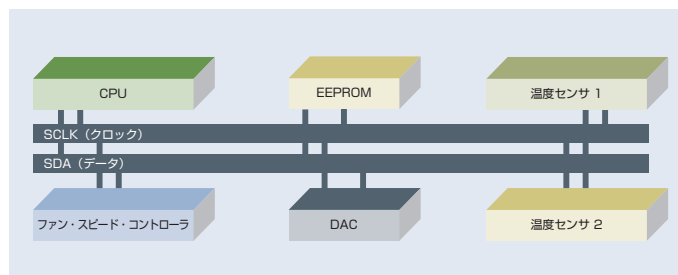


図6. I²Cバスの例

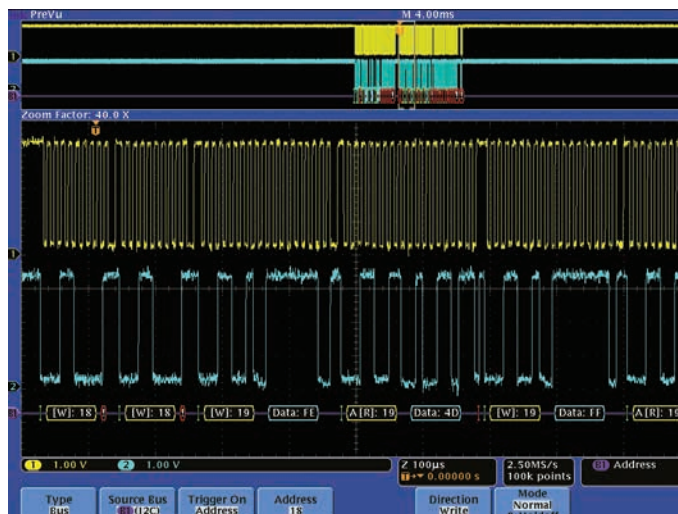


図7. I²Cのアドレスとデータ・バスの表示例

在の温度を収集します) イベントでトリガするように設定します。結果を図7に示します。

バスの状態	記号
開始点は緑の垂直バーで示されます。 前回の停止なしに別の繰り返し開始が表示された場合は、繰り返しの開始が発生します	
アドレスは、黄色の囲みで、ライトは白で [W] の後に、リードは赤で [R] の後に表示されます。 値は 16 進またはバイナリで表示されます。	
データは、青の囲みで表示されます。 値は 16 進またはバイナリで表示されます。	
Ackなしは、赤い四角の中に！マークで表示されます。	
停止は、赤の垂直バーで表示されます。	

表1. バスの状態

この例では、Ch1 (黄) はSCLKに、Ch2 (青) はSDAに接続されています。紫色の波形は、わずかなパラメータの入力で定義したI2Cバスです。上のエリアでは、取込んだすべての波形が表示されています。下の大きなエリアでは、ズーム波形が表示されています。ご覧のように、バス上の各メッセージの内容がデコードされています。メッセージの重要な部分を解り易く表示するために、表1のようなカラー／マークが使用されています。

取込まれた波形から、オシロスコープは確かにWrite to address 18 (図7の左下) でトリガされています。実際、ファン・スピード・コントローラはアドレス18に2回書き込もうとしています、温度セン

サへの書き込みに対するAckを受け取っていません。次にアドレス19の温度センサをチェックしていますが、期待した情報が返っています。なぜ、最初の温度センサの応答がファン・コントローラに返っていないのでしょうか。回路基板を確認したところ、アドレス・ラインの一つが正しくハンダ付けされていませんでした。温度センサはバスと通信できず、結果として機器が過熱していました。オシロスコープのI2Cトリガ機能、デコード機能により、検出が難しい問題をわずか数分で特定することができました。

図7の例では書き込みでトリガしましたが、MSO/DPOシリーズのI2Cトリガには、この他にも以下に示すような機能があります。

- Start – SCLがハイの状態でもSDAがローになるとトリガします。
- Repeated Start – 前のストップ・コンディションが発行されないでスタート・コンディションが発行される場合にトリガします。通常、バスを開放することなく、マスタが複数のメッセージを転送する場合に適用されます。
- Stop – SCLがハイの状態でもSDAがハイになるとトリガします。
- Missing Ack – スレーブは、アドレス、データの各バイトの後にアクノリッジを転送するよう構成されています。スレーブがアクノリッジ・ビットを戻さない場合にオシロスコープはトリガします。
- Address – ユーザが設定するアドレス、またはGeneral Call、Start Byte、HS-mode、EEPROM、CBUSを含む、あらかじめプログラムされた特定アドレスにトリガします。アドレスは7ビットまたは10ビットで、バイナリまたは16進で入力します。
- Data – バイナリまたは16進で設定する12バイトまでの任意のデータにトリガします。
- Address and Data – アドレスとデータ、リード対ライトの両方が入力でき、測定するイベントを正確に取込むことができます。

これらのトリガによりバスを特定することができ、また取込んだ波形データから、転送されるすべてのメッセージの内容をただちにデコードすることができます。

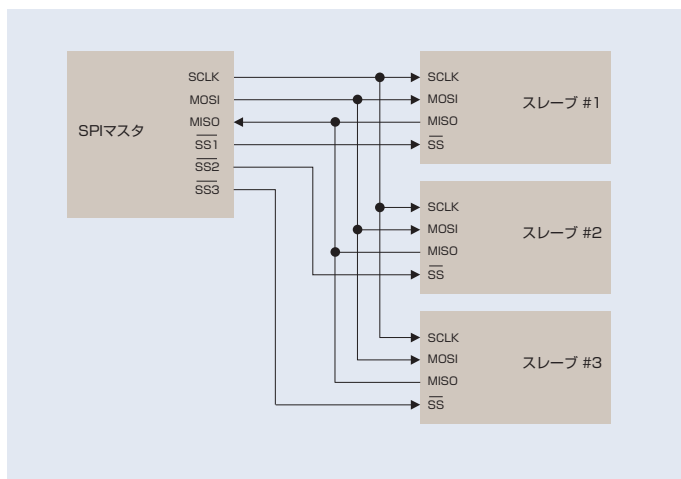


図8. 一般的なSPIの構成

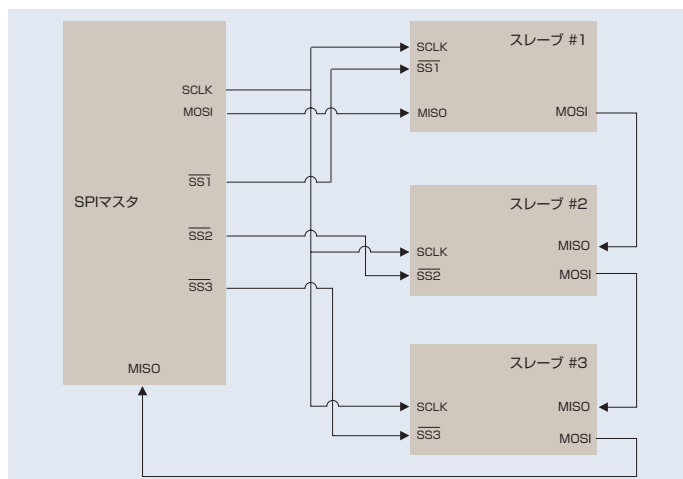


図9. デイジー・チェーン接続されたSPIの構成

SPI

概要

SPI (Serial Peripheral Interface) は、1980年代後半に、モトローラ社によって68000シリーズ・マイクロコントローラのために開発されました。バスが簡潔で一般性があるため、数多くの製造メーカが採用してきました。今では、主に組込みシステム設計で使用される広範囲な部品に見受けられます。SPIは、主にマイクロコントローラと、直接に接続される周辺デバイス間で使用され、携帯電話、PDAおよび携帯機器において、CPU、キーボード、ディスプレイ、メモリ・チップ間の通信で使用されています。

動作原理

SPIは、マスタ/スレーブ方式の4線式シリアル通信バスです。クロック (SCLK)、MOSI (Master Output/Slave Input)、MISO (Master Input/Slave Output)、スレーブ・セレクト (SS) の4つの信号からなります。2つのデバイスが通信する場合は、常に一方が「マスタ」になり、もう一方が「スレーブ」になります。マスタがシリアル・クロックをドライブします。データは同時に送受信され、全二重プロトコルです。バス上のデバイスごとにユニークなアドレスを持つのではなく、SSラインにより送受信されるデバイス・データを指定します。したがって、それぞれのデバイスは、マスタから独自のSSを割り当てられます。3つのスレーブ・デバイスがある場合は、マスタから3つのSSが発行されます (図8を参照)。

図8において、各スレーブはマスタのみに送信します。しかし、図9に示すように、SPIではスレーブ・デバイスはデイジー・チェーン接続され、各操作と実行は交互に行われ、結果はマスタに戻ります。

したがって、SPIの実装には「標準」がありません。場合によってはスレーブからの返信がマスタに届く必要がなく、そのような場合、MISO信号は省略されます。その他の場合ではマスタは一つ、スレーブも一つですので、SSはまとめてグランドされます。これを一般に2線式SPIと呼びます。

SPIデータ転送が実行されると、8ビット・データ・ワードはMOSIからシフトされる一方、別の8ビット・データ・ワードがMISOにシフトされます。これは、16ビットの循環シフト・レジスタと見ることができます。転送が実行されると16ビット・シフト・レジスタは8ビット分シフトし、マスタとスレーブ・デバイス間で8ビット・データが交換されます。二つのレジスタ、CPOL (クロック極性)、CPHA (クロック位相) により、データが扱われるクロックのエッジを決定します。各レジスタは2種類のステートを持ち、4種類の組み合わせがありますが、互換性はありません。したがって、マスタとスレーブが通信するためには、同じパラメータを使用する必要があります。複数のスレーブが異なった構成で固定されている場合、異なったスレーブと通信するためには、マスタはその都度再構成する必要があります。

SPIデバッグのポイント

前面パネルのBusボタンを使用することで、SPIバスと同様の機能を持つことができます。SCLK、SS、MOSI、MISOのチャンネル、スレシヨルド電圧、極性などの基本パラメータを設定します(図10を参照)。

ここでは、図11に示す組込みシステムを例に説明します。

SPIバスには、シンセサイザ、DAC、I/Oが接続されています。シンセサイザにはVCOが接続され、これからシステムに2.5GHzのクロックが供給されます。シンセサイザは、起動時にCPUによってプログラムされることになっています。しかし、正しく動作していないようで、VCOは3GHzを発生してスタックします。この問題のデバッグでは、まずCPUとシンセサイザ間で信号があるか、また、物理的な接続に問題がないかを確認します。しかし、どこにも問題はないようです。次に、シンセサイザをプログラムするための、SPIバスで実際に転送されている情報を観測します。この情報を取込むため、シンセサイザのSS (Slave Select) 信号がアクティブになってDUTを起動するところでトリガするよう、オシロスコープを設定します。取込まれた信号を図12に示します。

Ch1(黄)がSCLK、Ch2(青)がMOSI、Ch3(赤)がSSです。デバイスが正しくプログラミングされているか確認するため、シンセサイザのデータ・シートを確認します。バス上の最初の3つのメッセージは、シンセサイザの初期化、分周比のロード、データのラッチとなっています。仕様によると、最初の3つの転送の最後のニブル(16進の一字)はそれぞれ、3、0、1となっていなければなりません。しかし、実際には、0、0、0となっています。



図10. SPIバス・サポートの設定メニュー

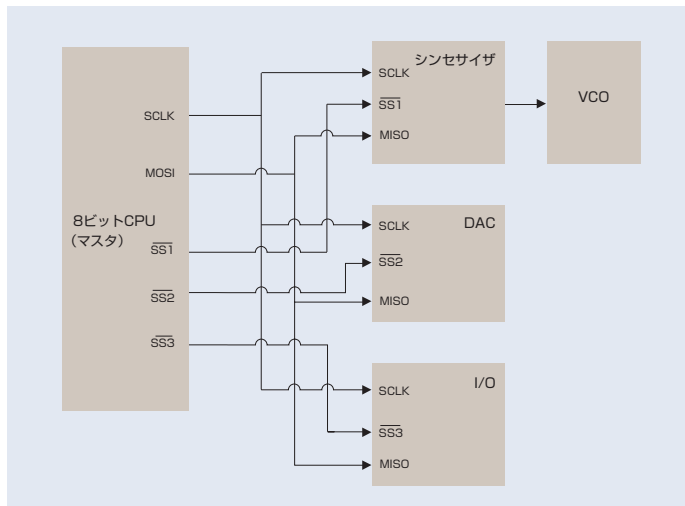


図11. SPI経由でコントロールされるシンセサイザ

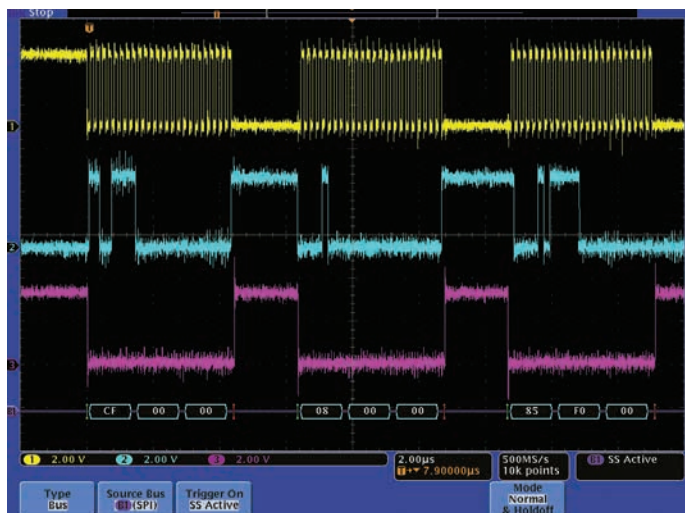


図12. SPIバスにおけるシンセサイザのコンフィグレーション・メッセージ



図13. 正しいシンセサイザのコンフィグレーション・メッセージ

メッセージの最後がすべて0になっていることから、ソフトウェアにおいて24ビット・ワードごとのビットの順序を逆にしてプログラムするという、SPIでよくあるミスであることがわかりました。プログラムを変更して再度信号をとり直すと、図13に示すように、VCOは2.5GHzで正しくロックされました。

この例では、SS Activeという簡単なトリガを使用しましたが、テクトロニクスのおシロスコープには、以下のようなSPIに関するトリガ機能があります。

- SS Active – スレーブ・デバイスでSS (Slave Select) 信号が真になるときにトリガします。
- MOSI – マスタからスレーブへの16バイトまでの任意のデータでトリガします。
- MISO – スレーブからマスタへの16バイトまでの任意のデータでトリガします。
- MOSI/MISO – マスタからスレーブへ、スレーブからマスタへの16バイトまでの任意のデータでトリガします。

