デバイス情報:

メーカー : Samsung
タイプ: K9F2G16Q0M [TSOP48]
8ビットバイト: 10800000h (276,824,064 Bytes)
構成: 8400000hx16 bit
アルゴリズム: Specialized
Implemented in SW ver.: 2.97 又は、前
Modified in SW ver.: 3.07
パッケージ情報: TSOP(48), 12x20mm

プログラマーとプログラミング・アダプタ/モジュールによりサポートされています:

- BeeHive208S アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) **(*1)** 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeHive204 アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) **(*1)** 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeHive204AP アダプタ/モジュールで:
  - AP1 TSOP48 ZIF 18.4mm NAND-3 (注文番号 71-3097) **(*1)** 又は
  - **AP1 TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeHive204AP-AU アダプタ/モジュールで:
  - AP1 TSOP48 ZIF 18.4mm NAND-3 (注文番号 71-3097) **(*1)** 又は
  - **AP1 TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeProg2 アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) **(*1)** 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeProg2C アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) **(*1)** 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeProg2AP アダプタ/モジュールで:
  - AP1 TSOP48 ZIF 18.4mm NAND-3 (注文番号 71-3097) **(*1)** 又は
  - **AP1 TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeHive8S **(ディスコン製品)** アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) 又は
  - DIL48/TSOP48 ZIF 18.4mm (注文番号 70-0065) 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeHive4+ **(ディスコン製品)** アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) 又は
  - DIL48/TSOP48 ZIF 18.4mm (注文番号 70-0065) 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeHive4 **(ディスコン製品)** アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) 又は
  - DIL48/TSOP48 ZIF 18.4mm (注文番号 70-0065) 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeProg+ **(ディスコン製品)** アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) 又は
  - DIL48/TSOP48 ZIF 18.4mm (注文番号 70-0065) 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- BeeProg **(ディスコン製品)** アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm NAND-3 (注文番号 70-3081) 又は
  - DIL48/TSOP48 ZIF 18.4mm (注文番号 70-0065) 又は
  - **DIL48/TSOP48 ZIF 18.4mm NAND (ディスコン)**
- JetProg **(DIL48) (ディスコン製品)** アダプタ/モジュールで:
  - DIL48/TSOP48 ZIF 18.4mm (注文番号 70-0065)

ノート:

(*1): このアダプターはプログラミング中に**TurboMode**が使用出来ます、StandardModeより2.5 倍高速です。さらなる情報のために**Help** or our **web site**をチェックして下さい

一般情報:

デバイスは2048ブロックあります。各ブロックは1024ワードのデータ領域と
32ワードの(コントロール)領域で構成された64ページを含んでいます

デバイスは最初に出荷された時には幾つかの無効ブロック(IB)を含んでいます。
追加の無効ブロックは使用される間に開発されます。従ってあるレベルでのデータ管理が この種のデバイスには不可欠です

Descriptions of ACCESS METHOD <Alt+S>メニュー:

INVALID BLOCK MANAGEMENT[無効ブロック管理]オプション:

IB管理方法は無効ブロックをどの様に扱うかを言います。多くの管理方法が利用出来ます:

- **Treat All Blocks[全ブロックを扱う]:** 無効ブロックを有効として扱います。デバイス全体のリード,
ベリファイとプログラムが可能です。開発目的には便利ですが、しかし、プログラミング又は、イレースの間に
データとIB情報のロスを起こすことがあります。 その様な場合デバイスのベリファイに失敗します。

- **Skip IB:** もし、無効ブロックが見つかりますとスキップされます。
関連データは次の有効ブロックに書き込まれます。全デバイス操作は有効ブロック内のみで可能です。

- **Skip IB with Map in 0-th Block[0番ブロックのMapでのSkip IB:** 前述と同じですが、
IBマップは0-thブロックにプログラムされます(古いシステムと互換のため)

- **Skip IB with Excess Abandon[超過放棄によるSkip IB]:** Skip IBと同じですが、しかし、もし、User
Areaの後のバッファにあるデータが残っていますと-最後のブロックが
処理(無効ブロックが発生したため)され、それらは破棄され操作は成功と見なされます。
この方法は、例えばあるファイル・システムと/又は、オペレーティング・システム・ヘッダーを User
Area内のパッディング(使用されない)ブロックにプログラミングするのにこの方法は便利です。

