EZ-USB™ FX3S を使用した USB RAID 1 ディスク設計

本書について

適用範囲と目的

AN89661 では、EZ-USB™ FX3S™ を使用して USB RAID (Redundant Array of Independent Disks) レベル 1 ディスクを設計および実装する方法について説明します。これには、サーバーブートストレージアプリケーション用の USB RAID 1 ディスクの設計例が含まれており、サーバーブートストレージソリューションの開発に役立ちます。

目次

本書について 1
目次 2
1 はじめに 3
2 RAID の紹介 4
3 システム概要 5
4 機能概要 7
5 ハードウェア 8
6 ファームウェア 9
6.1 ファームウェアソース 9
6.2 コールバック操作 12
6.3 大容量記憶装置操作 13
6.4 自動再構築 15
7 操作手順 17
8 パフォーマンス 18
8.1 USB⇔SD カード 18
8.2 SD カード⇔SD カード 18
9 まとめ 19
10 付録 A: FX3S RAID オンチップ USB ブート ディスク基板レイアウト ビュー 20
11 付録 B: FX3S RAID-on-Chip USB ブート ディスクの回路図 21
改訂履歴 22
免責事項 23

1 はじめに

USB 規格では、標準化された単一のインターフェースを使用して多くの周辺機器を接続できます。信頼性, 拡張性, 高速性, 低コスト, 低電力性, ホットプラグ対応のインターフェースとして長年にわたって証明されてきた USB は、現在、さまざまなアプリケーションでユビキタスになっています。このアプリケーションノートでは、そのようなアプリケーションの 1 つであるサーバーの RAID ブートディスクを取り上げます。

サーバー アプリケーションの中で、サーバー仮想化は IT のトレンドであり、サーバー市場で最も急速に成長しているセグメントになっています。サーバー仮想化は、ソフトウェアを使用して物理サーバーを複数の仮想サーバーに分割するプロセスです。サーバーの仮想化には、メイン サーバー ストレージから分離された、高速で信頼性が高くコンパクトなブート ディスクが必要です。

EZ-USB™ FX3S は、デュアル セキュア デジタル (SD) インターフェースを統合した RAID-on-Chip コントローラで、低コストの RAID ブートディスクに最適です。SuperSpeed USB 大容量ストレージデバイスの利点と機能を提供するだけでなく、RAID-on-Chip 機能を通じてストレージの冗長性を管理し、高い信頼性を実現します。

このアプリケーションノートでは、サーバーブートストレージアプリケーション用の EZ-USB™ FX3S を使用した USB RAID 1 ディスクの設計例を紹介します。RAID 1 機能は、FX3S 内で完全に制御されます。このアプリケーションは、2 つの SD カードを使用するスタンドアロンの大容量記憶装置としても使用できます。FX3S RAID-on-Chip ドングル キットは、市販で購入できます。完全なソースコードは、このアプリケーションノートの付録で入手できます。

2 RAID の紹介

RAID は、複数のストレージディスクコンポーネントを論理ユニットに結合するストレージテクノロジーです。データは、必要な冗長性とパフォーマンスの量に応じて、「RAID レベル」と呼ばれるいくつかの方法のいずれかでディスクに分散されます。さまざまなスキームまたはアーキテクチャーは、「RAID レベル」、または単に「RAID」と名付けられ、その後に数字が続きます。FX3S は RAID 0 と RAID 1 をサポートします。

RAID 0 (ブロックレベルストライピング) には冗長性がありません。2 つの物理ストレージ コンポーネント間でデータをインターリーブするため、パフォーマンスは向上しますが、フォールト トレランスは向上しません。ディスクに障害が発生すると、ストレージアレイが破壊されます。通常、RAID 0 アーキテクチャーは 2 つのディスクをサポートします (Figure 1 を参照)。

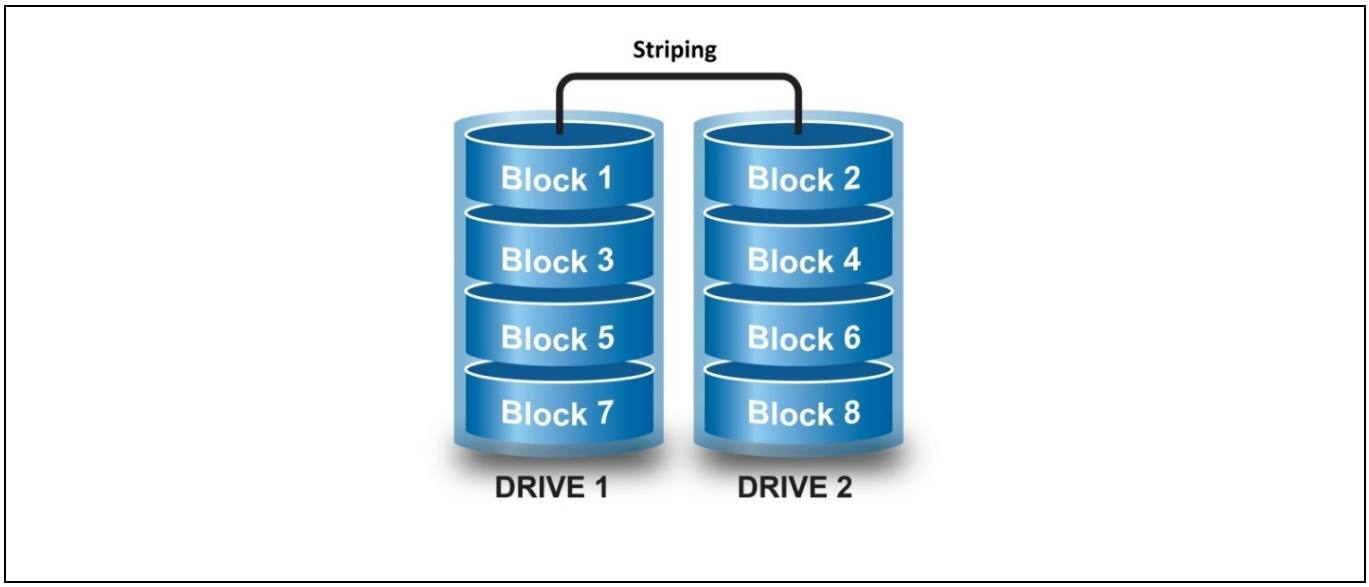


Figure 1 RAID レベル 0

RAID 1 (ミラーリング) の場合、データは 2 つのディスクに同じように書き込まれ、Figure 2 に示すように「ミラーセット」が生成されます。要求されたデータを含む 2 つのディスクのいずれかが読み出し要求を処理します。書き込み要求は、両方のディスクのストライプを更新します。このようなアレイを作成するには、少なくとも 2 つのディスクが必要です。