ここでも、観測したいトラフィックのみにトリガすることができます。また、デコード機能により、バスに転送されるすべてのメッセージの内容が容易に理解できます。

USBの速度	ビット・レート	ビット周期
Low-Speed USB 2.0	1.5Mbps	667ns
Full-Speed USB 2.0	12Mbps	83.3ns
High-Speed USB 2.0	480Mbps	2.08ns
SuperSpeed USB 3.0	5Gbps	200ps

表2. USBの速度

USB

概要

USB (Universal Serial Bus) は今日のパーソナル・コンピュータ (PC) の主要なインターフェースとなっており、従来の外部シリアル/パラレル・バスと置き換わっています。1995年に登場して以来、USBは元々のPC用途以外にも使用されるようになり、さまざまな種類の電気デバイスで数多く使用されるようになりました。

2000年に発表されたUSB 2.0の仕様は、今日使われているほとんどのUSBデバイスに対応しています。USB 1.1仕様から進化したUSB 2.0仕様では、USB 1.1仕様でサポートされたロースピードとフルスピード・インターフェースに、新たにハイスピード・インターフェース (表2を参照) が追加されました。

USBは、システム間通信だけの拡張にとどまりません。例えば、チップ間通信ではIC_USB (Inter-Chip USB) とHSIC (High-Speed Inter-Chip) が使われます。USB 2.0仕様の補則では、IC_USB、HSIC、その他の機能強化が図られています。

USB 3.0仕様は、2008年に発表されました。USB 3.0はSuperSpeed USBとも呼ばれ、転送速度はUSB 2.0の10倍になっています。SuperSpeed USBは、USB 2.0デバイスの下位互換性をとっています。USB 3.0はUSB 2.0に加えられる仕様であり、USB 2.0と置き換わるものではありません。SuperSpeed USBのデバイスは、USB 2.0デバイスのフレームワーク・コマンドとディスクリプションを満足する必要があります。

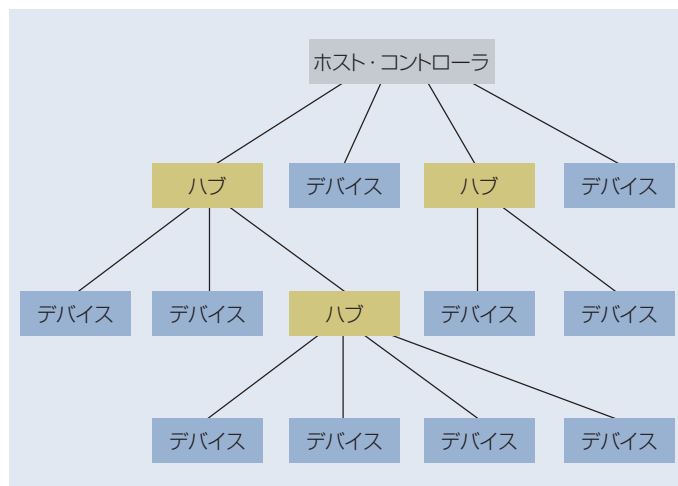


図14. USBツリー型トポロジによるバスの拡張

USB-IF (USB Implementers Forum) がUSBの規格と技術を管理、運営しています。USBの仕様は、USB-IFのウェブ・サイト (www.usb.org) で確認することができます。

動作原理

USBでは、1台のホストで1~127のデバイスを接続することができます。USBはツリー型トポロジであり、オプションのハブでさらに拡張することができます (図14を参照)。ホストのみがマスタとなり、すべてのバス・トラフィックを制御します。ホストはデバイスのすべての通信を初期化し、デバイスはホストに対して割り込むことはできません。

表2に示すように、USBには4種類の速度があります。ハイスピード・デバイスはフルスピードで開始し、その後ハイスピードに移行します。USB 2.0では、ホスト・コントローラに接続されている最も遅いデバイスによって転送速度は制限されます。

SuperSpeed USBでは2台のホスト・コントローラが使用され、1台はSuperSpeed USBデバイス用に、もう1台はUSB 2.0デバイスで使用されます。USB 2.0のシステムと同様、USB 2.0デバイスの速度は最も遅いデバイスによって制限されます。

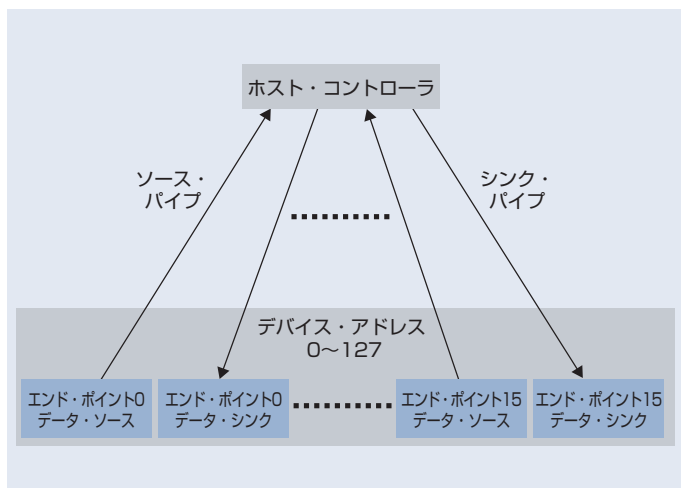


図15. USBのエンドポイントはデータを送信するとデータ・ソースになったり、データを受信するデータ・シンクになったりする

デバイス・エンドポイント

デバイス・エンドポイントは論理的な通信経路で、データを送信するデータ・ソースになったり、データを受信するデータ・シンクになったりします。各デバイスは、最大16のデータ・エンドポイントを持つことができます(図15を参照)。エンドポイント0は必須であり、ホストがデバイスと通信するために使用されます。パイプは、デバイスのエンドポイントとホストのアプリケーション・ソフトウェア間を結ぶ論理的な接続のことです。

エニユメレーション (Enumeration)

エニユメレーションは、電源投入時またはデバイスがホットプラグされる際に発生するコンフィグレーション・プロセスです。ホストがUSBバス上にデバイスを認識すると、アドレス0、エンドポイント0のSETUPトークン(パケット)でデバイスをポーリングします。次に、デバイスをユニークなアドレス1~127にアサインします。また、ホストはデバイスの速度とデータ転送タイプを識別します。エニユメレーションにおいて、デバイス・クラスは決定されます。デバイス・クラスでは、プリンタ、マス・ストレージ、ビデオ、オーディオ、ヒューマン・インタフェースなどのデバイスの機能が定義されます。

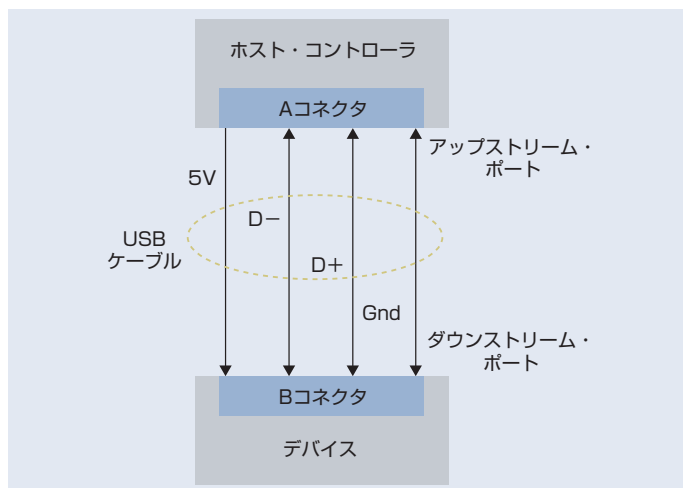


図16. USBの4線式ケーブルには、アップストリーム・ポート用のAコネクタとダウンストリーム・ポート用のBコネクタが装備されている

電氣的構成

ホストは、アップストリームでAコネクタを、デバイスはダウンストリームでBコネクタを使用します。それぞれのコネクタには、標準、ミニ、マイクロという3種類の形状があります。

USB 2.0のケーブルは、図16のように4線で構成されています。2本は、ホストからの電源(5V(赤)とグランド(黒))を供給します。コネクタは、電源とグランドのピンより前に、データ・ピンが接続するように設計されています。ホストは、パワー・マネージメントによって100mAから500mAまでの電流を供給します。例えば、デバイスへの電力はホストまたはハブでモニタされており、過電流になると電力は切り離されます。

USBの速度	ロー・ステート	ハイ・ステート
ロースピード	0.3V未満	2.8V以上
フルスピード	0.3V未満	2.8V以上
ハイスピード	0V±10%	400mV±10%

表3. 電気信号特性

PIDのタイプ	PID名	PID
トークン	OUT	0001
	IN	1001
	SOF	0101
	SETUP	1101
データ	DATA0	0011
	DATA1	1011
	DATA2	111
	MDATA	1111
ハンドシェイク	ACK	0010
	NAK	1010
	STALL	1110
	NYET	0110
スペシャル	PRE	1100
	ERR	1100
	SPLIT	1000
	PING	0100
	Reserved	0000

表4. USBパケットの種類

Data+ (D+緑) と Data- (D-白) のツイスト差動ペア線はホストによって制御され、半二重差動シグナリングにより双方向通信で使用されます。信号レベルを表3に記します。バスはDCカップリングです。

デバイスが接続されていない場合、ホストはD+とD-の両方をプルダウンします。これを、シングルエンド・ゼロ (SE0) ステートと呼びます。デバイスがUSBバスに接続されると転送速度によりD+またはD-がプルアップされ、信号レベルを変えることによりデバイスの転送速度を表します。

K	J	K	J	K	J	K	K
---	---	---	---	---	---	---	---

図17. ロースピードとハイスピードのSYNCフィールド

Jアイドル・ステートが続いているとき、D-スピード・デバイスはD-をHIにプルアップし、D-が正の電圧となります。フルスピード・デバイスの場合はD+をHIにプルアップし、D+が正の電圧となります。KステートではJステートの正負が逆転します。

データ伝送では、ビット・スタッフィングを用いたNRZI (Non Return Zero Inverted) 符号方式により、トランジション数を少なくしています。LSB (Least Significant Bit) が最初に送られ、MSB (Most Significant Bit) が最後に送られます。

パケット

パケットは、USB通信の基本単位です。パケットは、同期フィールド (SYNC) で始まり、パケット識別子 (PID) が続きます。PIDに続くフィールドは、パケットの種類 (フレーム開始パケット、トークン・パケット、データ・パケット等) によって異なり、ハンドシェイク・パケットの場合はフィールドがありません。エンドオブパケット (EOP) でパケットを終了します。

Jのアイドル・ステートに続いて、ロースピードとフルスピードではパケットは8ビットのSYNCフィールドが始まります。SYNCは3つのKJペアであり、その後2つのKが続きます (図17を参照)。ハイスピードのSYNCフィールドでは15のKJペアの後に2つのKが続きます。ハブでは簡素化され、5つのKJペアと2つのKで済ませることができます。

PIDは2番目のパケット・バイトであり、4ビットのPIDと、エラー・チェックのための反転した4ビットPIDからなります。最初のPIDの4ビットが後の反転した4ビットと一致しない場合、PIDの符号化エラーとなります。LSBが最初に、MSBが最後にバス上に送られます。

PIDの4ビット・パターンにより、表4に示すように17種類のパケットを区分します。PREとERRのパケットは、同じPIDコードが与えられます。パケットは、トークン、データ、ハンドシェイク、スペシャルの4つのグループに分けられます。

EOPは3ビットからなり、2ビットのSE0と1ビットのJステートで構成されています。

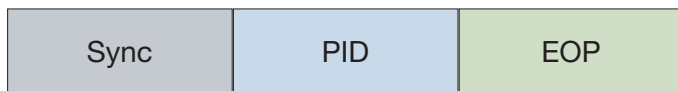


図18. ハンドシェイク・パケットの構成

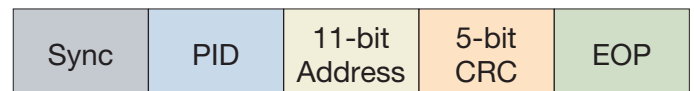


図19. OUT、IN、SETUPトークン・パケットの構成



図20. DATA0またはDATA1によるデータ・パケット

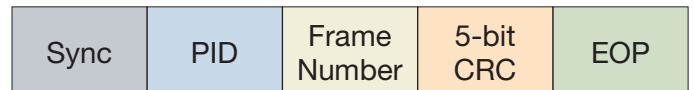


図21. SOF (Start Of Frame) パケット

ハンドシェイク・パケット

データ・パケットが正常に受信されたときに送られるACKや、データ・パケットを受信できないときに送られるNAKなどのハンドシェイク・パケットは、図18に示すようにシンク・バイト、PIDバイト、EOPから構成されます。

トークン・パケット

常にホストに送られるトークン・パケットは、SYNC、PID、および11ビットのアドレス、5ビットのCRC (Cyclic Redundancy Check) からなる2バイトとEOPから構成されます (図19を参照)。

OUT、IN、SETUPトークンの11ビット・アドレスは、7ビットのデバイス・アドレスと4ビットのエンドポイント識別子に分かれています。

アドレス0は特殊な用途であり、ホストがデバイスにアドレスを指定するエニュメレーション・プロセスの始めではデバイスのアドレスは割り当てられていません。エニュメレーション・プロセスの後半において、ホストはデバイスにゼロでないアドレスを割り当てます。

すべてのデバイスはエンドポイント0を持っています。エンドポイント0はデバイスの制御とステータスで使用されます。他のエンドポイントはデータを転送するデータ・ソース、データを受信するデータ・シンクで使用されます。

ホストは送信の場合、デバイスに対してまずOUTトークンを送り、データ・パケットを送ります。受信の場合、ホストがデバイスにINトークンを送り、それに反応したデバイスからのデータ・パケットまたはNAKなどのハンドシェイク・パケットを受け取ります。

データ・パケット

データ・パケットは、図20に示すようにPIDバイト、データ・バイト、16ビットのCRCとEOPで構成されます。

表4に示したように、DATA0とDATA1というパケットがあり、それぞれ1と0にトグルできる1ビットのシーケンス番号を持っています。シーケンス番号は停止や再送信の自動要求に使用されます。パケットDATA0とパケットDATA1は、転送が正常に行われると交互に切り替わります。転送エラーが発生すると、データ・パケットは同じシーケンス番号で再送されます。

正常なデータ転送においては、ホストがデバイスにDATA0パケットを送り、デバイスがハンドシェイクACKパケットを返し、続いてホストがDATA1パケットを送ることになります。

ホストがデバイスからのハンドシェイクACKパケットを受信しないか、またはNAKパケットを受信した場合、ホストはDATA0パケットを再送します。デバイスがACKパケットを送り、ホストの使ったシーケンス番号と同じシーケンス番号でデバイスがデータ・パケットを受信した場合、デバイスはデータ・パケットを認識し、古いデータは無視します。

SOF (Start Of Frame)