- **RBA(Reserved Block Area[リザーブされたブロック領域]):** 無効ブロックは
ブロック・リザーバーから分っている有効なブロックに置き換えられます。リザーバーはユーザー・セッティングに依ってUser
AreaとRBAテーブルの間, 又はRBAテーブルと
デバイスの終了の間に置く事が出来ます。これらの置き換えは2つコピー(1つのオリジナルと
1つのバックアップ・コピー)の指定されたブロックに書き込まれるRBAテーブルにストアされます。

- **Check IB without Access[アクセス無のIBチェック]:** User
Areaの無効ブロックの数をチェックし、そして、もし、スレッシュルドを超えていればエラーを報告します。
他の操作はデバイス上では行われません。
ノート: 選択されたプログラマーのみでオプションが利用出来ます。

- **Check IB with Skip IB[スキップIBでIBをチェック]:** User
Areaの無効ブロックの数をチェックし、そして、スレッシュルドを超えていればエラーを報告します。
もし、OKの場合、選択された操作がデバイス上で実行されます。
ノート: 選択されたプログラマーのみでオプションが利用出来ます

- **Multiply Partition(with Skip IB)[マルチプライ・パーティション(スキップIBで)]:** 基本パーティショニング方法[区分編成法],
パーティション内部のSkip IBを使用します。
入力データ・ファイル(ファイル・ロードで自動的に追加することが出来ます)にスペア領域が含まれることが必要です
ノート: 選択されたプログラマーのみでオプションが利用出来ます。更に詳しい情報は ディーラーに問い合わせて下さい。
又は、アプリケーション・ノートのhttp://www.elnec.com/sw/an_elnec_nand_partitions.pdf をお読み下さい。.

- **Linux MTD互換:** Linux MTDドライバーをエミュレートする特別なパーティショニング方法。パーティショニングとは別にBad Block
TableとECCをビルドしプログラムすることが出来ます。
ノート: 選択されたプログラマーのみでオプションが利用出来ます。更に詳しい情報は ディーラーに問い合わせて下さい。
又は、更に詳しい情報はアプリケーション・ノートのhttp://www.elnec.com/sw/an_elnec_linux_mtd.pdf をご覧下さい。.

- **Redirection with HW LUT[HW LUTでのリダイレクション] (Winbond):**
ある種のWinbondのデバイス上で利用可能な特別な無効ブロック・リダイレクション技術を利用するハードウェア・
ルックアップ・テーブル。詳細はデバイスのデータシートをご覧ください
ノート: プログラマーはデバイスの最後のブロックからデバイスの最初に向かったリダイレクション のためにブロック選択します。

SPARE AREA USAGE[スペア領域使用]オプション:

スペア領域使用の選択は何をスペース領域に書くのかを選択出来ます。いくつかのオプションがあります:

- **Do not Use[使用しない]:** スペア領域は使用されません(IB情報はすべて保持されます)

- **User Data[ユーザ・データ]:** スペア領域にユーザー・データが書き込まれます。それはプログラミング中に**IB情報が失われる**
ことがあります。その場合にはデバイスの ベリファイには失敗します。デバイスの読み込みは偽のIB情報に非該当なメモリ・イメージを提供
することが出来ます。

- **User Data with IB Info Forced[強制IB情報でのユーザー・データ]:**
ユーザー・データはスペア領域に書き込まれます。IB情報を持ったバイト/ワードは無視され 替わりに本当のIB情報が強制されます。

- **ECC-ハミング(サムスン方式):** ダブル・エラーがシングル・エラー修正を見つけますと
データ領域の信頼性を確保するためハミング・アルゴリズムが採用されます。
アルゴリズム・パラメータはサムスン社により指定されています-
http://www.samsung.com/global/business/semiconductor/products/flash/downloads/applicationnote/ecc_algorithm_for_web_512b.pdf
をご覧下さい。(もしこのリンクが駄目な場合はサムスン社のウェブサイト http://www.samsung.com でフラッシュ・アプリケーション・ノートの
コーナーでご覧下さい。 又は、http://www.elnec.com/sw/samsung_ecc_algorithm_for_512b.pdfからダウンロードして下さい。.)