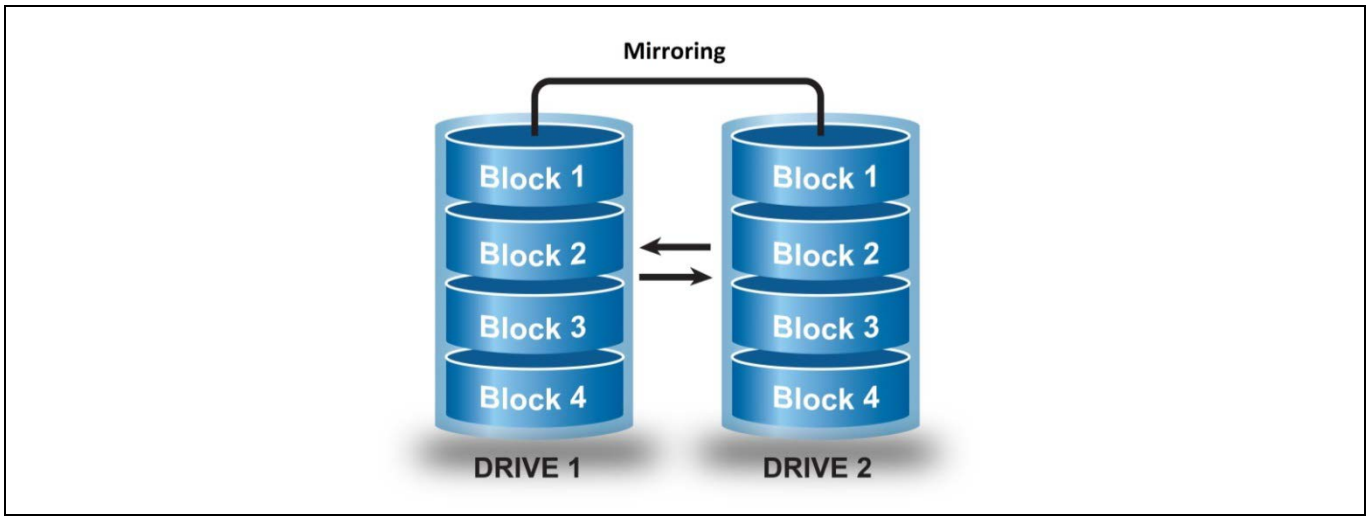


Figure 2 RAID レベル 1

3 システム概要

Figure 3 は、RAID ブートディスクを使用するサーバーの高レベルのブロック図です。サーバー仮想化の出現以来、仮想化ソフトウェアを使用して複数の仮想サーバーをホストするために、単一の物理サーバーが使用されてきました。仮想化ソフトウェアは、通常、サーバーの主記憶域とは別のブート ディスクに常駐します。信頼性を高めるには、この起動ディスクの RAID 1 設定が必要です。この RAID 1 ブートディスク (Figure 3 の青いボックス) は、Figure 4 に示す EZ-USB™ FX3S RAID-on-Chip USB ドングルを使用して実装できます。

FX3S USB RAID 1 ディスクは、メインサーバーシステムの初期ブートソースです。プラットフォームコントローラハブ (PCH) への高速 USB 3.0 リンクにより、サーバーシステムは迅速に起動できます。一般的なサーバーシステムは、ボード管理コントローラ (BMC) への個別の制御リンクも使用して RAID 動作を管理します。

FX3S RAID-on-Chip ドングルは、BMC 接続用に独立したスレーブ マルチメディアカード (MMC) インターフェースを提供します。BMC 制御リンクプロトコルはベンダ固有であるため、このアプリケーションノートでは説明しません。ただし、FX3S ファームウェアでこの機能を簡単に追加およびカスタマイズできます。特定のプロトコルの実装の詳細については、cypress.com/go/support からテクニカルサポートケースを作成してください。

このアプリケーションノートのファームウェア例は、BMC 制御を使用しない汎用 RAID 1 ソリューションを実装します。

Figure 4 に示す EZ-USB™ FX3S RAID-on-Chip ドングルハードウェアは Pactron によって製造され、pactronstore.com/products/cypress-fx3s.html から入手可能です。