図21に示すように、SOFは非同期データと同期をとるために使用されます。11ビットのフレーム番号は、SOFパケットが送信されるたびに1ずつ増えます。

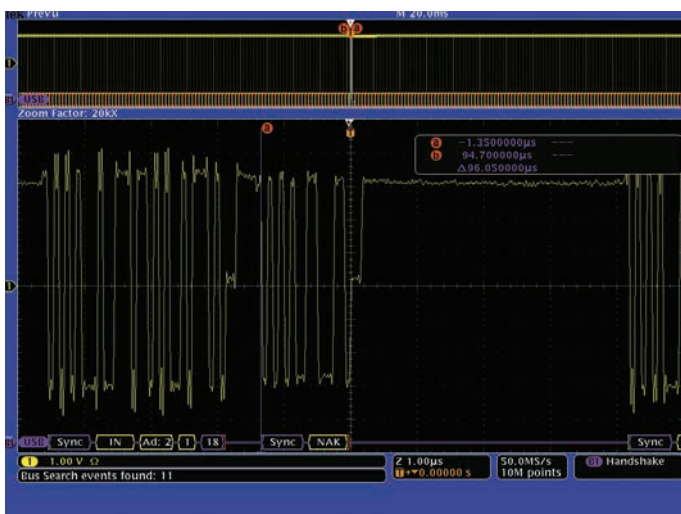


図22. 最初のINトークン・リクエストに対してカーソルA以降でホストへのNAKを返信したフルスピード・メモリ・デバイス

USB2.0 デバッグのポイント

テクトロニクスのおシロスコープには、USBのシリアル・トリガ／解析機能を持ったものがあります（付録Aを参照）。ロースピードおよびフルスピードUSBのトリガ、デコード、サーチが実行できます。ハイスピードUSBでは、1GHz以上の帯域を持ったおシロスコープが必要です。

例えば、フルスピードのUSBメモリ・デバイスのデータ・レイテンシ性能のチェックでは、USBメモリ・デバイスからデータを出力するようにコンピュータがINトークン・リクエストを発した後、メモリ・デバイスがNAKで応答するかどうかを観測します。

コンピュータとUSBメモリ・デバイス間を接続するUSB延長ケーブルには、TDP1000型差動プローブでプロービングします。ケーブルにプローブを接続する前に、TDP1000型プローブのメニュー・ボタンを押してプローブの4.25VレンジのAutoZeroを実行します。

USBバスを設定するには、バス・メニューからUSBを選択します。次に、設定ボタンの左から右へと速度、ソース・チャンネル、プローブの種類、スレッシュールドなどのバスのパラメータを設定します。この例では、フルスピードのスレッシュールドのプリセット値である1.4V、-1.4Vが使われています。

まず、SETUPトークンでトリガをかけ、エニュメレーション・プロセスをチェックします。エニュメレーション後にSOFパケットでトリガをかけてSOF（パケット）を確認し、Jアイドル・ステートが正であるかチェック、またはSOFのSYNCフィールドのビット幅を測定することで転送速度を判定します。

次に、NAKトークンでトリガするように、またシングル・アキュイジション・モードで取込むようにおシロスコープを設定します。次に、メモリ・デバイスからデータを出力するようにコンピュータからリクエストします。メモリ・デバイスのデータ転送準備ができていない場合はNAKが生じませんので、おシロスコープはトリガしません。しかし、USBメモリ・デバイスのデータ転送準備ができていない場合、コンピュータ・ホストのINトークンに対してUSBメモリ・デバイスはNAKで応答しますので、NAKでおシロスコープのトリガがかかります。NAKでトリガした結果を図22に示します。



図23. MSO5204型Opt. SR-USBで取込み、デコード表示したハイスピードUSB 2.0の波形

オシロスコープのトリガ設定は、Wave Inspectorのサーチ条件にコピーすることができます。Wave Inspectorは、すべての取込みからNAKをサーチします。この例では、Wave Inspectorによって11個のNAKが検出されました。最初のNAKがトリガ・ポジションであり、残りの10個のNAKがトリガの後に位置します。すべてのNAKは、コンピュータが再送したINトークンに応答したものです。Wave Inspectorの左右の矢印ボタンを押すことで、マークされたそれぞれのNAKに簡単に移動することができます。

USBに対して以下のようなトリガの種類があります。

- 同期
- リセット
- サスペンド
- レジウム
- エンドオブパケット
- トークン(アドレス)パケット
 - SETUP、IN、OUT、SOF、任意
- データ・パケット
 - 任意のデータ値、DATA0、DATA1またはDATA2またはMDATA (ハイスピードのみ)
 - データのマッチングは最大16バイト分のパターン
- ハンドシェイク・パケット
 - 任意のハンドシェイク値、またはACK、NAK、STALL、またはNYET
- 特殊パケット
 - 予備、PRE (フルスピードのみ)、またはERR、SPLIT、またはPING
- Error typesは、PID Check Bits、Token CRC5、Data CRC16とBit stuffing (ロースピードとフルスピードのみ)

Wave Inspectorは、トリガでを使用した同じ条件でサーチすることができます。

当社のオシロスコープを使用することで、USB 2.0信号、プロトコル、データを簡単に取込み、他のアナログ信号、デジタル信号との相関をとることができます。

Type	Preamble	Start-of-frame Delimiter	Destination Address	Source Address	Length/Type	Data + Pad	Frame check sequence
Bytes	7	1	6	6	2	46-1500	4

図24. IEEE 802.3規格のEthernetフレーム・フォーマット

Ethernet

概要

Ethernetは、1970年代初頭にXerox PARCによって開発された、ローカル・エリア・ネットワーク (LAN) 用のフレームベース・コンピュータ・ネットワーク技術ファミリです。最初の規格ドラフトは、1980年にIEEE (Institute of Electrical and Electronics Engineers) によって発行されました。IEEE 802.3 CSMA/CDは1982年に、国際規格であるISO/IEEE 802.3規格は1984年に承認されました。

動作原理

Ethernetで代表的な規格は10BASE-Tと100BASE-TXであり、ほとんどのパーソナル・コンピュータで使用されています。データ・レートはMbpsで表示されます。BASEとは、信号がベースバンド信号であり、RF信号変調されていないことを意味しています。Tとは、ネットワーク・ノード間で使用されるLANケーブルがツイスト (Twisted) ペア・ケーブルであることを意味しています。

10BASE-Tと100BASE-TXはハードウェアの実装コストが安価であることもあり、数多くの組込みシステム設計にも採用されるようになりました。

Ethernetはピアツーピアのフレームベース通信であり、直接のポイントトゥポイント通信が可能になります。物理レイヤでは、10BASE-T、100BASE-TX信号はアドレス、コントロール、データ、クロック情報を伝送します。データは、フレームと呼ばれるデータ・バイトのシーケンスとして伝送されます。Ethernetフレームは、ハイレベルのプロトコル・フレームを伝送することもできます。例えば、Ethernetフレームには、TCP (Transmission Control Protocol) を含むIP (Internet Protocol) が含まれることがあります。このように信号が複雑であるため、10BASE-T、100BASE-TX波形の解析において特定のイベントを分離するのが難しくなっています。

Ethernetのデータ・フレーム・フォーマットは、IEEE 802.3規格で定義されており、図24に示すように7つのフィールドで構成されています。

プリアンプルは7バイト長であり、同期のための1、0の繰り返しパターンになっています。

SFD (Start Frame Delimiter) は1バイト長であり、1、0の繰り返しパターンですが、最後は1が2個になります。

宛先アドレスと送信元メディア・アクセス・コントロール (MAC) アドレスは6バイト長であり、MSB (最上位ビット)、LSB (最下位ビット) の順番で伝送されます。各Ethernetノードには独自のMACアドレスが割り当てられており、各データ・フレームの宛先と送信元を指定します。これにより、ほとんどのリンク・レイヤ (OSI Layer 2) ネットワークの基礎となっており、上位のレイヤ・プロトコルについで複雑で、機能的なネットワークになります。

長さ/タイプのフィールドは2バイトの値です。長さ/タイプの10進の値が1500以下の場合、データ・フィールドにおけるデータ・バイトの数を表します。1536 (0x0600) より大きい場合、このフィールドはEthernetが指定されます。Ethernetのペイロードには、EtherTypeで指定した規格化されたプロトコルによりデータが書き込まれます。(例: EtherTypeはIPv4では0x0800に設定されます。)

データ・フレームは、46~1500バイトです。データが46バイト未満の場合は、データ・フィールドは46バイト長になります。

FCS (Frame Check Sequence) は32ビットのCRC (Cyclic Redundancy Check) であり、宛先アドレス、送信元アドレス、長さ/タイプ、データ・フィールドのエラーをチェックします。

各フレームを送った後、トランスミッタは次のフレームの前に最低でも12バイトのアイドル・キャラクタを送るか、あるいは伝送イネーブル信号をデアサートすることで同じ時間だけアイドルである必要があります。

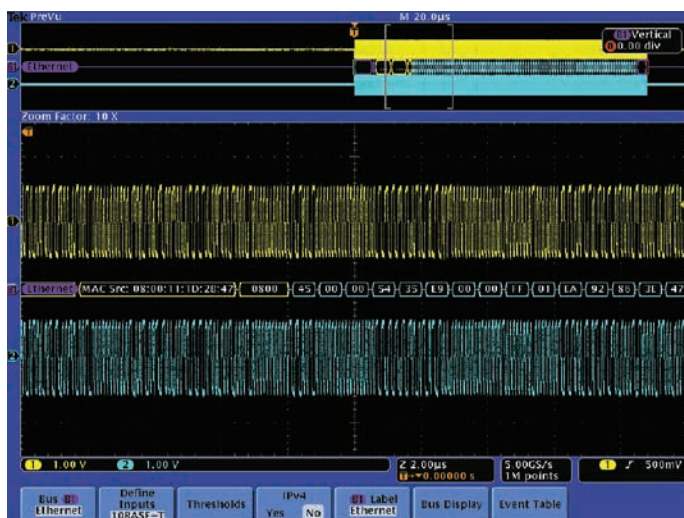


図25. 10BASE-Tのデコード表示

Ethernet デバッグのポイント

Ethernetは、組込み設計で広く使用されるようになりました。Ethernetのトラフィックを物理レイヤ、プロトコル・レイヤの両方で解析することで、組込み設計の他のサブシステムの動作を詳細に理解することができます。しかし、Ethernet信号にはアドレス、コントロール、データ、クロック情報が含まれているため、特定のイベントを分離するのが難しくなっています。MSO/DPO4000BシリーズのDPO4ENET型Ethernetシリアル・トリガ／解析アプリケーション・モジュールには自動トリガ、デコード、サーチ機能があり、オシロスコープと組み合わせることで10BASE-T、100BASE-TXをベースとしたシステム・デバッグのための強力なツールになります。

SFD (Start Frame Delimiter)、MACアドレス、MAC長さ／タイプ、MACクライアント・データ、Qタグ・コントロール情報、IPヘッダ、TCPヘッダ、TCP/IPv4クライアント・データ、EOP (End of Packet)、アイドル (100BASE-TXのみ)、FCS (CRC) エラーなどのEthernetパケット内容にトリガすることができます。

デコード表示には10BASE-T、100BASE-TX信号が表示され、パケットの始めと終わり、プリアンプル、SFD、MACアドレス、データ、FCS、エラーなどのサブパケット・コンポーネントが簡単に識別表示されます。バスの各パケットはデコードされ、値は16進、バイナリ、またはASCIIでバス波形とともに表示されます。

デコードされたパケット・データがバス波形上で見えるだけでなく、取込んだすべてのパケットを、ロジック・アナライザのようにリスト形式で見ることができます。パケットにはタイムスタンプが付き、各コンポーネント (時間、宛先アドレス、送信元アドレス、長さ、データ、FCS/CRC、エラー) の欄が連続的にリスト表示されます。

シリアル・トリガは特定のイベントを検出するのに非常に便利な機能ですが、取込んだ後、そのイベントの周辺のデータまでは解析できません。DPO4ENET型アプリケーション・モジュールはWave Inspectorにより、シリアル・パケットの内容など、ユーザが定義した条件で取込んだデータを自動的に検索することができます。検出されたイベントには検索マークが付きます。各ブックマークへの移動は、オシロスコープ前面パネルの(←) ボタン、(→) ボタンを押すだけです。

RS-232

概要

RS-232は、近距離の2つのデバイス間において広く使われているシリアル通信規格です。PCのシリアル・ポートが良く知られていますが、組込みシステムのデバッグ・ポートまたは2つのデバイスのリンクとしても使用されています。

RS-232-C規格は1969年に発表されました。2回改定されていますが、小さな変更です。関連した規格にRS-422やRS-485があります。同様の規格ですが、差動シグナリングにより長距離での通信を可能にしています。

動作原理

DTE (Data Terminal Equipment) と DCE (Data circuit-Terminating Equipment) と呼ばれる2種類のデバイスで構成されます。アプリケーションによってはDTEデバイスがDCEデバイスをコントロールすることがありますが、どちらも同格でDTEとDCE間の区別が任意の場合もあります。

RS-232規格には数多くの信号が規定されていますが、その多くはあまり使われていません。最も重要な信号がTx (Transmitted Data) と Rx (Received Data) です。TxはDTEからDCEへとデータを転送します。DTEデバイスのTxラインは、DCEデバイスのRxラインになります。同様に、RxはDCEからDTEにデータを転送します。

RS-232規格では使用するコネクタを規定していませんが、一般的に25ピンおよび9ピンのコネクタが使用されています。10ピン、8ピン、6ピンのコネクタもあります。同じ基板上で、標準のコネクタを使用せずに2つのRS-232デバイスを接続することもできます。

信号	略記	ピン
キャリア検出	DCD	1
受信データ	Rx	2
送信データ	Tx	3
データ端末レディ	DTR	4
コモン・グラウンド	G	5
データ・セット・レディ	DSR	6
送信リクエスト	RTS	7
送信可	CTS	8
リング・インジケータ	RI	9

表5. 一般的なRS-232コネクタのピン配置

2つのRS-232デバイスを接続する場合、一般にヌル・モデムが必要になります。ヌル・デバイスは、Tx、Rxラインなど、いくつかのラインをクロスにします。これにより、それぞれのデバイスはTxラインでデータを送り、Rxラインでデータを受け取ります。

表5は、RS-232で一般的に使用されている9ピン・コネクタにおけるピン配置を示します。ヌル・モデムで信号を伝送する場合、信号がクロスになるため注意が必要です。特に注意が必要なのは、TxとRxがクロスすることです。