- **ECC-ハミング(2x 256バイト・フレーム)バリアント1:** ハミングECCアルゴリズム はSmartMedia
cardのために指定されているのと同じ計算カーネルを使用しています。 SmartMedia
cardのために指定されたと同じコンピュテーショナル・カーネルを使っています。
これはサイズが256バイトのフレーム・ページを分割し3バイトのコントロール・データを
作成し、フレーム当たり1シングル・ビット・エラーをカバーします。 スペア領域レイアウトは一般的にLinux
MTDドライバーで使用されるそれと互換性があります。またコントロール・データ・バイトの順番はLinuxの順番に準じています。
ノート: 選択されたプログラマーのみでオプションが利用出来ます。更に詳しくはディーラー に連絡して下さい。

- **ECC-ハミング(2x 256バイト・フレーム)バリアント2:** このECCアルゴリズムは前のバリアント1と同じですが、SmartMedia cardの
オリジナル・コントロール・データ・バイトの順番を使用しています。バイトECC[0]とECC[1] はそれぞれの位置で切り替えられます。
ノート: 選択されたプログラマーのみでオプションが利用出来ます。更に詳しくはディーラー に連絡して下さい。

ENABLE DEVICE INTERNAL ECC CONTROLLER[デバイス内部ECCコントローラ有効]チェック-ボックス:

一部のNANDフラッシュ・デバイスの内部コントローラはホスト・コントローラの負荷を
大幅に軽減することができるECCユニットを含んでいます。有効にしますと、この内部ECCユニット
はECCチェックサムを計算し、それらをプログラミング時にスペア領域にストアします。
その後、読み出し時には、再度、サムをデータ領域から計算し、それらをスペア領域にストア

されたサムと比較します。もし、エラーが見つかった場合はECCアルゴリズム機能に従って データは自動的に再ストアされます。
ノート: このセッティングはデバイスが内部ECCユニットを含んでいる場合のみ表示されます。
ノート: 内部ECCユニットはある領域にスペア領域データを上書きします。
スペア領域データの置き換えはデバイスのデータシートをご覧下さい。

USER AREA[ユーザー領域]スペシフィケーション:

ここでアクセスされるデバイス領域を指定することが出来ます。バッファ・データの**ブロック数**は デバイスに書き込まれ、**スタート・ブロック**から始まりバッファの最初からデバイスに書き込まれます。**最終ブロック**はユーザー・データのために使用することが出来るデバイスの最後のブロックを現わしています。特定の不正なブロック管理方法ではユーザー領域内の**最大許可無効ブロック数**がチェックされます。しかしイレースとブランク・チェック操作はこの設定とは別にデバイス全体で行われます。

REQUIRED VALID BLOCKS AREA[要求された有効ブロック領域]スペシフィケーション:

時にはデータの部分が指定された場所に正確に位置し, そして無効ブロック(例, ブート・コード)により 中断されないことが必要です。無効ブロックの有無をチェックしなければいけない**開始ブロック**と**ブロック数**を指定することが出来ます。この領域はユーザー領域の内部で指定されていなければいけません。このセッティングは**Check Required Valid Blocks Area**チェック-ボックスがチェックに されることで受付られます

CHECK MAX. ALLOWED NUMBER OF INVALID BLOCKS IN DEVICE[デバイスの無効ブロックの最大許可数のチェック] チェック:

ボックスをチェックすることで**デバイスの最大許可無効ブロック数 [Max. Allowed Number of Invalid Blocks in Device]**を指定することが出来ます。 指定された数以上の無効ブロックを持ったデバイスはリジェクトされます。

IF NEW INVALID BLOCK IS DEVELOPED[新しい無効ブロックが開発された場合]オプション:

デバイスのイレースと/又は、プログラミングでは新しい無効ブロックの開発が可能です。
これはNANDフラッシュ媒体の自然な特性です。新しい無効ブロック開発の増加に伴ってデバイスは消耗します。
通常、その様なブロックは無効とマークされそれ以上は扱われません。 操作は選択されたInvalid Block Management[無効ブロック管理]方法を継続します。この動作は**mark it invalid and continue operation[それに無効のマークをし操作続行]**セッティングに対応します。
時に新しい無効ブロックが間違って開発される場合があります。
一般的にこれはデバイスとZIFソケット(汚れている、又は、酸化汚染やパッケージ・アダプター の不良)の不完全接続が原因です。
この様な場合は、即座に操作を中止してZIFソケット, アダプター, デバイスを確認して下さい。この動作は**abort operation immediately[即座に操作を破棄]**セッティングと対応しています。
ノート: 選択されたプログラマーのみオプションが利用出来ます。