ハードウェア回路図とサンプルファームウェアのソースコードは、このアプリケーションノートの付録に含まれています。

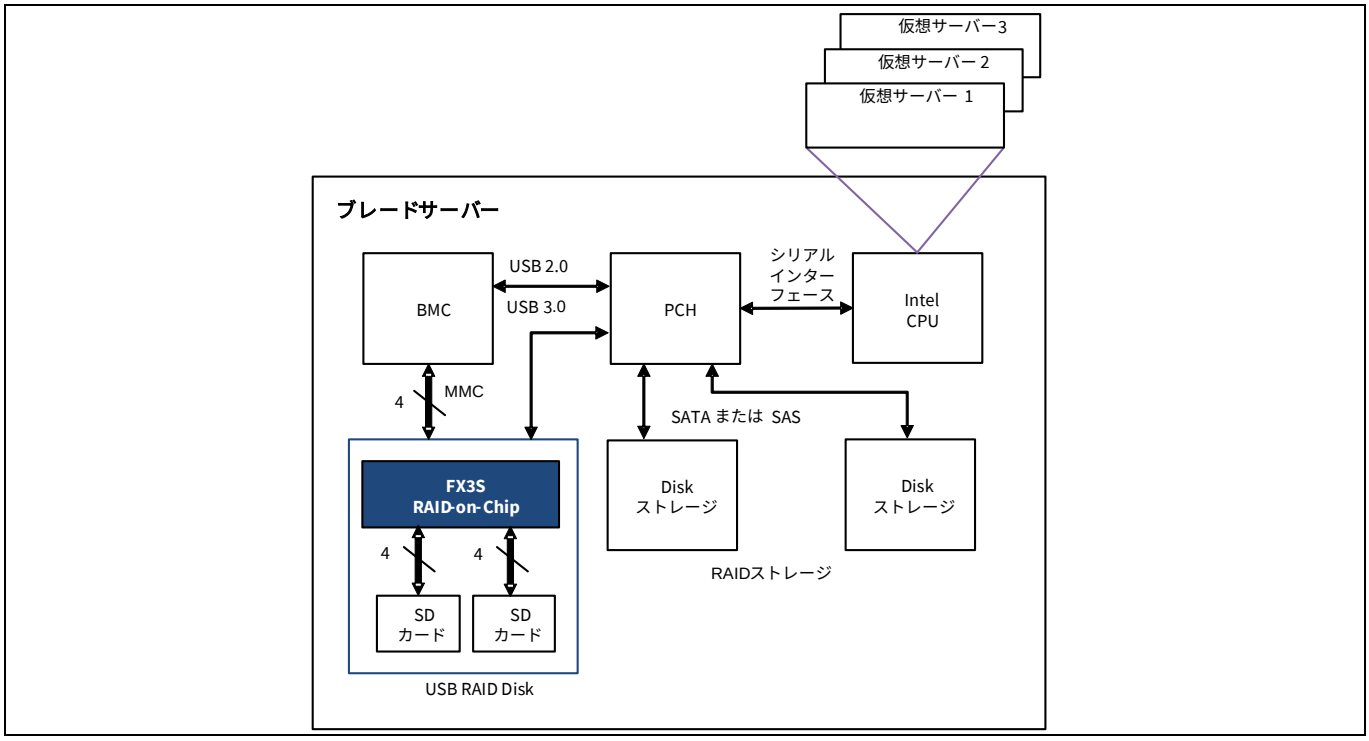


Figure 3 サーバーの概要

3 システム概要

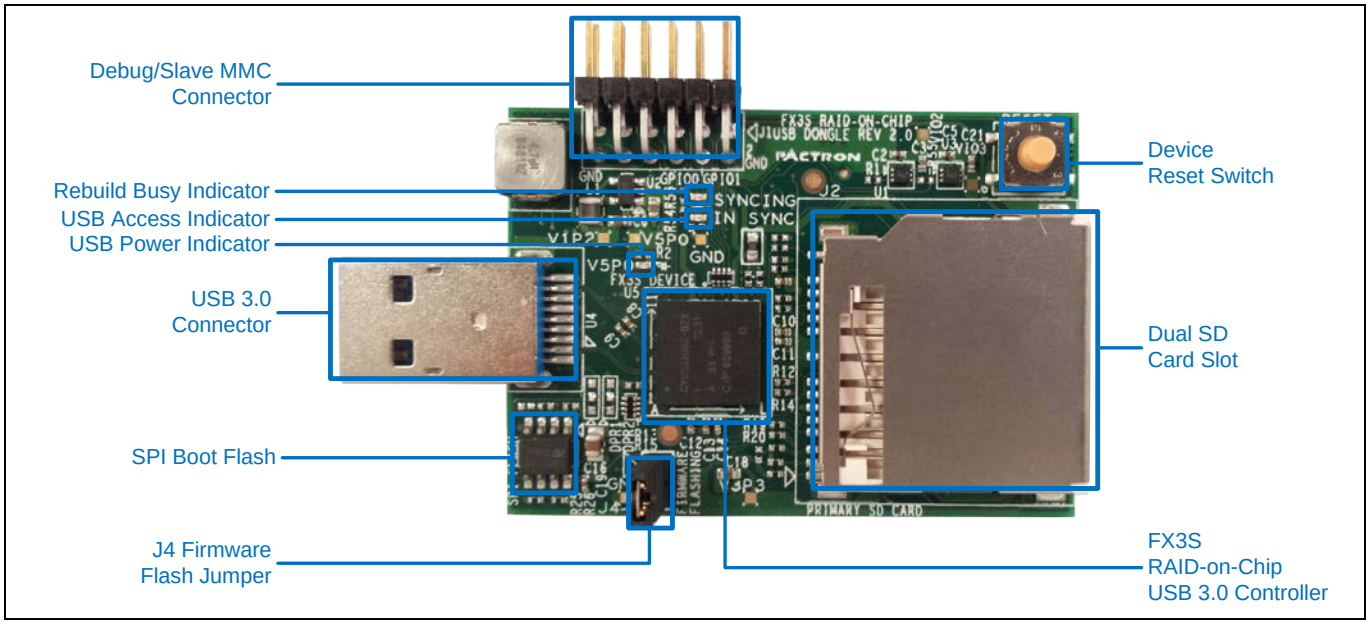


Figure 4 EZ-USB™ FX3S RAID-on-Chip USB ドングル

4 機能概要

FX3S USB RAID 1 ディスクは、標準の USB 3.0 大容量ストレージデバイスとして機能します。このアプリケーションノートで提供するファームウェアの例では、2 つの SD カードを使用して RAID 1 機能を実装しています。デバイスは、USB ホスト PC 上の 1 つのミラー化されたストレージボリュームとしてエニュメレートされます。すべての RAID 操作は FX3S 内で制御されます。ユーザーが外部 RAID を明示的に処理する必要はありません。

FX3S は、プライマリカードとセカンダリカードの 2 つの SD カードを認識します。読み出し操作の場合、USB ホストに返されるデータはプライマリカードから読み出されます。プライマリカードに障害が発生したり、イジェクトされたりした場合、セカンダリカードから読み出し操作が続行されます。書き込み操作では、USB ホストからのデータがプライマリカードとセカンダリカードの両方に同時に書き込まれます。一方のカードが故障または排出された場合、トランザクションは他のカードを中断することなく、このカードに対してのみ中止されます。どちらの場合も、FX3S は、2 つのカードのいずれかでエラー状態が発生するたびに、シームレスな USB ディスク操作を維持します。

RAID 1 ファームウェアの例では、プライマリカードまたはセカンダリカードが交換されたときの自動ボリューム再構築がサポートされています。ディスクの動作中に 1 つのカードが取り出されても、USB ホストは中断しません。新しいカードが挿入されると、RAID 1 ファームウェアは、アクティブカードの内容全体を新しく挿入されたカードにコピーして、再構築プロセスをトリガーします。

再作成プロセス中は、Figure 4 に示すように、LED 再作成ビジーインジケーターが点灯したままになります。USB ホストから SD カードへのストレージアクセスは、再構築プロセスが完了するまでブロックされます。USB ホストは、一時的な読み出し/書き込み要求の拒否を確認し、再構築プロセスが完了するまで試行を続けます。ホストオペレーティングシステムはこれを自動的に行うため、ユーザーの介入は必要ありません。

要約すると、FX3S USB RAID 1 ディスクは次のことを行います。

- USB SuperSpeed, High Speed, および Full Speed の標準大容量ストレージデバイスとして動作します。
- SD 3.0 (UHS-I) カードを 2 つ使用
- 2 つの SD カードのサイズが異なる場合は、小さい方の容量を報告して使用します。
- 通常の大容量記憶装置操作中にデータ ミラーリングを実行します。
- 一方の SD カードを取り外すと、もう一方の SD カードに中断なくアクセスできます。
- 1 つのカードを交換すると自動的にデータ同期を実行
- SD カードのホットプラグイベント (取り外しと挿入) を処理

5 ハードウェア

Figure 5 に示す USB RAID-on-Chip ドングルのハードウェア設計では、FX3S を使用して RAID 管理が組み込まれた SuperSpeed USB 大容量ストレージデバイスを実装しています。基板には、次のコンポーネントがあります。

- SuperSpeed, High Speed, Full Speed をサポートする USB 3.0 ペリフェラル インターフェース
- SD/SDIO 3.0 (UHS-I) および組込み MMC 4.41 デバイスをサポートする 2 つの独立したストレージポート
- FX3S ファームウェア ストレージ用の 4Mb SPI フラッシュ
- 19.2 MHz の水晶振動子入力をシステム クロック源として実現
- デバッグ用の UART と MMC 4.2 インターフェースを含む 12 ピンヘッダ (このアプリケーションノートで紹介する RAID 1 の例では使用しません)
- 電力, USB アクセス, および再構築操作用の LED インジケーター
- ハードウェアシステムリセットボタン

RAID-on-Chip ドングルは、USB インターフェースを介してバスパワーで駆動されます。FX3S と SD カードの電力レール (電源電圧) はすべて、USB ホストから供給される 5V VBUS から供給されます。

FX3S は、200 MHz の ARM9 MCU, 512 KB のシステムメモリ, およびこのアプリケーションに適したペリフェラルのフルアレイを備えています。基板は、オンボード SPI フラッシュデバイスからファームウェアをブートロードするように設定されていますが、システム開発用に USB 経由でブートロードするように基板を設定することもできます。

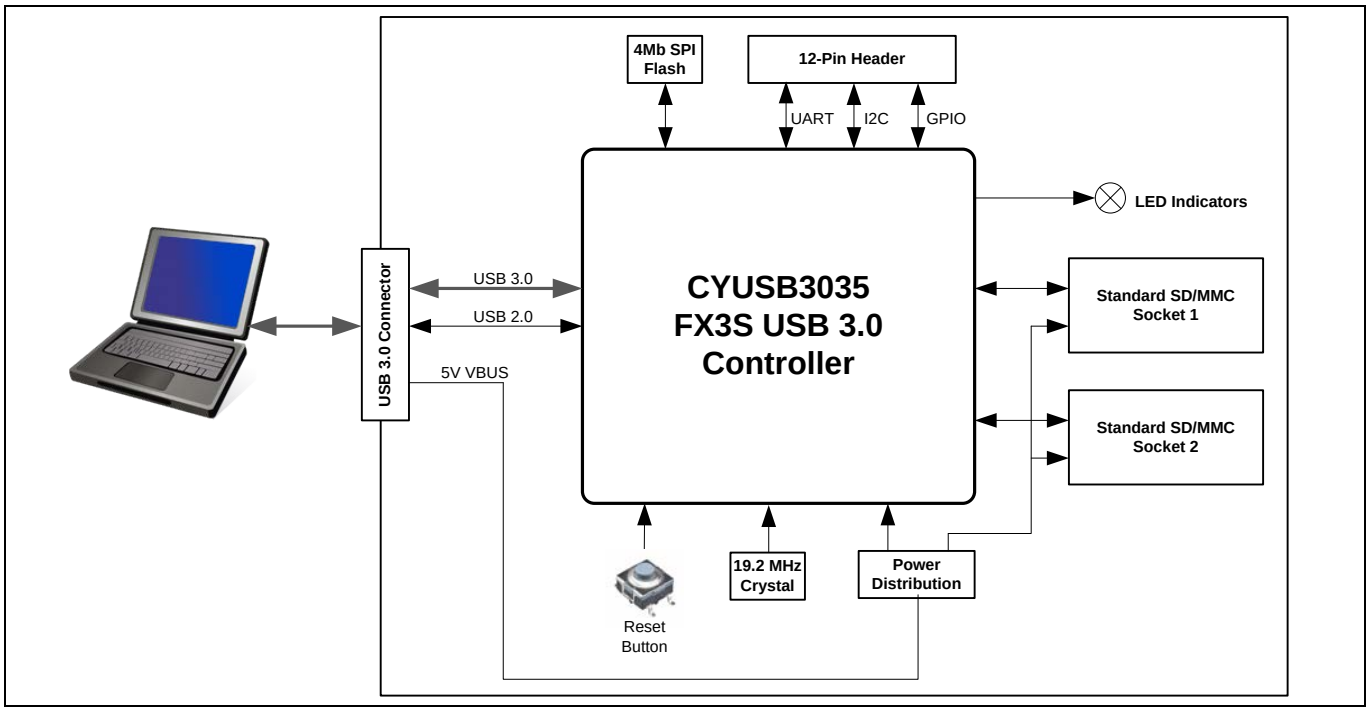


Figure 5 ハードウェアの概要

6 ファームウェア

このアプリケーションノートに付属するファームウェアサンプルプロジェクトは、ソース zip ファイルの FX3MSC_RAID1 フォルダにあります。このサンプルプロジェクトは、変更せずにバイナリ イメージをコンパイルして生成します。バイナリ イメージを FX3S にダウンロードするには、CYPRESS USB Control Center を使用します。ダウンロードする前に、FX3S がブートローダモードで起動できるように、基板の J4 が開いていることを確認してください。