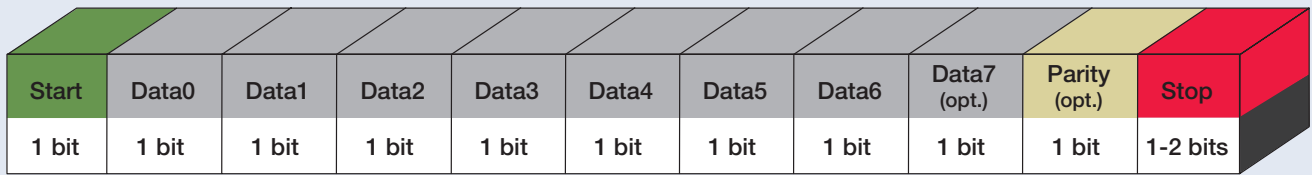


図26. RS-232のバイト構成

RS-232信号をプロービングする場合、ブレイクアウト・ボックスを使うと便利ことがあります。ブレイクアウト・ボックスを使うことで、RS-232ケーブル内の信号が簡単にプロービングできます。高価なものでないため、電気量販店などで購入することができます。

RS-232では、バスを伝送する信号の内容は規定していません。ASCIIテキストが一般的ですが、バイナリ・データも使用できます。データはパケットに分割されることがあります。ASCIIテキストの場合、パケットは新しいラインまたは改行キャラクタで終了します。バイナリ・データでは、一般に00またはFF (Hex) が使用されます。

デバイスでは、UART (Universal Asynchronous Receiver/Transmitter) を使用してRS-232を実装します。UARTは広く普及しているパーツで、シフト・レジスタによりバイト・データをシリアル・ストリームに、またその逆に変換します。組込み設計では、UARTはRS-232のトランシーバを使用せずに直接通信することもできます。

図26は、RS-232データの1バイトを示しています。バイトを構成する各ビットを以下に説明します。

- Start – バイトは、Startビットから始めます。
- データ – 次にいくつかのビットが続きます。8ビット・データが一般的ですが、アプリケーションによっては7ビット・データを使用することもあります。7ビットしか伝送しない場合でも、非公式ですが「バイト」と呼びます。UARTとUARTの通信では、9ビット・ワードが使用されることもあります。
- Parity – オプションのパリティ・ビットです。
- Stop – 1、1.5または2ビットがStopビットとして使用されます。

RS-232バスには、クロック・ラインがありません。各デバイスは、独自のクロックによりデータ・ラインにサンプルするタイミングを決めます。多くの設計では、UARTはTx、Rx信号の立上りエッジによりクロックと他のデバイスのクロックを同期させます。

RS-232 デバッグのポイント

テクトロニクスのおよそすべてのオシロスコープは、RS-232バスのトリガ／解析が行えます（付録Aを参照）。オシロスコープにPCや特殊なデコーダを接続することなく、RS-232、RS-422、RS-485またはUARTのデータを簡単に観測することができます。

RS-232バスのチャンネル、ビット・レート、パリティなど基本的なパラメータは、前面パネルのバス・ボタンで設定します（図27を参照）。

この例では、ASCIIのデコードを選択しているため、RS-232のデータをバイナリまたは16進で表示します。

RS-232バスでセンサのデータを収集するデバイスの例を考えてみます。センサのデータをリクエストしていますが、応答がありません。センサがリクエストを受け取っていないか、受け取っているがリクエストを無視しているかを調べる必要があります。

まずTxとRxにプロービングし、オシロスコープでバスを設定します。次に、データ・リクエストがTxラインに送られたときにトリガするようにオシロスコープを設定します。取込まれた信号を図28に示します。

デジタル・チャンネル1にはTxラインが、デジタル・チャンネル0にはRxラインが表示されます。ここでは、Rx、Txの波形の上に表示される、デコードされたデータに注目します。センサからの応答をズーム表示して観測します。Txラインにはリクエストが、Rxラインには応答が観測されています。カーソルで測定したところ、リクエスト後約37msに応答がきていることがわかります。コントローラのタイムアウトを増やしてセンサからの応答時間に余裕を持たせたところ、問題は解決できました。



図27. RS-232バス・サポートの設定メニュー

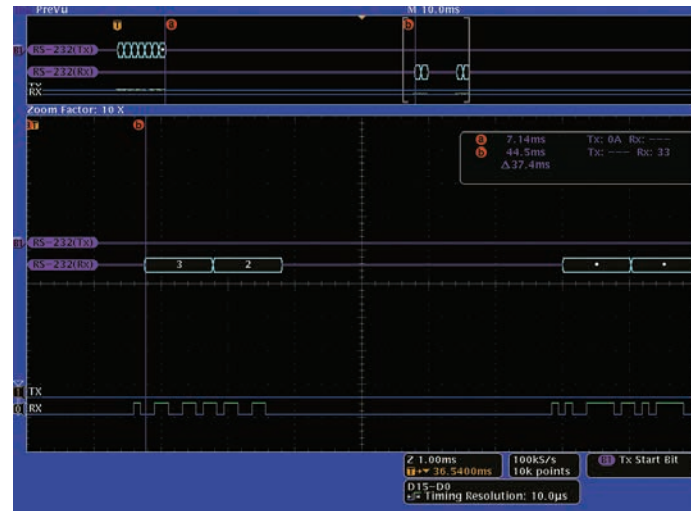


図28. 2つのRS-232バスのメッセージ間の時間差測定例

RS-232トリガ機能を以下に説明します。

- Tx Start Bit — バイトの開始点のビットにトリガします。
- Tx End of Packet — パケットの最後のバイトにトリガします。パケットは、次のバイトで終了します：ヌル (00 hex)、ラインフィード (0A hex)、改行 (0D hex)、スペース (20 hex)、または FF hex。
- Tx Data — 10バイトまでの特定の値を設定してトリガします。
- Rx Start Bit、Rx End of Packet、Rx Data — Txトリガに似ていますが、Rxラインのものです。

当社のオシロスコープは、RS-232信号が簡単に観測し、解析し、デバイスのその他の動きと相関をとって表示することができます。

CAN/CAN FD

概要

CAN (Controller Area Network) は、1980年代に Robert Bosch GmbH 社により、電氣的ノイズの多い環境におけるデバイス間の低コストの通信バスとして開発されました。1992年、メルセデス・ベンツ社は、自動車システムで初めてCANを採用しました。今日では、ほとんどすべての自動車メーカーがCANコントローラ、ネットワークを採用しており、さまざまなECUを制御しています。CANは、エンジンのタイミング・コントロール、アンチロック・ブレーキ・システム、パワー・トレイン・コントロールなどの基本的なバスとして使用されています。また、電気ノイズ耐性、少ない配線、優れたエラー検出機能、高速データ転送などにより、CANは産業用制御システム、海洋産業、航空宇宙産業などのアプリケーションでも急速に拡がりつつあります。

自動車のネットワークがさらに数多くの機能をサポートするように進化してきたため、ノード間における高速なデータ通信のニーズが高まってきました。これを受け、最高データ・レート1Mbps、ペイロード・データ8バイトのCANに比べて最高データ・レート8Mbps、ペイロード64バイトを実現したCANのハイスピード版であるCAN FDが登場しました。CAN FDの最初のバージョンは2012年に発表されましたが、2015年にはISO CAN FDというISO規格にアップデートされました。ISOバージョンでは、セーフガードが追加されたことで通信の信頼性を改善しています。元々のバージョンは現在では非ISO CAN FDとして知られており、ISO CAN FDとの互換性はありません。

動作原理

CAN/CAN FDバスは、シールド・ツイスト・ペア (STP)、非シールド・ツイスト・ペア (UTP)、リボン・ケーブルなどを使用した、平衡 (差動) 2線式インタフェースです。各ノードは、オスの9ピンDコネクタを使用します。NRZ (None Return Zero) ビット・エンコードとビット・スタッフィングにより、数少ない伝送と優れた雑音耐性を実現しています。CANバス・インタフェースでは非同期伝送を使用し、バスが空いている場合は、どのノードでも伝送を開始できます。メッセージは、ネットワーク上のすべてのノードに配信されます。複数のノードが同時にメッセージを伝送すると、IDのビット単位のアービトラージにより優先度の高いメッセージを決定します。メッセージ・タイプには、データ・フレーム、リモート・トランスミッション・リクエスト (RTR) フレーム、エラー・フレーム、またはオーバーロード・フレームの4種類があります。エラーを検出したノードはエラー・フレームを送信し、バス上のすべてのノードは現在のメッセージが不完全であるとみなし、送信したメッセージは再送信されます。

CAN 2.0	SOF 1ビット	アービトレーション・ フィールド 11ビット (標準ID) 29ビット (拡張ID)	コントロール・ フィールド 6ビット	データ・ フィールド 0~8バイト	CRCフィールド 16ビット	ACK 2ビット	EOF 7ビット	INT 3ビット
CAN FD	SOF 1ビット	アービトレーション・ フィールド 12ビット (標準ID) 32ビット (拡張ID)	コントロール・ フィールド 8または9ビット	データ・ フィールド 0~64バイト	CRCフィールド 18または 22ビット	ACK 2ビット	EOF 7ビット	INT 3ビット

図29.CAN、CAN FD

オーバーロード・フレームは受信デバイスにより送信され、データを受信する準備ができていないことを示します。データ・フレームは、リモート・フレームがデータをリクエストするのに対してデータ送信で使用されます。データ・フレーム、リモート・フレームは、図29に示すように、各フレームの最初と最後にあるスタート・ビットとストップ・ビットで制御され、アービトレーション・フィールド、コントロール・フィールド、データ・フィールド、CRCフィールド、ACKフィールドを含んでいます。

- SOF – CAN、CAN FD共通であり、SOF (Start Of Frame) ビットでフレームは開始します。
- アービトレーション – CANのアービトレーション・フィールドは識別子 (アドレス) とRTR (Remote Transmission Request) ビットを含み、データ・フレームとリクエスト・フレーム (リモート・フレームとも呼ばれる) を区別します。識別子には、標準フォーマット (11ビット、バージョン2.0A) と拡張フォーマット (29ビット、バージョン2.0B) があります。CAN FDでは、標準/拡張フォーマットでCANと同じアドレス (11ビットIDまたは29ビットID) を共有しますが、RTRビットを外してドミナントr1ビットを維持します。
- コントロール – CANのコントロール・フィールドは、CAN 2.0A (11ビット識別子) 標準フレームとCAN 2.0B (29ビット識別子) 拡張フレームを区別するための6ビットの識別拡張子 (IDE) ビットからなります。コントロール・フィールドに含まれるDLCの4ビットは、データ長コード (DLC) も含んでいます。DLCの4ビットは、データ・フレームのデータ・フィールドに含まれるバイト数、またはリモート・フレームによってリクエストされるバイト数を示します。CAN FDではコントロール・フィールドで8または9ビットを使用し、CANと同様にIDE、r0、DLCビットも使用します。CAN FDではEDL、BRSとBRSの3つのビットも追加され、EDL (Extended Data Length) はパケットがCANかCAN FDかを識別し、BRS (Bit Rate Switch) はデータ・フェーズでビット・レートをスイッチするかどうかを示し、アービトレーション・フェーズとデータ・フェーズを分離します。CAN FDでは、同じ4ビットのDLCが8バイト以上のデータ長の指定でも使用されます。

- データ – CANのデータ・フィールドは、0~8バイトのデータからなります。CAN FDは0~8バイトをサポートしますが、12、16、20、32、48または64バイトもサポートされます。
- CRC – CANで使用される15ビットのCRCコードとリセッブ・デリミタ・ビットです。CAN FDでは16バイト以下のペイロードで17ビット (+CRCデリミタ・ビット) を、16バイト以上では21ビット (+CRCデリミタ・ビット) を使用します。CAN FDでは、4つのスタッフ・ビットが追加されます。
- ACK – アクノリッジ・フィールドは2ビット長。最初はスロット・ビットで、リセッブとして送信され、他のノードから正常にメッセージを受信した場合は送信されたドミナント・ビットにして上書きされます。二番目のビットはリセッブ・デリミタ・ビットです。CAN FDでは少し異なり、レシーバは有効ACKとして2ビットの時間を識別します。
- EOF – 7ビットのリセッブ・ビットで、フレームの終了 (End Of Frame) を示します。

3つのリセッブ・ビットのINT (Intermission) フィールドは、バスがフリーであることを示します。バス・アイドル時間は、0を含む任意の長さです。さまざまなデータ・レートが規定されており、CANでは最速1Mbps、CAN FDでは8Mbpsです。すべてのモジュールは最低20kbpsをサポートする必要があります。ケーブル長は、使用されるデータ・レートによって異なります。通常、システム内のすべてのデバイスは均一で固定のビット・レートで情報を伝送します。最大ライン長は、ロースピードでは数千m、1Mbpsでは40mが代表値です。ケーブル端では、終端抵抗を使用します。



図30. CANバス・サポートの設定メニュー



図31. CANバスの識別子とデータでトリガし、取込んだデータのすべてのメッセージをデコードした例

CAN、CAN FD デバッグのポイント

テクトロニクスのオシロスコープには、CAN、CAN FDのシリアル・トリガ/解析オプションを装備できるものがあります(付録Aを参照)。前面パネルのBusボタンを使用することでCANまたはCAN FDバスを設定でき、プロービングするチャンネル、ビット・レート、スレッシュホールド、サンプル・ポイント(ビット時間のパーセントとして)などの基本パラメータを設定します(図30を参照)。例えば、運転席のドア・スイッチで、助手席の窓を降ろすスイッチを押してから、実際に窓が動き出すまでの待ち時間を測定するケースを考えてみます。運転席ドアのCANモジュールのID、「窓を降ろす」コマンドに関連するデータを設定することで、探しているデータ・フレームに簡単にトリガすることができます。運転席ドアにある、「窓を降ろす」スイッチと、助手席ドアにあるモータ駆動回路に同時にプロービングすることで、このタイミング測定は非常に簡単に実行できます(図31参照)。

図31において、白い三角矢印のマーカは、基準ポイントとして設定しました。このマーカは、オシロスコープの前面パネルにあるSet/Clearボタンを押すことで、追加、削除できます。前面パネルにある左右の矢印ボタンを押すことで、ズーム・ウィンドウを前後のマーカまでジャンプでき、興味のあるイベントまで簡単に移動できます。

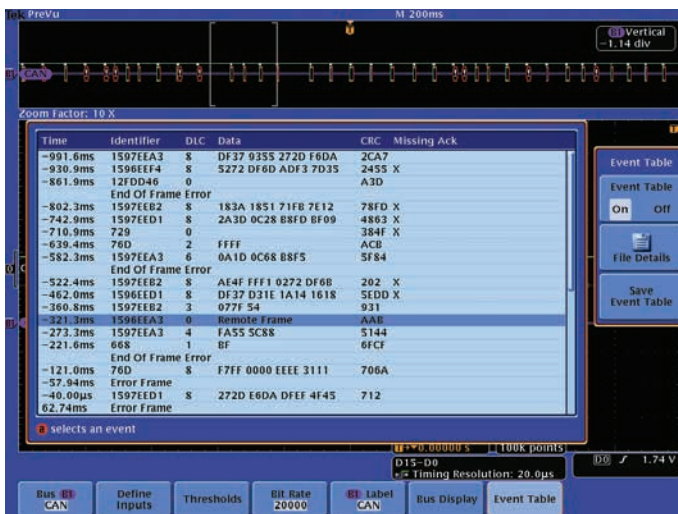


図32. CAN イベント・テーブル

この作業を、この機能を使用せずにを行うケースを考えます。CAN/CAN FD トリガがない場合は、スイッチそのものでトリガし、十分に長いアクティビティ時間ウィンドウを取込み、正しいイベントが見つかるまで、CANバスを徐々に手作業でデコードします。イベントが見つかるまで何十分、何時間もかかることがあります。