INCLUDE CHIP GAP INTO INVALID BLOCKS TABLES[Chip Gapを無効ブロック・テーブルに含める]チェック-ボックス:

もし、無効のブロックテーブルをデバイスにストアする何らかのアルゴリズムが選択されているなら、テーブルへのChip Gapに属すブロックを含めることを無効/有効にすることができます。
ノート: Chip Gapを含んだデバイスのみでオプションが利用出来ます。

OTP AREA[OTP領域]オプション:

操作からOTP(ワンタイム・プログラマブル)領域を含めることも除外することも出来ます。
もし、OTP領域処理が有効になっている場合、OTP領域はブランク・チェック、読み出し、ベリファイとプログラム操作に含まれます。自動的に消去(含む、プログラミングの前に消去)から除外されていますので、消去は出来ません。
もし、サポートされている場合、OTP領域プロテクション・セッティングも出来ます。これはOTP領域を以後の変更からロックします。一般的にこの操作は不可逆です。
ノート: 選択されたデバイスと/又は、プログラマーのみでオプションが利用出来ます(デバイスはOTP領域がサポートされていなければいけません)。

BLOCK PROTECTION[ブロック・プロテクション]セッティング:

プロック・プロテクション・セッティングのプログラミングが出来ます。操作は不可逆です。
この場合、再プログラミングされることを防ぎます。利用できるセッティングの種類はターゲット・デバイスによりサポートされている機能に依存します。
ノート: 選択されたデバイスと/又は、プログラマーのみでオプションが利用出来ます(デバイスはブロック・プロテクションがサポートされていなければいけません)。

RESERVED BLOCK AREA[リザーブされたブロック領域]オプション:

RBAテーブルのプログラミングを成功させるために領域を指定することが出来ます。**開始ブロック**から始まる**ブロック数**の領域をRBAテーブルのためにリザーブすることが出来ます。RBA領域から最初の2つの有効ブロックがRABテーブルの2つのコピーをストアするために使用されます。そのブロックが無効ブロックに置き換えられることを考慮にいれてRBA領域はユーザー領域の後ろに位置しなければいけません。

INVALID BLOCK INDICATION[無効ブロック表示]オプション:

このオプションを使用しデバイスで如何に認識し無効ブロックを表示するかを指定することが出来ます。
利用出来る2つのオプションがあります:

- **Use customized invalid blocks indication scheme[カスタマイズ無効ブロック表示スキーム]:**
もし、ユーザー自身の表示スキームを指定したい場合にこのチェックボックスをチェックします。
一般的に、もしBIバイト値が0xFFと異なる場合ブロックは無効と見なされます。
通常は0x00の値が無効ブロックを表示するために使用されます。
カスタマイズされたスキームの使用は**Alternative block validity indication byte value for invalid block[無効ブロックに対する代替ブロックの有効表示バイト値]**のために値を指定するためにプログラムと/又はイレース操作中に開発され発生した新しい無効ブロックを表示する時に使用されます。**Alternative block validity indication byte value for good block[有効ブロックに対する 代替ブロックの有効表示バイト値]**のために指定された値は0xFF値と同様に無効ブロック・テーブル 作成中に有効ブロックとしてインタープリットされます。
またページの範囲でBIバイト・オフセットを指定すること出来、そして、プログラマーが有効なBIバイトを探すべきページを指定出来ます。(ブロック内で最大2ページが受け付けられます)
- **Fill invalid block with predefined value[事前定義値で無効ブロックをフィル]:** 指定された値と全体の無効ブロックを埋める場合は、**Invalid block filling value[無効ブロックの充填値]** チェック・ボックスをチェックにします。
プログラムと/又は、イレース操作中に開発された新しい無効ブロックのみフィルされます。

TOLERANT VERIFICATION[トレランス・ベリフィケーション]オプション:
この機能は独自のECCアルゴリズムの使用を有効にします。そのアルゴリズムが
ユーザーのアルゴリズムが修正出来る機能があるならECCフレーズ・サイズ(通常512バイト)と
このフレームのシングルービット・エラー数を指定することが出来ます。
ベリファイ操作は定義されたフレームの範囲で指定されたエラー数を許容します。
スペア領域で相対するシングルービット・エラー数を許容します(スペア領域サイズに依存)