Cypress USB Control Center を使用して FX3S ファームウェアプロジェクトを操作し、イメージをダウンロードする方法の詳細については、AN75705 - EZ-USB™ FX3 入門を参照してください。

6.1 ファームウェアソース

RAID 1 ファームウェアアプリケーションは、FX3S の内部リソースを効率的に管理するリアルタイムオペレーティングシステム (RTOS) 上で実行されます。FX3S ファームウェアアプリケーションは、低レベルのデバイスの詳細を抽象化する一連のアプリケーションプログラミングインターフェース (API) ライブラリ関数を介して FX3S ハードウェアペリフェラルと通信し、開発作業を大幅に簡素化します。

Figure 6 に、サンプルの RAID 1 ファームウェアプロジェクトに含まれるファイルとディレクトリを示します。

- Include: C ライブラリヘッダファイルとプロジェクト固有のヘッダファイルのコレクションを含むディレクトリ。
- api: FX3S API ライブラリのバイナリを含むディレクトリ。
- Debug と Release: プロジェクトは、デバッグ設定またはリリース設定のいずれかでコンパイルできます。Debug ディレクトリと Release ディレクトリには、それぞれのコンパイル設定の出力が含まれます。
- cyfx_gcc_startup.S: FX3S デバイス上の ARM9 コアのスタートアップコード。このアセンブリーソースファイルは、GNU アセンブラーの構文に従います。GNU は、自己参照的には、GNU’s not UNIX の略で、フリーソフトウェア財団 (FSF) によって開発された UNIX 互換のソフトウェアシステムです。
- cyfx3s_raid1.c: RAID 1 USB 大容量記憶装置の例の主な実装。
- cyfx3s_raid1.h: RAID 1 大容量記憶装置アプリケーションの定数定義と関数宣言。
- cyfxtx.c: FX3S API ライブラリに必要な ThreadX RTOS ラッパーとユーティリティ関数。
- cyraid1_descr.c: USB ディスクリプタの定義。
- makefile: コードのコンパイルを管理および整理するシェルコマンドファイル。
- readme.txt: FX3MSC_RAID1 プロジェクトの簡単な説明。

6 ファームウェア

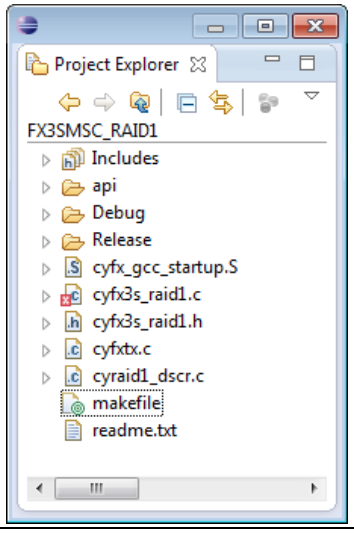


Figure 6　RAID 1 ファームウェアソース

すべての RAID 機能は、cyfx3s_raid1.c ファイル内に実装されています。Table 1 は、これらの機能をまとめたものです。

Table 1　cyfx3s_raid1.c の関数 RAID 1 アプリケーションの実装

関数カテゴリ	参照セクション	関数名	説明
RTOS	6.2	main()	この関数は、Arm® コアと C ライブラリの初期化後のファームウェア C コードエントリポイントです。キャッシュを設定し、FX3 I/O を設定し、RTOS カーネルを起動します。
	6.2	CyFxApplicationDefine()	この関数は、RTOS によって実行されるアプリケーションスレッドを定義します。
	6.2, 6.3, 6.4	MscAppThread_Entry()	この機能は、次のことを行う RAID 1 アプリケーションスレッドです: アプリケーション レベルの初期化を呼び出す 大容量記憶装置の内部状態遷移を管理する
大容量記憶装置アプリケーション レベルの初期化	6.3	CyFxMscAppInInit()	この関数は、FX3S デバイスレベルの初期化関数を呼び出して、次のタスクを実行します: 1. GPIO を初期化する。 2. 2 つの SD ストレージポートを初期化する。 3. USB ポートを初期化する。 4. イベントコールバックを登録する。 5. USB ディスクリプタを登録する。 6. 必要な DMA チャネルを初期化する。 7. ホストへの USB 接続を有効にする。

関数カテゴリ	参照セクション	関数名	説明
	6.3	CyFxMscApplnDebugInit()	この関数は、デバッグメッセージを出力するために FX3 UART ブロックを初期化します。
	6.3	CyFxMscApplnDmaInit()	この関数は、バッファメモリを割り当て、USB とストレージポート間の必要な DMA チャネルを初期化します。通常の状態では、2 枚の SD カードが接続されると、2 つの DMA 設定が初期化されます。 読み出し動作の場合: USB BULK-IN エンドポイントとプライマリ SD カードを接続する 1 対 1 のシングルチャネル DMA 書き込み動作の場合: USB BULK-OUT エンドポイントを両方の SD カードに接続する 1 対 2 のマルチキャストチャネル DMA
	6.3	CyFxMscApplnDmaReInit()	2 枚の SD カードのうち 1 枚がオフラインになり、再びオンラインに戻った場合、1 対 2 のマルチキャストチャネル DMA を再設定する機能です。
FX3S デバイスレベルの初期化	6.3	CyFxRdDmaInit()	読み出し動作の場合、この関数は USB BULK-IN エンドポイントとプライマリ SD カードを接続する 1 対 1 のシングルチャネル DMA を作成します。
	6.3	CyFxWrErrDmaInit()	2 枚の SD カードのうち 1 枚がオフラインになった場合、マルチキャストチャネル DMA を呼び出し、シングルチャネル DMA に再設定する機能です。
	6.3	CyFxMscApplnSibInit()	この機能は、2 つの SD ストレージコントローラを初期化します。
	6.3	CyFxMscApplnGpioInit()	アプリケーションに必要な GPIO を初期化する関数です。
大容量記憶装置アプリケーションのコールバック	6.2	CyFxMscApplnUSBEventCB()	このコールバック関数は、一般的な USB イベント (リセット, 接続, 切断など) を処理します。
	6.2	CyFxMscApplnUSBSetupCB()	このコールバック関数は、エニュメレーションとコントロールの転送中に USB セットアップイベントを処理します。
	6.2	CyFxMscApplnSibCB()	このコールバック関数は、ストレージポート イベント (カードの挿入と取り外し、データ転送の完了など) を処理します。
	6.2	CyFxMscApplnDmaCb()	このコールバック関数は、1 対 1 の単一チャネル DMA イベントを処理します。
	6.2	CyFxMscApplnMultiDmaCb()	このコールバック関数は、1 対 2 のマルチキャストチャネル DMA イベントを処理します。

関数カテゴリ	参照セクション	関数名	説明
大容量記憶装置アプリケーションの操作	参照されていません	CyFxMscApplnResetCtlr()	この機能は、USB データパスのクリアをリセットします。
	6.3	CyFxMscApplnParseCbw()	この関数は、受信したストレージコマンドを解析し、それに応じて処理します。
	参照されていません	CyFxMscApplnSendDataToHost()	この関数は、ホストが取得する BULK-OUT エンドポイントにデータを準備します。
	参照されていません	CyFxMscApplnSendCsw()	この関数は、ストレージコマンドの実行状態を、ホストが取得する BULK-OUT エンドポイントに準備します。
	6.4	CyFxMscApplnQueryDevStatus()	この機能は、2 つの SD ストレージポートをプローブし、そのステータスを報告します。
	参照されていません	CyFxAppLedOn()	この機能は、LED をオンにするように GPIO を設定します。
		CyFxAppLedOff()	この機能は、LED をオフにするように GPIO を設定します。