オシロスコープによる強力なCAN/CAN FD トリガ機能には、次のようなものがあります。

- Start of Frame – SOF フィールドにトリガします。
- Frame Type – Data Frame、Remote Frame、Error Frame、Overload Frame から選択します。
- Identifier – 特定の11または29ビットの識別子の値とリード/ライトのクオリフィケーションにトリガします。
- Data – ユーザ指定の1~8バイトのCANまたはCAN FD データでトリガします。大きなCAN FD ペイロードでは、トリガ・バイト・オフセットを使用してパケット内の正確な位置を指定し、1~8バイトの特定のブロックと比較します。
- Ack – 受信したデバイスがアクノリッジを戻さない場合、いつでもトリガします。
- Bit Stuffing Error – ビット・スタッフィング・エラーが検出されるとトリガします。
- FD Form Error – FD フレーム・フォーマットにエラーがあるビットでトリガします。
- Any Error – Missing Ack、Bit Stuffing、FD Form エラーにトリガします。
- End of Frame – EOF フィールドにトリガします。



図33. CAN 2.0またはCAN FD規格、ISOと非ISOバージョン、FDビット・レートなどの設定メニュー

このトリガ・タイプを使用することで、ほとんどすべてのCAN/CAN FDバスを効率的に特定することができます。しかし、トリガは単に始まりに過ぎません。トラブルシュートでは、トリガ・イベント前後のメッセージ内容を調べる必要があります。図32に示すように、イベント・テーブルでは取込んだデータから、複数のメッセージの内容を簡単に見ることができます。

イベント・テーブルには、取込んだデータから、すべてのデコードされたメッセージ内容が、タイムスタンプとともに表形式で表示されます。単にバス上のすべてのトラフィックが観測できるだけでなく、メッセージ間のタイミング測定も簡単に実行できます。イベント・テーブルは、当社オシロスコープが対応するすべてのバスで表示できます。

CAN FDは多くの設定パラメータをCANと共有しているため、その設定、動作はCANとよく似ています。図33は、CAN 2.0またはCAN FD規格、ISOと非ISOバージョン、FDビット・レートなどの設定メニューを示しています。アービトレーション・フェーズ・ビットはCANとCAN FDで同じですが (Bit Rateのラベル)、FD Bit RateはCAN FDの高いビット・レートに対応します。

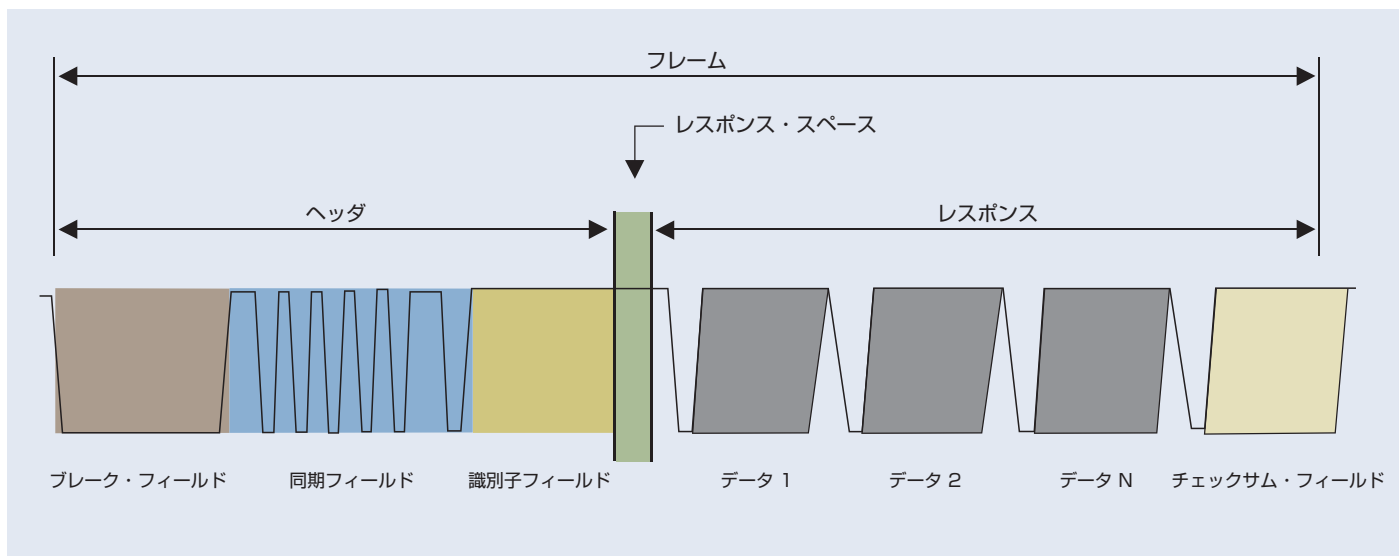


図34. LINフレームの構造

LIN

概要

LIN (Local Interconnect Network) バスは、CANではコスト、汎用性、スピードのスペックが十分すぎる場合のCANの代替として1999年にLINコンソーシアムによって開発された規格です。アプリケーションとしては、ウィンドウ・コントロール、ドア・ロック、雨滴センサ、ワイパ・コントロール、空調コントロールなどにおける、センサとアクチュエータ間の通信があります。

しかし、優れた電気ノイズの耐性、エラー検出機能、高速データ転送などのため、エンジンのタイミング・コントロール、アンチロック・ブレーキ・システム、パワー・とレイン・コントロールなどではCANが使用されています。

動作原理

LINバスは、ISO9141規格を拡張した、ローコスト、1線式の実装です。LINのネットワークは、1つのマスタと1つまたはそれ以上のスレーブからなります。すべてのメッセージはマスタから発行され、各メッセージにはただ1つのスレーブが応答します。このため、CANのような衝突検出やアービトレーション機能は必要ありません。UART/SCIをベースに、8ビット・バイト、スタート・ビット、ストップ・ビット、パリティなしでデータを送ります。データ・レートは1kbps～20kbpsです。遅いようですが、特定のアプリケーションには適しており、EMIも抑えられます。LINバスは、常にアクティブかスリープのどちらかの状態をとります。アクティブの状態では、バス上のすべてのノードはウェイクアップの状態であり、バスのコマンドを待ち受けます。バス上のノードは、マスタがスリープ・フレームを発行するか、あらかじめ設定された時間、バスに情報が来ない場合にスリープに入ります。ノードがウェイクアップをリクエストするか、マスタ・ノードがブレーク・フィールドを発行するとウェイクアップします。

LINのフレームは、ヘッダとレスポンスで構成されます。ヘッダはマスタによって出力され、レスポンスはスレーブによって出力されます。図34に示すように、それぞれがサブコンポーネントを持っています。

ヘッダ・コンポーネント：

- Break Field — ブレーク・フィールドは、新しいフレームの開始信号として使用されます。すべてのスレーブ・デバイスをアクティブにし、ヘッダの残りの部分を受信するように指示します。
- Sync Field — 同期フィールドはスレーブ・デバイスによって使用され、マスタ・ノードで使用されるボー・レートを決定し、順次同期します。
- Identifier Field — 識別子はどのスレーブ・デバイスがアクションを起こすかを指定します。

レスポンス・コンポーネント：

- Data — 指定されたスレーブ・デバイスが1～8バイト・データで応答します。
- Checksum — データ伝送におけるエラー検出のために使用される演算フィールドです。LIN規格は何種類かのバージョンで進化しており、2種類の異なるチェックサムを使用しています。標準チェックサム (Classic checksum) はデータ・バイトのみで計算され、バージョン1.xのLINシステムで使用されます。拡張チェックサム (Enhanced checksum) はデータ・バイトと識別子フィールドで計算され、バージョン2.xのLINシステムで使用されます。



図35. LINバス・サポートの設定メニューとデコードされたフレーム

LIN デバッグのポイント

テクトロニクスのおシロスコープには、オプションでLINのシリアル・トリガ／解析が行えるものがあります（付録Aを参照）。前面パネルのBusボタンを使用することでLINバスを設定でき、LINのバージョン、ビット・レート、極性、スレッシュホールド、サンプル・ポイント（ビット時間のパーセントとして）などの基本パラメータを設定します。LINの設定メニューとデコードされたLINフレームの例を、図35に示します。

MSO/DPOシリーズには、最大4種類のバスを同時に定義してデコードするという強力な機能が備わっています。先にCANバスを例にとりましたが、ここではLINバスでウィンドウをコントロールする例を考えてみます。ドライバが助手席側の窓を下げるボタンを押すと、運転席ドアのLINバスでメッセージが発行され、中央のCANゲートウェイを通して助手席ドアのLINネットワークに送られます。この例では、バスの1つのメッセージにトリガし、3つのバスを同時に取込んでデコードします。これにより、トラフィックが1つのバスがシステムを通して別なバスに伝わる様子を驚くほど簡単に観測できます。図36にその様子を示します。この例では最初のLINメッセージにトリガし、3つのすべてのバスを取込んでいます。

MSO/DPOシリーズのLINトリガには、以下のような強力な機能があります。

- Sync – 同期フィールドにトリガします。
- Identifier – 特定の識別子にトリガします。
- Data – 特定のデータの1～8バイトまたはデータ・レンジにトリガします。
- Identifier & Data – 識別子データの組合せでトリガします。
- Wakeup Frame – ウェイクアップ・フレームにトリガします。
- Sleep Frame – スリープ・フレームにトリガします。
- Error – 同期エラー、IDパリティ・エラーまたはチェックサム・エラーにトリガします。

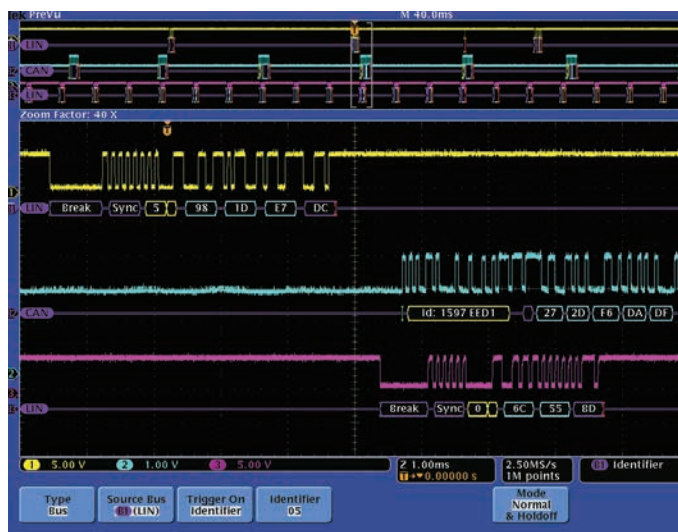


図36. 複数のオートモーティブ・シリアル・バスの同時取込みとデコードの例

これらのトリガにより、LINバスで特定のイベントを迅速に探し出すことができます。さらに、当社オシロスコープにはイベント・テーブルやサーチ/マークなどの優れたシリアル機能があり、LINをベースとしたオートモーティブ回路設計のデバッグが簡単に行えます。

Command Word	Sync	Terminal Address	T/R	Subaddress / Mode	Word Count / Mode Code	Parity
Bits	3	5	1	5	5	1

図37. MIL-STD-1553 コマンド・ワード・フォーマット

Data Word	Sync	Data (D15 – D0)	Parity
Bits	3	16	1

図38. MIL-STD-1553 データ・ワード・フォーマット

MIL-STD-1553

概要

MIL-STD-1553はコンピュータ業界のLANと同様の規格であり、航空アプリケーションのデータ通信のシリアル・バスにおける電気、プロトコルの特性を規定した軍用規格です。

MIL-STD-1553は、1973年、SAE (Society of Automotive Engineers) のA2-Kドラフト規格の開発から始まります。行政、軍のレビュー、改訂の後、1973年にMIL-STD-1553 (USAF) として発行されました。軍のすべての部門に対応するため、1975年にMIL-STD-1553Aが発行され、さらにMIL-STD-1553Bを発行し、コンポーネント製造メーカーが準拠した製品を製造できるようにBレベルで凍結しました。最新の変更はNotice 2として1986年に発行されており、動作特性を規定しています。この規格は、SAEによって商用ドキュメントAS15531として見ることができます。

この規格は米国の軍用アプリケーションで広く使用されていますが、公共交通機関、宇宙探査機、製造アプリケーションなどの民生用途でも使用されたり、NATOやその他の政府機関でも実装されています。

動作原理

MIL-STD-1553は、シールドされたツイストペア・ケーブル、Twinaxケーブルを使用し、最高10Mbpsのビット・レートで最大32の16ビット・データ・ワード・メッセージを非同期に伝送します。1553のネットワークでは、1本のケーブルで時分割多重された半二重通信を使用します。安全が重視されるアプリケーションでは、二重冗長バスが一般的に使用され、さらに信頼性の高い通信を可能にしています。直接カップリングまたは変圧器結合のために、マンチェスタIIバイフェイズ符号化が使用されています。マンチェスタ符号化はセルフクロックであり、ビット・シーケンスは独立し、DCバランスです。マンチェスタ符号化された信号情報にはゼロクロスの極性とタイミングが含まれているため、1553バスは大きな信号レベルの変動耐性があります。

MIL-STD-1553では、コマンド・ワード、データ・ワード、ステータス・ワードの3種類のワードが規定されています。すべてのワードは20ビット長であり、3ビットの同期フィールド、16ビットの情報フィールド、最後が奇数パリティ・ビットで簡単なエラー検出を行います。同期フィールドは正規のマンチェスタ信号ではなく、二番目のビット時間の途中で一回遷移します。コマンド/ステータス同期は途中で負に遷移し、データ同期では正に遷移します。

アクティブのバス・コントローラによって送られるコマンド・ワードは、リモート・ターミナルが実行すべき機能を規定します。16ビットの情報フィールドには、ターミナルを識別する5ビットのターミナル・アドレス、ターミナル、送信/受信ビット、サブアドレスまたはモードの5ビット、ワード・カウントまたはモード・コードの5ビットが含まれています。

データ・ワードはバス・コントローラまたはリモート・ターミナルによって送信され、最上位ビット (MSB) が最初に送信されます。

Status Word	Sync	Terminal Address	ME	IB	SR	Re-served	BCR	Busy	SF	DBA	TF	Parity
Bits	3	5	1	1	1	3	1	1	1	1	1	1