6.2 コールバック操作

RTOS がシステムリソースを管理するため、RAID 1 ファームウェアの実行はイベント駆動型です。ペリフェラル (USB および SD ポート) と内部 DMA システムはイベントを生成します。これらのイベントは、ファームウェアの初期化中に登録されたコールバック関数によって処理されます。

RAID 1 ファームウェアの実行が開始されると、ARM9 コア, GNU ツールチェーンライブラリ, および RTOS の一連の初期化シーケンスが実行されてから、*cyfx3s_raid1.c* の `main()` 関数 に入ります。RTOS は、`main()` から `CyU3PKernelEntry()` を呼び出すことから始まります。RTOS がスレッドスケジューリングを開始する前に、アプリケーション タスクを実行するために少なくとも 1 つのスレッドが作成されます。RAID 1 サンプルプロジェクトの場合、アプリケーションスレッドは `MscAppThread_Entry()`で、まずペリフェラルインターフェースを設定し、USB, ストレージ, および DMA イベントを処理するイベントコールバック関数を登録します。これらのコールバック関数には、次のものが含まれます。

- `CyFxMscApplnUSBEventCB`: このコールバック関数は、次の USB イベントを処理します。
  - `CY_U3P_USB_EVENT_SUSPEND`: USB サスペンドイベント。デバイスは低 POWER_DRILL2GO モードになります。
  - `CY_U3P_USB_EVENT_DISCONNECT`: USB 切断イベント。デバイスがホストから切断されます。
  - `CY_U3P_USB_EVENT_RESET`: USB リセット イベント。ホストからリセットを受信します。
  - `CY_U3P_USB_EVENT_CONNECT`: USB MULTICH_CONNECT_PCB イベント。デバイスがホストに接続されています。
  - `CY_U3P_USB_EVENT_SETCONF`: ホストは、エニュメレーションプロセスを完了するようにデバイスを"設定"します。
- `CyFxMscApplnUSBSetupCB`: このコールバック関数は、主にエニュメレーション中に発生する次の USB セットアップイベントを処理します。
  - `CY_FX_MSC_USB_STANDARD_REQ`: ホストから標準セットアップ要求を受信します。

  - CY_FX_MSC_USB_CLASS_REQ: クラス固有の要求がホストから受信されます。
  - CY_FX_MSC_USB_VENDOR_REQ: ベンダ固有の要求がホストから受信されます。これは、カスタムベンダホストドライバが使用される場合のプレースホルダーです。
- CyFxMscApplnSibCB: このコールバック関数は、次のストレージポートイベントを処理します。
  - CY_U3P_SIB_EVENT_XFER_CPLT: DMA 転送が完了しました。
  - CY_U3P_SIB_EVENT_INSERT: 2 つの SD ソケットのうちの 1 つからカードの挿入が検出されました。
  - CY_U3P_SIB_EVENT_REMOVE: 2 つの SD ソケットのうちの 1 つからカードの取り外しが検出されました。
  - CY_U3P_SIB_EVENT_DATA_ERROR: 2 枚のカードのうちの 1 枚でエラーが発生しました。
  - CY_U3P_SIB_EVENT_ABORT: 読み出し/書き込み操作が中止されました。
- CyFxMscApplnDmaCb: このコールバック関数は、1 対 1 の DMA チャネルのイベントを処理します。1 対 1 のチャネルには、1 つのデータプロバイダーと 1 つのデータコンシューマーがあります。これらのイベントは通常、USB 経由で大容量記憶装置コマンドブロックラッパー (CBW) を受信したり、USB 経由で大容量記憶装置コマンドステータスラッパー (CSW) を送信したり、1 つのストレージポートから読み出したりするときに発生します。
  - CY_U3P_DMA_CB_SEND_CPLT: DMA 送信データ転送が完了しました。
  - CY_U3P_DMA_CB_RECV_CPLT: DMA 受信データ転送が完了しました。
- CyFxMscApplnMultiDmaCb: このコールバック関数は、1 対多の DMA チャネルの次のイベントを処理します。1 対多の転送では、1 つのデータプロバイダーと複数のデータコンシューマー (この場合は 2 つの SD カード) があります。これらのイベントは通常、USB からのデータを両方のストレージカードに書き込む必要があるときに、ストレージ書き込みコマンド中に発生します。
  - CY_U3P_DMA_CB_RECV_CPLT: 送信データを含む DMA 転送が完了しました。
  - CY_U3P_DMA_CB_PROD_EVENT: 受信データを含む DMA 転送が完了しました。

6.3 大容量記憶装置操作

RAID 1 ファームウェアがデバイスレベルおよびアプリケーションレベルの初期化シーケンスを完了すると、実行は MscAppThread_Entry() で続行され、デバイスによって発生した USB 大容量ストレージアプリケーションイベントを待機する無限ループにとどまります。

RAID 1 ファームウェアアプリケーションは、バルクオンリートランスポート (BOT) を使用して標準の USB マス ストレージ クラス (MSC) を実装します。BOT は、MSC 操作の 3 つの基本的な段階を定義しています。

- コマンド転送: コマンド転送では、ホスト PC は BULK-OUT 転送を使用してデバイスにコマンドを送信します。このコマンドパケットは CBW として定義され、BOT は常に CBW で始まる必要があります。
- データ転送: データ転送は、ホストとデバイス間でデータを転送するために使用されます。例えば、read/write コマンドでは、さまざまなストレージセクターの実際のデータが、複数のバストランザクションを利用するデータ転送を使用して送信されます。データ転送では、BULK-OUT または BULK-IN 転送を使用するデータ転送を使用して実行されます。
- ステータス トランスポート: ステータストランスポートは、BULK-IN 転送を使用して、コマンド実行の結果をデバイスからホストに送り返すために使用されます。CSW は、このステータスパケットを定義します。BOT は常に CSW で終わる必要があります。