図39. MIL-STD-1553ステータス・ワード・フォーマット



図40. MIL-STD-1553バスのデコード表示

ステータス・ワードは、有効なメッセージに応答してリモート・ターミナルから返され、メッセージの受信を認識するか、またはリモート・ターミナルのステータスを伝えます。16ビットの情報フィールドの最初の5ビットは、ターミナル・アドレスです。残りのビットは、メッセージ・エラー、インスツルメンテーション・ビット、サービス・リクエスト、ブロードキャスト・コマンド受信、ビジー、サブシステム・フラッグ、動的バス制御受諾、ターミナル・フラッグなどの特殊なステータス情報です。

MIL-STD-1553 デバッグのポイント

テクトロニクスのおシロスコープには、MIL-STD-1553のシリアル・トリガ／解析機能を持ったものがあります(付録Aを参照)。

1553バスには、受動プローブを使用して任意のアナログ・チャンネル(MSOシリーズでは任意のデジタル・チャンネル)に簡単に接続でき、前面パネルのバス・ボタン、スクリーン・メニューからバス・パラメータを設定することができます。MIL-STD-1553バスの特定のイベントを特定するため、当社のオシロスコープは同期、ワード・タイプ、データ・ワードの値、パリティ・エラーにトリガすることができます。

DPO4AERO型アプリケーション・モジュールを使用することで、MIL-STD-1553のシリアル信号が容易に観測でき、解析し、他のイベントと相関をとることができます。

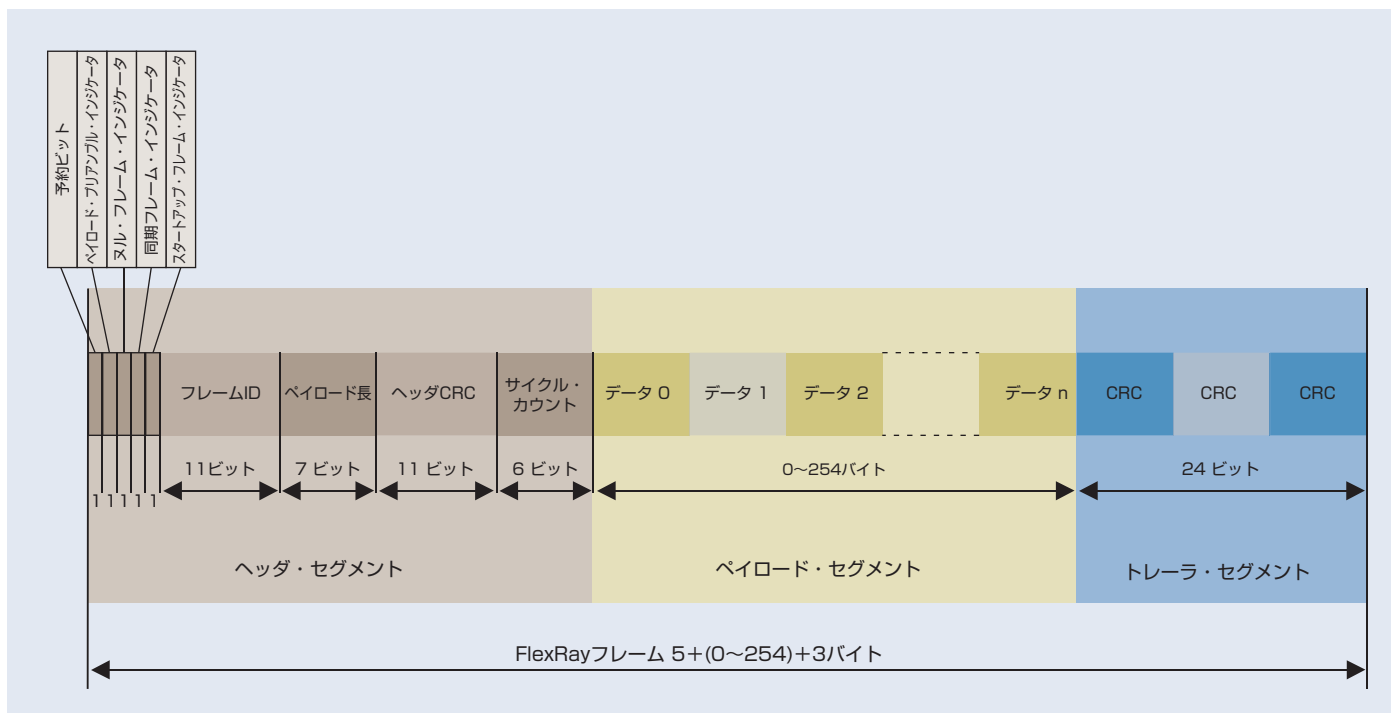


図41. FlexRayのフレーム構造

FlexRay

概要

FlexRayは比較的新しい自動車用のバスであり、現在でも業界トップクラスの自動車メーカーや伝送機器メーカーによって構成されるFlexRayコンソーシアムによって開発されています。自動車がますます洗練され、電気回路がますます自動車のアプリケーションに入り込むにつれ、CANやLINなど既存のオートモーティブ・シリアル規格では、ブレーキワイヤやステアワイヤなどのx-by-wireで必要とされるスピード、信頼性、冗長性が十分でないことがわかってきました。現在では、これらの機能は主にメカニカル・システムや油圧システムで実現されています。将来は、センサ・ネットワークや、自動車のコストを下げるだけでなく、予測ブレーキング、衝突回避、車間距離適応走行制御システムなど、電気回路がベースとなった機能によって同乗者の安全性を高めることのできる信頼性の高い電気回路に置き換わることになります。

動作原理

FlexRayは、シールド・ツイスト・ペア (STP) またはアンシールド・ツイスト・ペア (UTP) を使う差動バスで、最高10Mbpsのスピードがあります。これは、LINの20kbps、CANの1Mbpsに比べても際立った速度です。FlexRayはデュアル・チャンネル・アーキテクチャを使用しており、大きな利点が2つあります。2つのチャンネルを構成して、x-by-wireなど、安全性が重要となるアプリケーションにおいては冗長性の高い通信により確実にメッセージが届きます。それほど安全性が重要とされないアプリケーションでは、2つのチャンネルを構成して、各チャンネルに独自の情報を10Mbpsで送ることができ、全体のバス転送レートは20Mbpsになります。

FlexRayは、静的フレームと動的フレームの両方を含むコミュニケーション・サイクルによる同期および非同期プロトコルの利点を持ったタイム・トリガ・プロトコルを使用しています。静的フレームは、バスの各デバイスであらかじめ決められたタイム・スロットで、各サイクルで通信されます。各デバイスは、長さ(と時間)が変化するダイナミック・フレームにより各サイクル時に通信する機会も与えられます。FlexRayのフレームは、ヘッダ・セグメント、ペイロード・セグメント、トレーラ・セグメントという3つのセグメントで構成されます。それぞれのセグメントは、図41に示すように独自のコンポーネントを持っています。

ヘッダ・セグメントのコンポーネント：

- Indicator Bits – 最初の5ビットはインジケータ・ビットと呼ばれ、転送されるフレームのタイプを示し、ノーマル、ペイロード、ヌル、シンク、スタートアップがあります。
- Frame ID – フレームIDはフレームが転送されるスロットを規定します。フレームIDは1～2047の値をとり、使用される個々のフレームIDはコミュニケーション・サイクルにおいて各チャンネルで一度以上使用されます。
- Payload Length – ペイロード長フィールドは、ペイロード・セグメントで使用されるデータ・ワード数を示します。
- CRC – 同期フレーム・インジケータ、スタートアップ・フレーム・インジケータ、フレームID、ペイロード長から計算されるCRC (Cyclic Redundancy Check、周期的冗長チェック)コードです。
- Cycle Count – 現在のコミュニケーション・サイクルの値で、0～63の値をとりま。

ペイロード・セグメントのコンポーネント：

- Data – 最大254バイトのデータを含むデータ・フィールドです。静的セグメントで転送されるフレームでは、ペイロード・セグメントの最初の0～12バイトは、オプションでネットワーク管理ベクタとして使用できます。フレーム・ヘッダのペイロード・プリアンブル・インジケータは、ペイロード・セグメントがネットワーク管理ベクタを含んでいるかどうかを示します。動的セグメントで転送されるフレームでは、ペイロード・セグメントの最初の2バイトは、オプションでメッセージIDフィールドとして使用でき、ノードを受信し、フィールドの内容に応じてデータをフィルタまたは導きます。フレーム・ヘッダのペイロード・プリアンブル・インジケータは、ペイロード・セグメントがメッセージIDを含んでいるかどうかを示します。

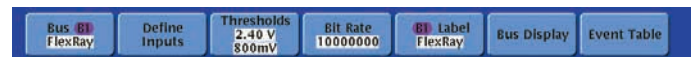


図42. FlexRayバス・サポートの設定メニュー

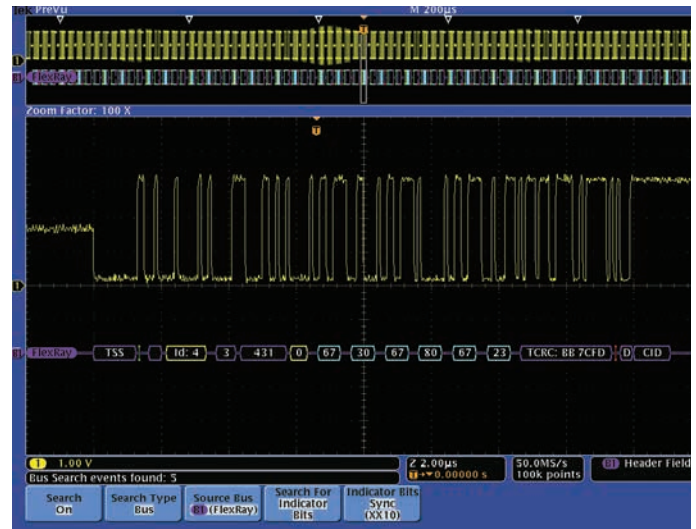


図43. フレームIDとサイクル・カウントでトリガし、取込んだデータからスタートアップ・フレームをサーチした例

トレーラ・セグメントのコンポーネント：

- CRC – フレームのヘッダ・セグメントとペイロード・セグメントのすべてのコンポーネントから計算したCRC (Cyclic Redundancy Check、周期性冗長チェック)コードです。

動的フレームには、トレーラのCRCの後に、DTS (Dynamic Trailing Sequence、動的トレーリング・シーケンス)と呼ばれるコンポーネントが追加されており、バス・レシーバによる早過ぎるチャンネル・アイドル検出を防ぐことができます。

FlexRay デバッグのポイント

テクトロニクスのおシロスコープには、FlexRayのシリアル・トリガ／解析機能を持ったものがあります（付録Aを参照）。FlexRayバスを設定するには、バス・メニューからFlexRayを選択します。FlexRayの設定メニューを図42に示します。

次に、Define Inputsメニューから、FlexRayのチャンネルAまたはチャンネルBを選択し、プロービングする信号の種類（差動、差動ペアの半分、またはコントローラとバス・ドライバ間のロジック信号）を選択し、スレッシュホールドとビット・レートを設定します。MSO/DPO4000シリーズでサポートされる他のシリアル規格と違い、FlexRayは3値レベルのバスであるため、Tx/Rxでない信号を観測するためには2つのスレッシュホールドが必要です。これにより、オシロスコープはハイのデータ、ローのデータだけでなく、両方の信号が同じ電圧であるアイドル・ステートを認識することもできます。

当社オシロスコープの強力なFlexRay機能の例を図43に示しています。この例では、フレームID=4とサイクル・カウント=0の組合せでトリガし、約80のFlexRayフレームを取込み、すべての取込みをデコードし、オシロスコープのサーチ機能を使ってすべての同期フレームを検出してマークを付けています。しかも、わずか100,000ポイントのレコード長で実現しています。MDO4000シリーズ、MSO/DPO4000Bシリーズには最大で20Mポイント（全チャンネル）のメモリ長があるため、シリアル信号の長い時間ウィンドウでも取込むことが可能です。

当社オシロスコープのFlexRayトリガには、以下のような強力な機能があります。

- Start of Frame — FSS (Frame Start Sequence) の立上りエッジにトリガします。
- Indicator Bits — ノーマル、ペイロード、ヌル、シンク、またはスタートアップ・フレームにトリガします。
- Identifier — 特定のフレームIDまたはフレームIDの範囲でトリガします。
- Cycle Count — 特定のサイクル・カウントの値またはサイクル・カウントの値の範囲でトリガします。
- Header Fields — インジケータ・ビット、フレームID、ペイロード長、ヘッダCRC、サイクル・カウントを含む、任意またはすべてのヘッダ・フィールドの値の組合せにトリガします。

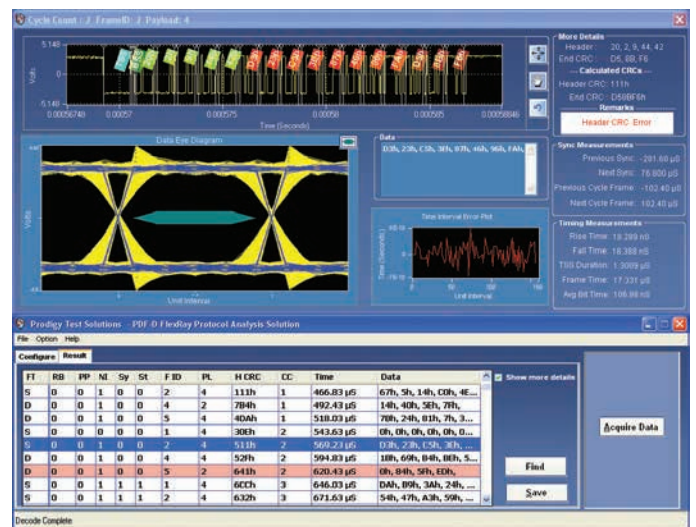


図44. FlexRay信号のアイ・ダイアグラム解析

- Data — 16バイトまでのデータにトリガします。データ・ウィンドウは、非常に長いデータ・ペイロードのフレーム内でユーザ設定のバイト数だけオフセットすることができます。特定の値または特定の値の範囲としてデータを指定することができます。
- Identifier & Data — フィールドIDとデータの組合せでトリガします。
- End of Frame — 静的フレーム、動的フレームまたはすべてのフレームにトリガします。
- Error — ヘッダCRCエラー、トレーラCRCエラー、ヌル・フレーム・エラー、同期フレーム・エラー、スタートアップ・フレーム・エラーなど、さまざまなエラー・タイプにトリガします。

DPO4AUTOMAX型には、トリガ、デコード機能の他にFlexRay信号のアイ・ダイアグラム解析機能もあり、物理レイヤ問題の診断に使用できます。PCにソフトウェア・パッケージをロードし、LANまたはUSB経由でオシロスコープと接続し、Acquire Dataボタンを押すと、図44に示すようなさまざまな情報が表示されます。