Figure 7 に、MSC ステージの遷移の概要を示します。

6 ファームウェア

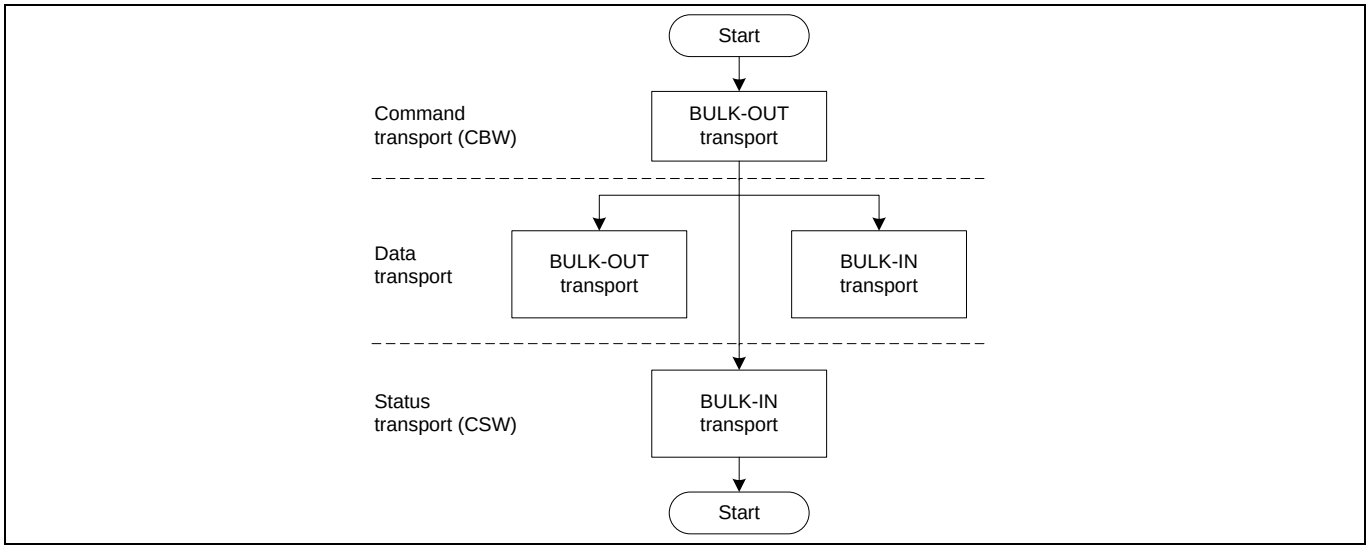


Figure 7　USB MSC 転送ステージ

標準 MSC プロトコルをサポートするために、以下のイベントが `MscAppThread_Entry()` 内で処理されます。

- `CY_FX_MSC_RESET_EVENT`: 大容量記憶装置がリセットされました。
- `CY_FX_MSC_SETCONF_EVENT`: エニュメレーションが完了しました。MSC 操作用にデバイスをセットアップする必要があります。
- `CY_FX_MSC_CBW_EVENT`:MSC CBW パッケージが届きました。コマンド (SCSI 形式を使用) は、このイベントが受信されたときに解析され、実行されます。
- `CY_FX_MSC_DATASENT_EVENT`: データ転送段階では、データがホストに送信され、ファームウェアはデバイスをステータス転送ステージに移行します。
- `CY_FX_MSC_SIBCB_EVENT`: このイベントでは、大容量記憶装置が次のデータトランザクション用にストレージ ポートを設定する必要があります。
  - デバイスがコマンド転送段階にある場合、ファームウェアはホストから CBW パッケージを受け入れるように受信バッファを設定します。
  - デバイスがデータ転送段階にある場合、ファームウェアは現在のコマンドのデータ方向に応じて送信バッファまたは受信バッファを設定します。
  - デバイスがステータス転送段階にある場合、ファームウェアは CSW パッケージを更新し、ホストに送信します。

RAID 1 ファームウェアは、内部ステートマシンを介して大容量ストレージデバイスの機能を管理します。状態は、Figure 7 に示す MSC 転送ステージごとに定義されます。次のサンプル C コードに、これらの状態がファームウェアでどのように定義されるかを示します。

```
typedef enum
{
    /* Inactive state, waiting for SET_CONFIG. */
    CY_FX_MSC_STATE_INACTIVE = 0

    /* Waiting to queue a CBW command. */
    CY_FX_MSC_STATE_CBW,
```

```
    /* transitional state that it tells USB host it is ready to receive a CBW
command. */
    CY_FX_MSC_STATE_WAITING,

    /* Waiting to complete data transfer for a command. */
    CY_FX_MSC_STATE_DATA,

    /* Waiting to send CSW for a command. */
    CY_FX_MSC_STATE_STATUS,

    /* Waiting for host to read out CSW packet. */
    CY_FX_MSC_STATE_CSW
} CyFxMscFuncState;
```

FX3S USB RAID ディスクがエニュメレートされると、USB ホストは CBW パッケージを BULK-OUT エンドポイントに送信します。FX3S は、CBW データをバッファ glMscCbwBuffer に保存します。デバイスがコマンドを実行する準備が整うと、現在のデバイスの状態が CY_FX_MSC_STATE_CBW に変わり、MSC CBW イベント CY_FX_MSC_CBW_EVENT_FLAG が発生します。デバイスは、USB エニュメレーションが成功した後、または最後のコマンドが正常に完了した後に、CY_FX_MSC_STATE_CBW 状態に設定されます。

MscAppThread_Entry()は、CY_FX_MSC_CBW_EVENT_FLAG イベントを検出した後、CyFxMscApplnParseCbw()を呼び出して、glMscCbwBuffer に格納されているコマンドを解析します。コマンドの種類に応じて、デバイスの状態を CY_FX_MSC_STATE_DATA または CY_FX_MSC_STATE_STATUS に移行します。

コマンドが CY_FX_MSC_SCSI_READ_10 または CY_FX_MSC_SCSI_WRITE_10 の場合は、ストレージコールバックイベント CY_FX_MSC_SIBCB_EVENT_FLAG が発生します。MscAppThread_Entry() 関数は、SD カードと USB エンドポイントの間に適切な DMA チャネルを設定することで、このイベントを処理します。USB ホストが転送を開始すると、データはデータ数に達するまで DMA チャネルを介して SD カードとの間で流れます。DMA 転送が完了すると、2 つの DMA コールバック関数のうちの 1 つが転送完了イベントを受信し、MSC データ送信イベント CY_FX_MSC_DATASENT_EVENT_FLAG が発生します。MscAppThread_Entry() 関数が MSC データ送信イベントを処理すると、コマンド状況が更新されます。デバイスの状態も CY_FX_MSC_STATE_STATUS に移行されます。

デバイスが CY_FX_MSC_STATE_STATUS 状態になると、MSC CSW パケットが構築され、USB ホストに送り返されます。CSW が送信されると、デバイスの状態は CY_FX_MSC_STATE_CBW に戻り、デバイスは次のコマンドを実行する準備が整います。

6.4 自動再構築

RAID 1 ファームウェアは、ストレージの冗長性を復元するための自動ボリューム再構築をサポートしています。MscAppThread_Entry() は、2 つのステータスフラグ (古いステータスとカードプレゼンス) をストレージごとに 1 セットずつ監視することで、ボリュームの再構築を管理します。

カードが故障したり、ソケットから取り外されたりするたびに古いステータスフラグが設定されます。これは、両方のカードが動作し、同期されている場合、通常は再構築操作が成功した後にクリアされます。古いステータスは、2 つの SD カードを挿入した状態での初期の電源投入後にもデフォルトで設定されます。ファームウェアは、常に 2 つの SD カードが初期の電源投入状態で同期されていることを前提

としています。最初の電源投入後にディスクをフォーマットして、開始時に両方のカードが同期されるようにすることを推奨します。

カードプレゼンスフラグは、カードの挿入 (CY_U3P_SIB_EVENT_INSERT) または取り外し (CY_U3P_SIB_EVENT_REMOVE) イベントが発生すると更新されます。これらのイベントは、ストレージコールバック関数 CyFxMscApplnSibCB() 内で処理されます。カードを取り外すと、カードプレゼンスフラグがクリアされます。カードが挿入されると、CY_U3P_SIB_EVENT_INSERT イベントが発生し、CyFxMscApplnSibCB()は CyFxMscApplnQueryDevStatus() を呼び出してカードを検出して初期化します。カードが正常に初期化されると、カードプレゼンスフラグが設定されます。

MscAppThread_Entry()は、カードプレゼンスフラグと古いステータスフラグの両方が設定されていることを検出すると、CyWbRaidFullCardSync()を呼び出してボリュームの再構築プロセスを開始します。ボリュームの再構築プロセスは、古いステータスがクリアされるまで実行を続けます。古いステータスは、SD カードの全内容が新しく挿入されたカードに正常に転送された後、冗長性が復元されるとクリアされます。再構築プロセス中に、プライマリ SD カードが再構築プロセスを処理するためにビジー状態になるため、USB ホストのディスクアクセスが使用できなくなることがあります。この場合、ホストは、再構築操作が完了し、ディスクが再びアクセス可能になるまで、ディスクアクセスを再試行し続けます。

7 操作手順

次の操作手順を使用して、FX3S USB RAID ディスクのサポートされている機能をテストできます。

1. FX3S USB RAID ディスクを任意の USB スロット (SuperSpeed, High Speed, または Full Speed) に接続します。LED POWER_DRILL2GO インジケーターが点灯します。SD カードへの継続的な USB アクセスも、LED USB アクセスインジケーターが点滅します。Figure 4 に、これらの LED インジケーターの位置を示します。
2. USB エニュメレーション後、Windows エクスプローラーの自動再生ウィンドウにディスク ドライブが表示されます (Figure 8 参照)。

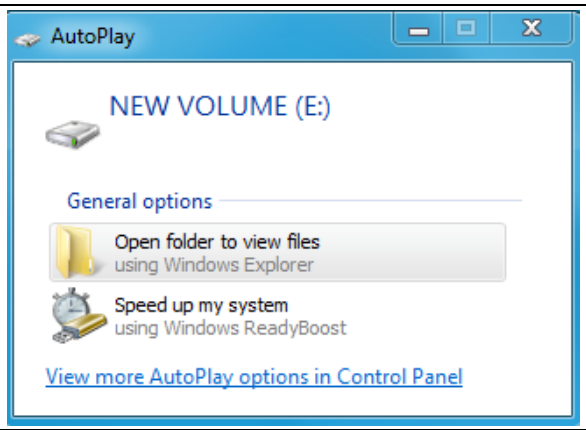


Figure 8 Windows エクスプローラーの自動再生ウィンドウ

3. FAT または FAT32 ファイルシステムを使用してディスクをフォーマットします。1 つのディスクボリュームとして表示され、ミラーリングされた 2 つの SD カードを使用しているという事実が隠されていることに注意してください。
4. ビデオファイルをカードに移動し、ビデオを再生します。
5. ビデオの再生中に、SD カードの 1 つを取り外します。ビデオが中断することなく再生され続けることを確認します。これは、RAID 設計の安全機能を実証しており、1 枚のカードが故障したり取り外されたりしても、システムは 1 枚のカードで動作し続けます。
6. 手順 4 で取り外した SD カードの内容を削除し、カードを挿入し直します。これにより、再構築操作がトリガーされます。Figure 4 に示す LED 再作成ビジーインジケーターが点灯し、PC がカードにアクセスできなくなります。内部バッファリングにより、ビデオの再生が少し長くなる場合がありますが、最終的には停止します。
7. 再構築操作が完了すると、LED 再構築ビジーインジケーターが消灯し、ビデオの再生が再開されます。これは、再構築操作にユーザーの介入が必要ないことを示します。
8. 手順 4 と同じ SD カードを取り出し、内容を確認します。これは、FX3S RAID-on-Chip USB ドングル KIT_XC2210U_SK に接続されたままの SD カードのコピーである必要があります。

8 パフォーマンス

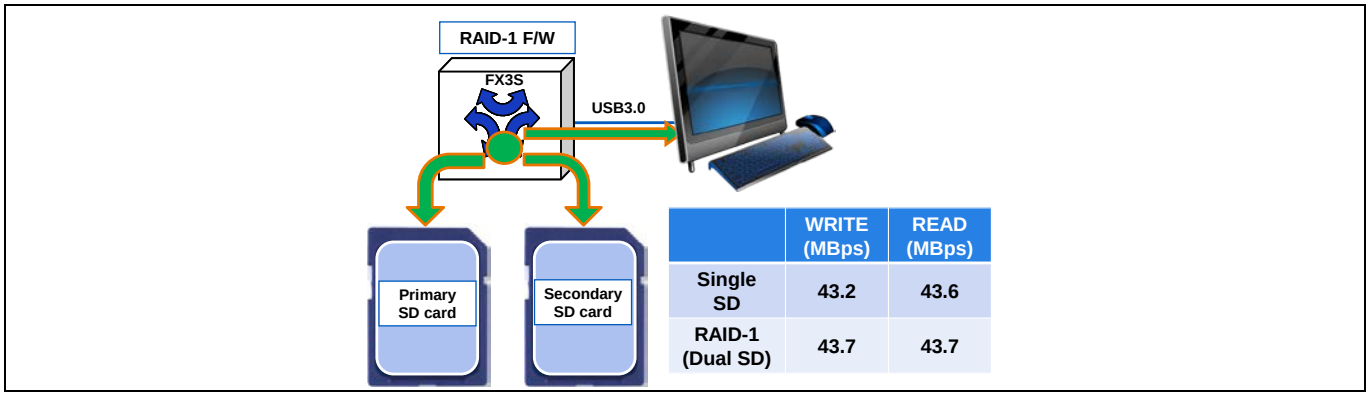
RAID 1 ファームウェアを EZ-USB™ FX3S RAID-on-Chip USB ブートディスク上で実行した場合、ここで示すパフォーマンスは、次の条件下で測定されました。

- 測定ツール: CrystalDiskMark v3.0.2 x64
- SD カードタイプ: SanDisk Extreme Pro 8 GB
- USB ホスト: Windows 7 x64 を実行する Intel Ivy 統合 USB 3.0 ホストを搭載した Lenovo ThinkPad T430

一般に、最適なパフォーマンスを実現するには、同一の SD カードを使用することを推奨します。2 つのカードのサイズと速度が異なる場合は、小さい方のサイズが論理 RAID 1 ディスクボリュームに使用されます。また、カードが遅いとパフォーマンスも制限されます。

8.1 USB⇔SD カード

USB から SD カードへの転送パフォーマンスは、一般的なディスクベンチマークツールである CrystalDiskMark を使用して、USB 3.0 ホストからドライブを読み書きすることによって測定されます。このツールは、crystalmark.info/software/CrystalDiskMark/index-e.html からダウンロードできます。

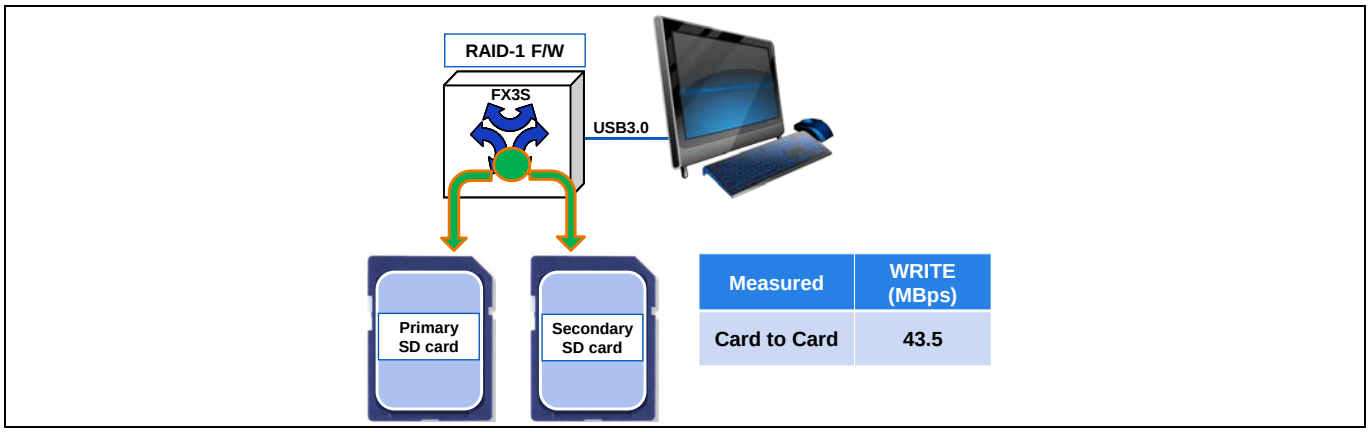


	WRITE (MBps)	READ (MBps)
Single SD	43.2	43.6
RAID-1 (Dual SD)	43.7	43.7

Figure 9 USB 3.0⇔SD パフォーマンス

8.2 SD カード⇔SD カード

SD カードから SD カードへの転送は、SD カードの 1 つを取り外してから再挿入することで開始されます。これにより、再構築操作がトリガーされ、アクティブなカードの内容全体が新しく挿入されたカードに転送されます。



Measured	WRITE (MBps)
Card to Card	43.5

Figure 10 SD⇔SD パフォーマンス

9 まとめ

このアプリケーションノートでは、サーバーブートストレージアプリケーション用の EZ-USB™ FX3S を使用した RAID-on-Chip USB 大容量ストレージ デバイスのデモを行いました。本書には、基板設計図とサンプルファームウェアのソースコードが関連付けられています。