解析項目を以下に示します。

- アイ・ダイアグラム — 取込んだすべてのメッセージから作成され、選択されているフレームは青でハイライト表示されます。TP1またはTP4のマスクと簡単に比較でき、違反は赤でハイライト表示されます。
- デコード — 選択されているフレームがデコードされ、アナログ波形とともに表示されます。一方、取込まれたすべてのフレームはデコードされて下の表に表示されます。

- TIE (Time Interval Error) プロット – フレーム内のジッタを視覚的に把握することができます。
- エラー・チェック – エラーは赤でハイライト表示されます。ヘッダCRCとトレーラCRCは計算され、転送されたフレームの情報と比較されます。
- タイミング測定 – 立上り時間、立下り時間、TSS期間、フレーム時間、平均ビット時間、前のシンク、後のシンク、前のサイクル・フレーム、後のサイクル・フレーム
- Find – パケット内容をもとに、特定のフレームを検出します。
- Save – デコードされた結果を、オフラインでの解析のために .CSVファイルに保存します。

先に説明したCAN、LINの機能に加え、統合的なFlexRayソリューションにより、当社オシロスコープはオートモーティブ回路設計の究極のデバッグ・ツールとなります。



図45. CANバスの取込みから、指定した識別子とデータで検索した結果の例

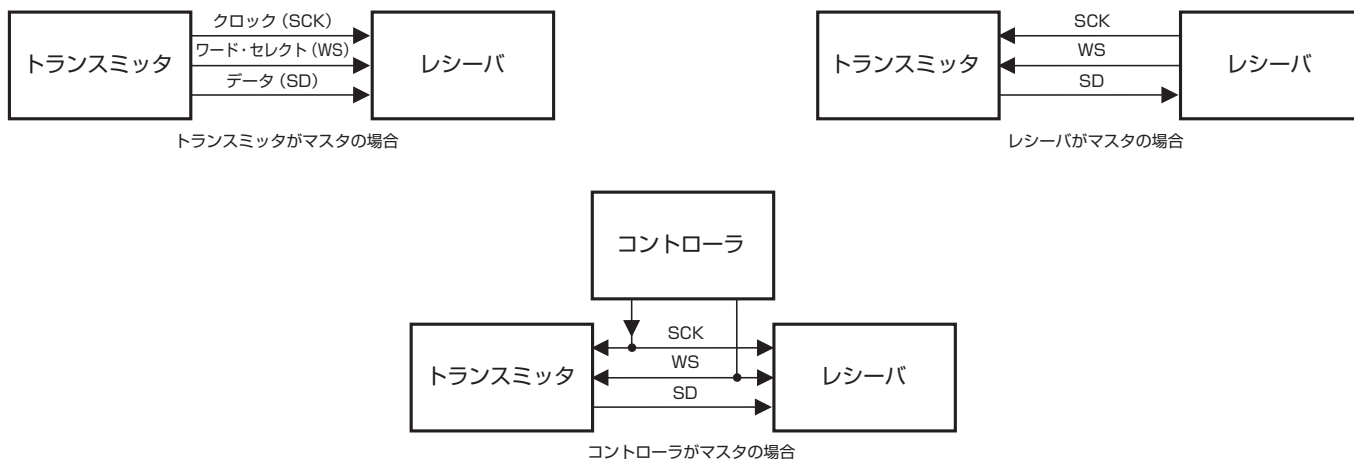


図46. 差動のI2Sバスの構成

オーディオ・バス

概要

I2S (アイ・スクエア・エス) は、Inter-IC Soundの略で、CDプレーヤやデジタル・テレビなどの民生機器のためのデジタル・オーディオ信号の標準化通信バスとして、1980年代中頃にフィリップス社によって開発されました。その後、民生家電の市場で20年以上にわたって改良され、I2Sバスとなりました。今日では、携帯電話、MP3プレーヤ、セットトップ・ボックス、業務用オーディオ機器、ゲーム機などで使用されています。

動作原理

I2Sバスは、マスタ/スレーブによる3線式シリアル通信バスです。クロック (SCK)、ワード・セレクト (WS)、データ (SD) という3つの信号からなります。一般的に、トランスミッタはマスタであり、レシーバはスレーブになります。しかし、場合によってはレシーバがクロックとワード・セレクト信号を出力することからマスタになることもあります。あるいは、トランスミッタとレシーバは他のデバイスからコントロールされることもあります。バスの構成を図46に示します。

シリアル・データは、MSB (最上位ビット) を先頭にして (2の補数形式で) 伝送されます。トランスミッタとレシーバではワード長が異なることがあるため、MSBを先に送ります。トランスミッタは、レシーバが処理できるビット長を知る必要はなく、レシーバも伝送されるビット数を知る必要もありません。伝送されるシステムのワード長がトランスミッタの扱えるワード長よりも長くなると、伝送されるデータのワードは短く切り詰められます (LBSは'0'にセットされます)。レシーバで扱えるワード長以上のビットがレシーバに送られてくると、レシーバはLSB以降のビットを無視します。一方、ワード長より短いビットがレシーバに送られると、不足のビットは内部で0にセットされます。したがって、MSBは固定の位置になり、一方、LSBの位置はワード長によって異なります。トランスミッタは、WSが変化した1クロック周期後にMSBを送ります。

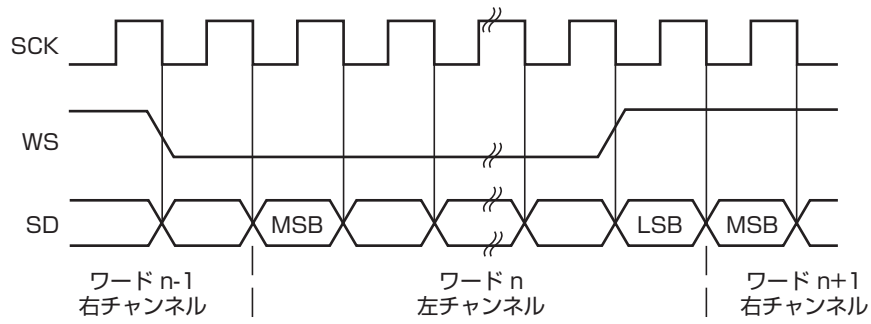


図47-1. PSのフォーマット

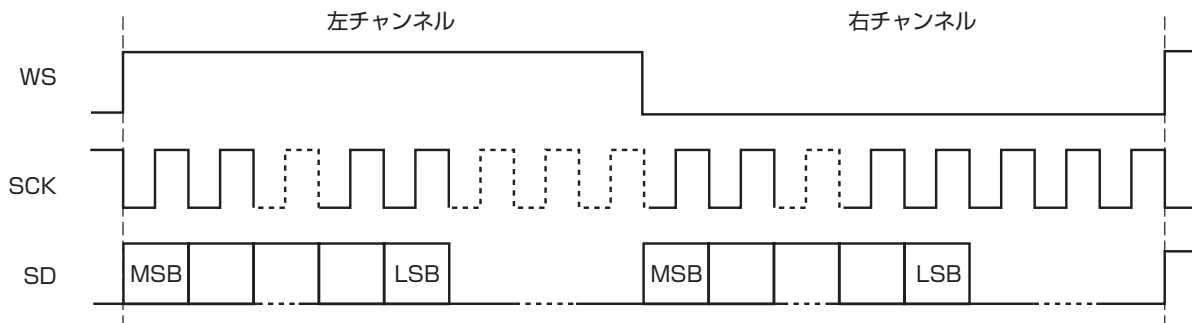


図47-2. LJのフォーマット

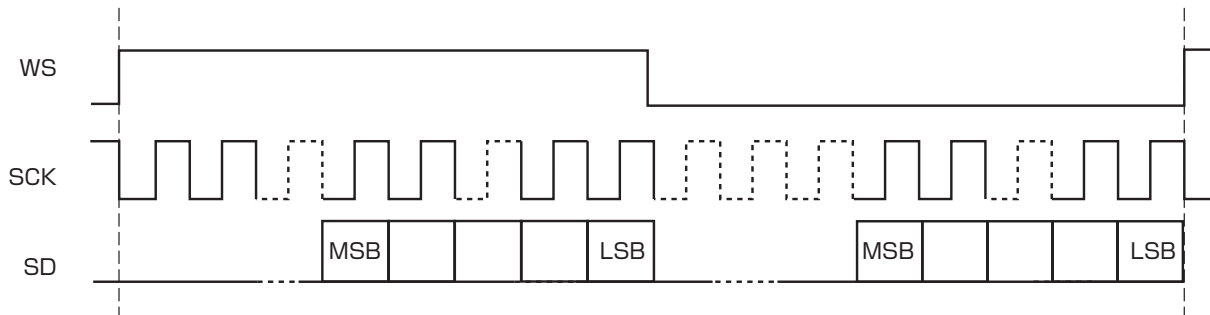


図47-3. RJのフォーマット

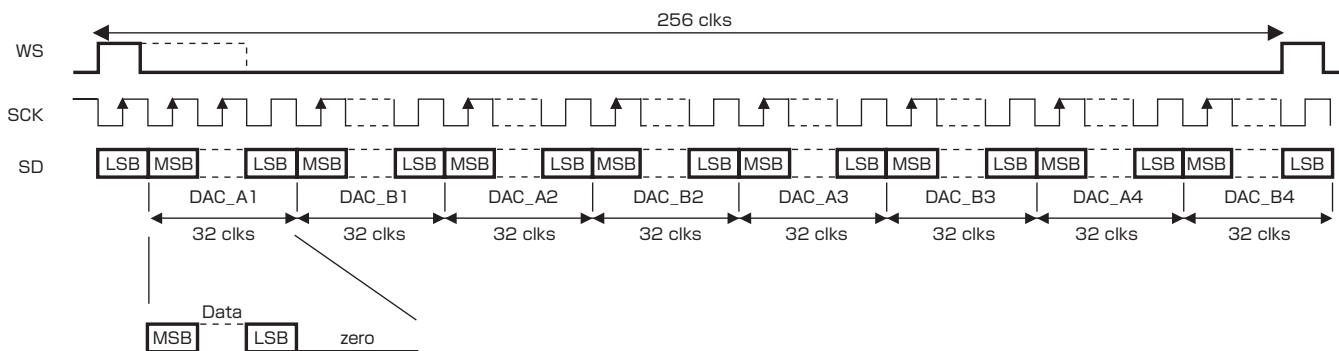


図48. TDMのフォーマット

I²Sバスには、LJ (Left Justified)、RJ (Right Justified)、TDM (Time Division Multiplexing) などと呼ばれるバリエーションもあります。I²S、LJ、RJの違いは、WS信号に対して時間的にどこにデータが置かれるかによります。I²Sでは、MSBはWSの1クロック後になります。LJではデータ・ビットとWSは一致し、RJではデータ・ビットはWSの右に揃います。この様子を図47に示します。

TDMはI²S、LJ、RJに似ていますが、2チャンネル以上が必要です。図48の例では、32データ・ビット、8チャンネルの様子を示しています。

これらのデジタル・オーディオ・バスは、非常に簡単なデータ構造になっています。このアプリケーション・ノートで説明されているその他のバスの多くには、アドレス・フィールド、CRCフィールド、パリティ・ビット、スタート/ストップ・ビット、インジケータ・ビットなどがありますが、デジタル・オーディオ・バスには各チャンネルのデータ値があるだけです。

デジタル・オーディオ・バスのデバッグのポイント

デジタル・オーディオ・バスのデバッグには、DPOxAUDIO型デジタル・オーディオ・トリガ/解析モジュールを使用します。オシロスコープ前面パネルにあるBusボタンを押してデジタル・オーディオ・バスを選択し、ワード・サイズ、信号の極性、ビットの順序、スレッシュホールドなどを設定します。TDMでは、チャンネルごとのデータ・ビット数、チャンネルごとのクロック・ビット数、ビット遅延、フレームごとのチャンネル数を設定します。

バスの設定が終わると、バスの特定のデータ内容にトリガしてすべての取込みをデコードし、特定のデータをすべての取込みデータの中から検索することができます。次の例では、ADコンバータにおけるI2Sバスを観測しています。Ch1(黄)はクロック信号、Ch2(青)はワード・セレクト信号、Ch3(ピンク)はデータ信号です。設定範囲を外れるデータ値が発生したときにトリガがかかるように設定し、サンプリングしている信号がADコンバータのリミットにかかっていないかを調べます。図49は、Outside Range(設定範囲外)のトリガで-128の値を取込んだ例を示します。

デジタル・オーディオ・トリガの種類を以下に示します。

- Word Select — I2S、LJ、RJバスのフレームを開始するWSのエッジにトリガします。
- Frame Sync — TDMでフレームの開始を示すフレーム・シンク信号にトリガします。
- TDM Data — I2S、LJ、RJのLeft Word、Right Word、またはEither Word範囲内にあるデータからユーザが任意に設定したデータにトリガします。TDMでは、チャンネル番号を設定します。データ設定では、=、≠、≤、<、>、≥、設定範囲内、設定範囲外が設定できます。

当社オシロスコープでサポートされている他のシリアル・バスと同様、これらのトリガ・タイプはWave Inspectorの検索条件としても利用でき、デコードされたデジタル・オーディオ・データはイベント・テーブル形式で表示することもできます。



図49. I2Sバスにおいて設定範囲外の値にトリガした例

MIPI DSI-1/CSI-2バス

概要

このアプリケーション・ノートで説明している規格の多くは、登場以来すでに10年近く経っていますが、MIPI (Mobile Industry Processor Interface) は比較的新しい規格であり、場合によっては今でも進歩しています。MIPIアライアンス (www.mipi.org) では、次のように謳っています。

“MIPIは、新しい技術を使用し、新しい機能、サービスの迅速な採用を可能にするハードウェアとソフトウェア・インタフェースの規格を策定しています。” “モバイル業界は多くのインタフェースに互換性がなく、差動でないことに悩んでいます。互換性がないために機器間も互換性がなく、複数のインタフェース技術を保つために無駄なエンジニアリング投資が行われ、高いコストとなっています(そうでありながら高いマージン/価値になっていません)。MIPIは、技術的および知的財産を持った魅力的なターゲットを開発することにより、この無駄を削減します。”

MIPIアライアンスはいくつかの仕様を策定し、数多くのモバイル製品で採用されています。そのうちの2つがDSI-1とCSI-2であり、ホスト・プロセッサとディスプレイ・チップ間(DSI-1)とホスト・プロセッサとカメラ・チップ間(CSI-2)の情報の転送方法を規定したプロトコルレベルの仕様です。いずれのプロトコルも、MIPIアライアンスで開発されたD-PHY、M-PHYという物理レイヤ・インタフェースを利用しています。