10 付録 A: FX3S RAID オンチップ USB ブート ディスク基板レイアウト ビュー

この画像をダブルクリックすると、基板レイアウトの上部ビューと下部ビューが開きます。

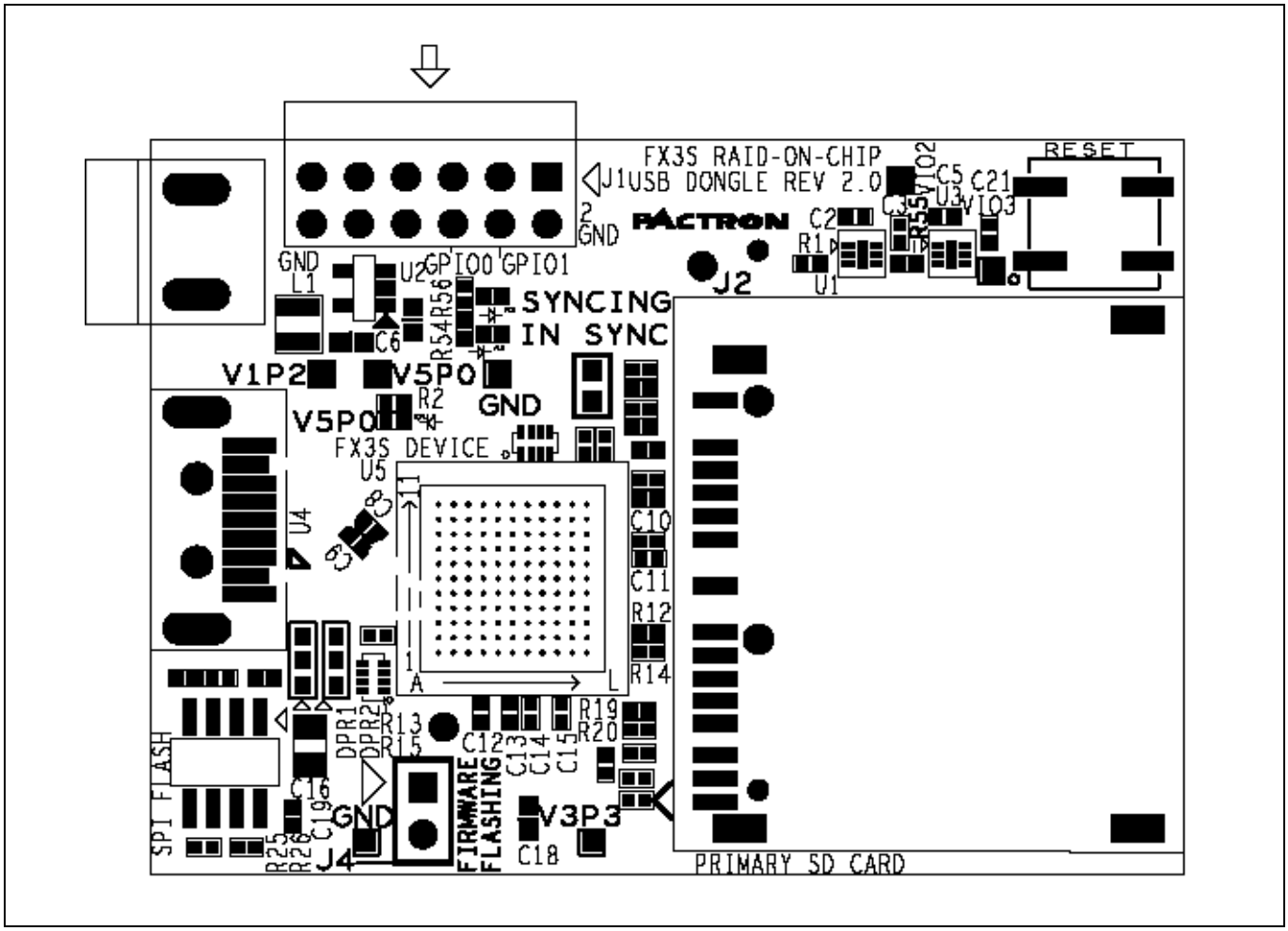


Figure 11

11 付録 B: FX3S RAID-on-Chip USB ブート ディスクの回路図

この画像をダブルクリックすると、基板回路図ファイルが開きます。

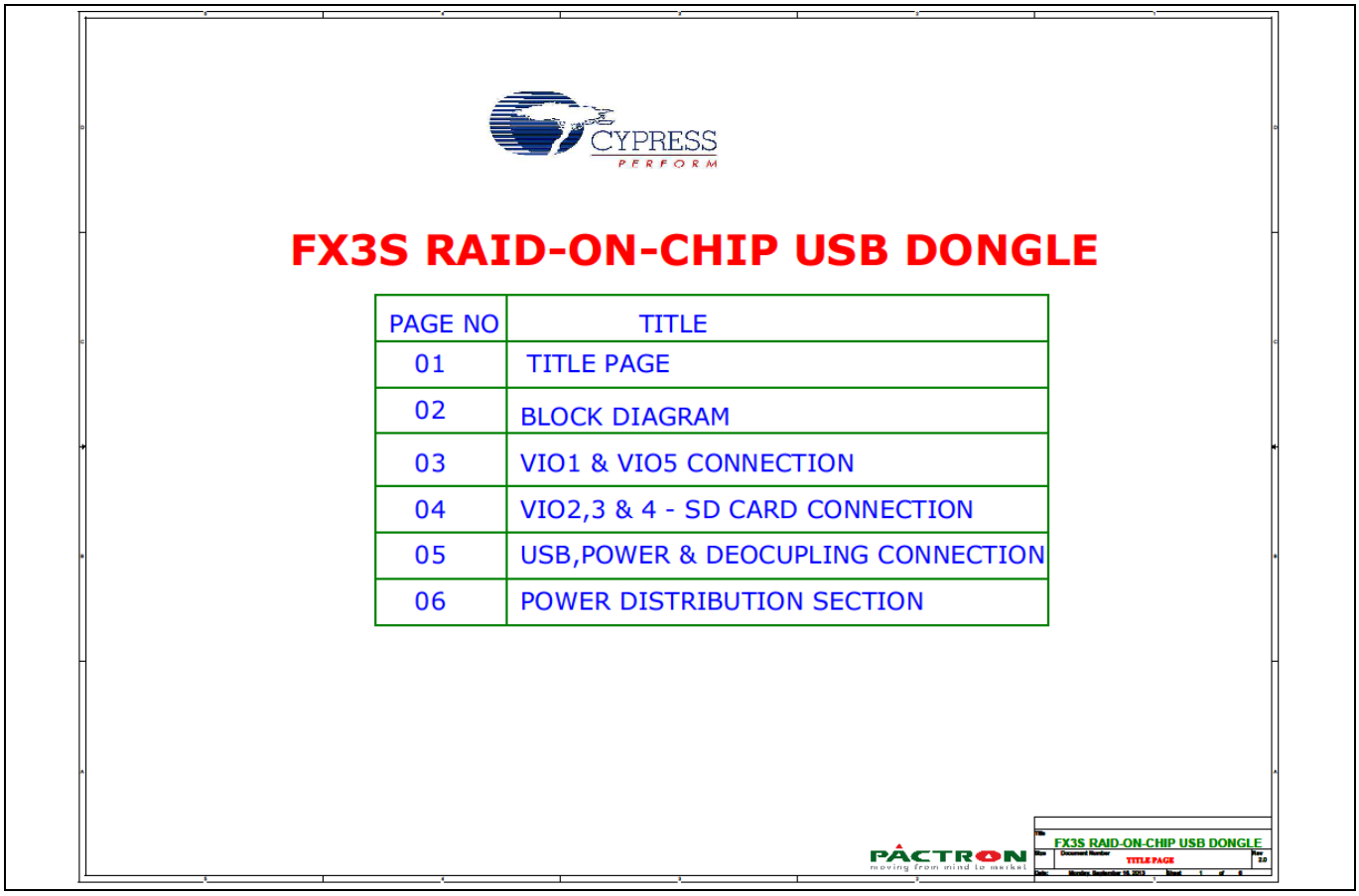


Figure 12

改訂履歴

版数	発行日	変更内容
**	2025-08-18	本版は英語版 001-89661 Rev. *B について、インフィニオン DEVELOPER COMMUNITY の参画者によって日本語に翻訳されたドキュメントです。

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-08-18
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2025 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
002-41937 Rev. **

重要事項
本手引書に記載された本製品の使用に関する手引きとして提供されるものであり、いかなる場合も、本製品における特定の機能性能や品質について保証するものではありません。本製品の使用の前に、当該手引書の受領者は実際の使用環境の下であらゆる本製品の機能及びその他本手引書に記された一切の技術的情報について確認する義務が有ります。インフィニオンテクノロジーズはここに当該手引書内で記される情報につき、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

警告事項
技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。
インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。