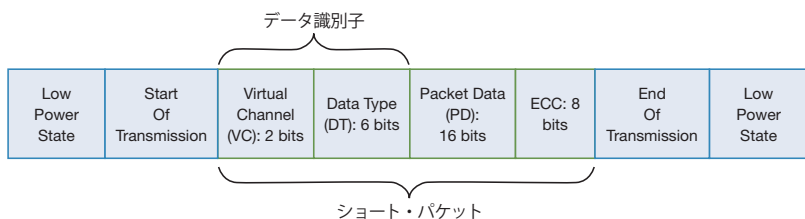


図50. ショート MIPI パケット

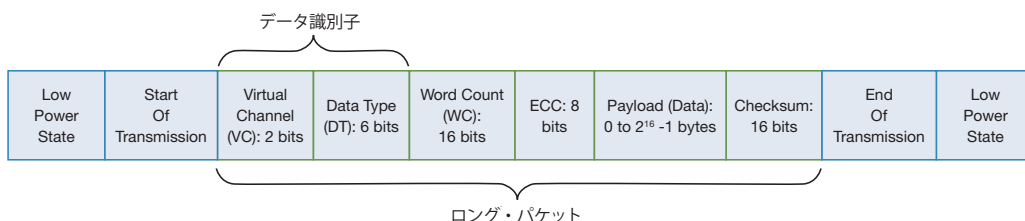


図51. ロング MIPI パケット

動作原理

D-PHYの物理レイヤは、ホスト・プロセッサとディスプレイ、カメラなどのデバイス間のハイスピード・シリアル・リンクを規定しています。最少のバス構成は1つのクロック・レーンと1つの信号データ・レーンですが、バスの帯域によっては最大4データ・レーンまで使用できます。

バスは、ローパワー、ハイスピードの2種類のモードで動作します。ローパワー・モードでは、シングルエンド・シグナリングを使用し、クロックをデータに埋め込みます。主にコマンドとコントロールに使用し、最高データ転送レートは10Mbpsです。ハイスピード・モードでは差動シグナリングであり、主に高速のデータ転送に使用します。例えば、携帯電話のディスプレイの垂直、水平同期情報は比較的小さな情報を転送すればよいいため、また転送レートも低速でよいいため、ローパワー・モードで転送します。しかし、実際のビデオ・コンテンツを最新の高分解像度ディスプレイで表示させるためには大容量、高速のデータ転送が必要になるため、ハイスピード・モードを使用します。ハイスピード・モードの実際の最高転送レートは実装ベースの値ですが、一般的にレーンごとに80Mbps~1Gbpsのレンジで動作します。

DSI-1、CSI-2のプロトコルでは、ショート・パケット、ロング・パケットを使用してD-PHY物理レイヤで情報を転送します。

ショート・パケットは主に同期や設定などのコマンド、コントロール情報で、ロング・パケットは主にビデオ・コンテンツで使用されます。ショート・パケットの構成は次の通りです。

- Data Identifier — 8ビットで構成され、バーチャル・チャンネルとデータ・タイプ・フィールドが含まれています。
- Virtual Channel — 複数のカメラ、ディスプレイ・デバイスがバス上にある場合、どちらのデバイスのパケットであるかを指定します。2ビットを使用し、1つのバスで4つまでのデバイスが共有できます。
- Data Type — 6ビットを使用し、どのような種類のコマンド、アクションが転送されるのか、パケット・データ・フィールドがどのようなデータであるのか、さらにどのように構成されているのかを指定します。
- ECC — エラー訂正フィールドであり、ショート・パケットの1ビット・エラーを訂正し、2ビット・エラーを検出します。

ロング・パケットにはさらにフィールドがあり、以下のような構成になっています。

バーチャル・チャンネル、データ・タイプ、ECCはショート・パケットと同じですが、ショート・パケットとの違いを次に示します。

- Word Count — 16ビットの値でペイロード・データに含まれるバイト数を指定します。
- Payload — 主に、さまざまなビデオ・フォーマットで大容量のビデオ・データを送る場合に使用します。フォーマットごとにData Typeが異なります。Payloadフィールドは、0~65,535 (2¹⁶ - 1) バイトの値をとります。
- Checksum — ペイロードのエラーをチェックします。

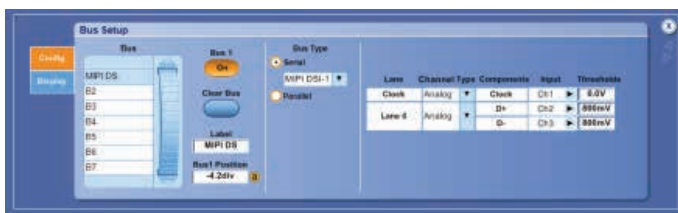


図52. MIPI DSI-1バスの設定例



図54. MSO70000Cシリーズで表示されたPacked Pixel Streamのアナログ・チャンネル・デコード表示 - チェックサム・エラーが確認できる

DSI-1、CSI-2のデバッグ・ポイント

SR-DPHYは、DSI-1、CSI-2のバスをデコードします。バスを設定するには、まずBus Setupメニューを選択し、次にSerialを、さらにMIPI DSI-1またはMIPI CSI-2を選択します。この例では、DSI-1が選択されています。

バスを設定する場合、チャンネルの種類（アナログ、デジタル）と、バスをプロービングするプローブの種類を設定する必要があります。アナログ・チャンネルを使用する場合は差動プローブでクロックを、2本のシングルエンド・プローブでデータ・レーンをプロービングします。MSO70000CシリーズでP6780型差動ロジック・プローブを使用すると、デジタル・チャンネルで複数のレーンをプロービングできます。1つのチャンネルでクロックを、1つのチャンネルでD+/D- 差動信号を、1つのチャンネルでD+/GNDのシングルエンド信号を、1つのチャンネルでD-/GNDのシングルエンド信号をプロービングします。このように、1つのレーンのDSI-1またはCSI-2をプロービングするには、3つのアナログ・チャンネルと4つのデジタル・チャンネルが必要になります。

設定して表示すると、バス形式表示でショート・パケット、ロング・パケットの他、BTA (Bus Turn Around)、エスケープ・モード・コマンドなどの通信タイプのコンポーネントがデコード表示されます。



図53. MSO70000Cシリーズで表示された、Packed Pixel Stream、24ビットRGB 8-8-8フォーマット・ロング・パケットのデジタル・チャンネル・デコード表示

デフォルトでは、Virtual ChannelとWord Countのフィールドは10進で表示されます。Data Typeのフィールドは、パケットの種類が表示されます。ECCとChecksumは16進で表示されます。さらに、設定されたデータの種類の、ペイロード・フィールドには転送されるデータがわかりやすい表示タイプの情報としてデコードされます。この例では、転送される最初のピクセル値のペイロードは、Redが255、Greenが216、Blueが0と表示されています。

DSI-1、CSI-2をデコードするだけでなく、ロング・メモリをサーチして以下のようなすべてのパケット内容を検出します。

- ショート・パケット (VC、DT、Direction、Packet Dataの値)
- ロング・パケット (VC、DT、Direction、WC、ピクセル値を含む Data Payload)
- 終了
- 伝送開始 (SoT)
- 伝送終了 (EoT)
- バス・ターン・アラウンド (DSI-1のみ)
- エスケープ・モード
- ECC警告
- ECCエラー
- チェックサム・エラー

SR-DPHYアプリケーションとテクトロニクスのおシロスコープを使用することで、DSI-1、CSI-2のシリアル・バスが容易にデコード、観測でき、解析し、他のデバイスの動きと相関をとることができます。

トリガとサーチ

このアプリケーション・ノートで説明してきたように、シリアル・バスで特定のイベントを検出するには、優れたトリガ機能が必要になります。しかし、データを取込み終わった後（オシロスコープは取込みを停止）解析しようと思っても、もはやトリガは機能しません。取込み終わった波形データを解析するためのトリガのような機能があれば、と思いませんか。

MSO/DPOシリーズに搭載された、強力なサーチ機能を持つWave Inspector[®]は、これを実現しています。このアプリケーション・ノートで説明したすべてのバス・トリガ機能は、取込み終わったデータにおいてもサーチ条件として利用できます。

例えば、図45に示すように、オシロスコープは、長いレコード長から、特定のアドレス、データですべてのCANメッセージをサーチし、波形の上部に白い三角矢印のマーカーを付けます。サーチされたイベントは、前面パネルにある左右の矢印ボタンを押すことで簡単にジャンプできます。サーチ条件では、従来のトリガ・タイプ（エッジ、パルス幅、ラント、セットアップ／ホールド時間、ロジック、立上り／立下り時間）も利用できます。

まとめ

組込みシステム設計において、パラレル・バスからシリアル・バスへ移行することは多くの利点がありますが、同時に、設計エンジニアは多くの課題に直面することにもなります。従来の測定器では、測定したいイベントにトリガすることは難しく、アナログ信号の観測だけでは、どのような情報が表示されているかを知ることはほとんど不可能です。また、長いバス信号を手作業でデコードし、問題点を診断することは、非常に時間のかかる、またエラーを起こしやすい作業でもあります。MSO/DPOシリーズには強力なトリガ機能、デコード機能、サーチ機能などが備わっており、組込み設計の問題を効率的に解決することができます。

付録 A

テクトロニクスは豊富な機種を用意しており、最適な一台をお選びいただけます。

	MSO/DPO70000 シリーズ	DPO7000C シリーズ	5シリーズMSO	MSO/DPO5000 シリーズ	MDO4000C シリーズ	MDO3000 シリーズ	MSO/DPO2000シ リーズ
周波数帯域	33GHz、25GHz、 23GHz、20GHz、 16GHz、12.5GHz、 8GHz、6GHz、 4GHz	3.5GHz、2.5GHz、 1GHz、500MHz	2GHz、1GHz、 500MHz、 350MHz	2GHz、1GHz、 500MHz、 350MHz	1GHz、500MHz、 350MHz、 200MHz	1GHz、500MHz、 350MHz、 200MHz、 100MHz	200MHz、 100MHz、70MHz
アナログ・ チャンネル数	4	4	4、6、8	4	4	2または4	2または4
デジタル・ チャンネル数	16 (MSO)	--	8~64 (オプション)	16 (MSO)	16 (オプション)	16 (オプション)	16 (MSO)
スペクトラム・ アナライザ・ チャンネル数	--	--	--	--	1 (オプション)	1	--
レコード長 (全チャンネル、 ポイント)	最大62.5M (標準) 最大250M (オプション)	25M (標準) 最大125M (オプション)	62.5M (標準) 125M (オプション)	25M (標準) 最大125M (オプション)	20M	10M	1M
サンプル・レート (アナログ)	最高100GS/s	最高40GS/s	最高6.25GS/s	最高10GS/s	最高5GS/s	最高5GS/s	1GS/s
カラー・ ディスプレイ	12.1型XGA	12.1型XGA	15.6型HD	10.4型XGA	10.4型XGA	9型WVGA	7型WQVGA
シリアル・バス・ トリガ／解析	I ² C SPI RS-232/422/485/ UART CAN LIN FlexRay USB 2.0/3.1 Gen1 HSIC 10/100BASE-T Ethernet MIL-STD-1553 8B/10Bデコード D-PHY MIPI デコード PCI Express デコード	I ² C SPI RS-232/422/485/ UART CAN LIN FlexRay USB 2.0 HSIC 10/100BASE-T Ethernet MIL-STD-1553 8B/10Bデコード D-PHY MIPI デコード PCI Express デコード	I ² C SPI RS-232/422/485/ UART CAN CAN FD LIN FlexRay USB 2.0 HSIC 10/100BASE-T Ethernet I ² S/LJ/RJ/TDM ARINC 429 MIL-STD 1553	I ² C SPI RS-232/422/485/ UART CAN LIN FlexRay USB 2.0 HSIC 10/100BASE-T Ethernet MIL-STD-1553 8B/10Bデコード D-PHY MIPI デコード PCI Express デコード	I ² C SPI RS-232/422/485/ UART CAN CAN FD LIN FlexRay USB 2.0 HSIC 10/100BASE-T Ethernet I ² S/LJ/RJ/TDM ARINC 429 MIL-STD 1553	I ² C SPI RS-232/422/485/ UART CAN CAN FD LIN FlexRay USB 2.0 HSIC 10/100BASE-T Ethernet I ² S/LJ/RJ/TDM ARINC 429 MIL-STD-1553	I ² C SPI RS-232/422/485/ UART CAN LIN
同時に表示可能なシ リアル・バス数	16	16	原則、無制限	16	3	2	2

お問い合わせ先：

オーストラリア 1 800 709 465
オーストリア 00800 2255 4835
バルカン諸国、イスラエル、南アフリカ、その他ISE諸国 +41 52 675 3777
ベルギー 00800 2255 4835
ブラジル +55 (11) 3759 7627
カナダ 1 800 833 9200
中央／東ヨーロッパ、バルト海諸国 +41 52 675 3777
中央ヨーロッパ／ギリシャ +41 52 675 3777
デンマーク +45 80 88 1401
フィンランド +41 52 675 3777
フランス 00800 2255 4835
ドイツ 00800 2255 4835
香港 400 820 5835
インド 000 800 650 1835
インドネシア 007 803 601 5249
イタリア 00800 2255 4835
日本 81 (3) 6714 3086
ルクセンブルク +41 52 675 3777
マレーシア 1 800 22 55835
メキシコ、中央／南アメリカ、カリブ海諸国 52 (55) 56 04 50 90
中東、アジア、北アフリカ +41 52 675 3777
オランダ 00800 2255 4835
ニュージーランド 0800 800 238
ノルウェー 800 16098
中国 400 820 5835
フィリピン 1 800 1601 0077
ポーランド +41 52 675 3777
ボルトガル 80 08 12370
韓国 +82 2 6917 5000
ロシア +7 (495) 6647564
シンガポール 800 6011 473
南アフリカ +41 52 675 3777
スペイン 00800 2255 4835
スウェーデン 00800 2255 4835
スイス 00800 2255 4835
台湾 886 (2) 2656 6688
タイ 1 800 011 931
イギリス、アイルランド 00800 2255 4835
アメリカ 1 800 833 9200
ベトナム 12060128

2016年4月現在



jp.tek.com

テクトロニクス／ケースレーインスツルメンツ

お客様コールセンター：技術的な質問、製品の購入、価格・納期、営業への連絡

TEL: 0120-441-046 ヨク良い オシロ 営業時間／9:00～12:00・13:00～18:00
(土日祝日および当社休日を除く)

サービス・コールセンター：修理・校正の依頼

TEL: 0120-741-046 なんと良い オシロ 営業時間／9:00～12:00・13:00～17:30
(土日祝日および当社休日を除く)

〒108-6106 東京都港区港南2-15-2 品川インターシティB棟6階

記載内容は予告なく変更することがありますので、あらかじめご了承ください。

Copyright © 2017, Tektronix. All rights reserved. TEKTRONIX およびTEK はTektronix, Inc. の登録商標です。
記載された製品名はすべて各社の商標あるいは登録商標です。

2017年12月 48Z-19040-21