ONE TIME PROTECT AREA[ワン・タイム・プロテクト領域]オプション:
ある種のデバイスは**One Time Protect Area[ワン・タイム・プロテクト領域]**を持っています。
デフォルト・オーバーレイイング・モードと同様に個々のページに対して書き込みプロテクションを セットすることが可能です。
ワン・タイム・プロテクト領域に対するデータはバッファの最後に位置し、アドレスからの開始は デバイズのサイズと同じです。
ノート: デバイスの読み込みはAccess Method[アクセス・メソッド]ダイアログ
のワン・タイム・プロテクト領域セッティングに影響を及ぼします。
ノート: ワン・タイム・プロテクト領域のセッティングはデバイスがワン・タイム・プロテクト 領域を持っているときのみ表示されます。

LINUX MTD COMPATIBLE[リナックスMTD互換]オプション:
このセクションは**Linux MTD compatible** 無効ブロック管理方法(選択されたプログラマーのみ
で利用出来ます)のための数々のセッティングを含んでいます。
更に詳しくはアプリケーション・ノートのhttp://www.elnec.com/sw/an_elnec_linux_mtd.pdfをお読み下さい .

DEVICE OPERATION[デバイス操作]オプションの説明<Alt+0>メニュー:

ERASE BEFORE PROGRAMMING[プログラミングの前にイレース]オプション:
NANDフラッシュ・デバイスはイレースされた状態で出荷されていますので**Erase before programming [プログラミング前のイレース]**
は必須ではありません。しかし、内部コントローラの動作による理由からブランク・チェック(時間はデバイスにより数秒から数時間)
のために一度はイレース(通常は数秒)されることが望ましいと思います
ノート: 常にデバイス全体がイレースされます。

BLANK CHECK BEFORE PROGRAMMING[プログラミング前のブランク・チェック]オプション:
これは**Erase before programming[プログラミングの前にイレース]**と非常に密接に関係します。
もしデバイスが最初にイレースされますとブランク・チェックは必要ありません-それはデバイスの
内部コントローラにより行われます。従って、操作時間短縮のために**もしプログラミング前の
イレースが有効にされますとプログラミング前のブランク・チェックはスキップ**されます。
そうでない場合はデバイス全体のアドレス範囲がブランク・チェックされます。

VERIFY AFTER PROGRAMMING[プログラミング後のベリファイ]オプション:
プログラムの後のベリフィケーションを有効又は無効にすることが出来ます。
もし開発の段階で時間を短縮したい場合-(起こりうる結果を予測できる場合)においては
プログラミング後のベリフィケーションを無効に指定することが出来ます。
**大量生産ではいかなる場合もプログラミング後のベリフィケーションを無効にしないで下さい。
プログラミング操作のいずれにおいてもベリファイが行われません!**

TARGET DEVICE USES[ターゲット・デバイス使用]オプション:
このオプションは**Access Method[アクセス方法]** **<Alt+S>**ダイアログの **Invalid Block Indication options[無効ブロック表示オプション]**
に関連します。 無効ブロックを正しく認識させるためにはプログラマーはZIFソケットに装着された
実際のデバイスを使用するスキームを知っていなければいけません。
もし、特別な表示スキームを使用しない場合は**manufacturer original invalid blocks indication
scheme[マニュファクチャラー・オリジナル無効ブロック表示スキーム]**を使って下さい。
もし、**Access Method[アクセス方法]<Alt+S>**ダイアログでスペシャル表示スキーム を指定した場合は**customized invalid blocks indication
scheme[カスタマイズド・ 無効ブロック表示スキーム]**を使用して下さい。

スペシャルNANDフラッシュ・デバイス・コマンド(メニュー デバイス):

READ ONFI PARAMETER PAGE[リードONFIパラメータ・ページ]コマンド:
もし、デバイスがONFIスタンダードをサポートしている場合、このコマンドはデバイスから
ONFIパラメータ・ページを読み込むために使用することが出来ます。
リード・データはデコードされ良くアレンジされたフォームで出力ファイルにセーブされます。ONFIスタンダードに関する更に詳しい情報は
http://onfi.org/ をご覧下さい。特定のデバイスのONFIサポートの詳細はデバイスのデータシートを見て下さい
ノート: 選択されたプログラマーのみでオプションが利用出来ます。

READ JEDEC PARAMETER PAGE コマンド:
もし、デバイスがJEDECスタンダードをサポートしている場合、このコマンドはデバイスから
JEDECパラメータ・ページを読みだすために使用することが出来ます。
リード・データはデコードされ、そして、よく配置されたフォームで出力ファイルにセーブされます。JEDEC
standardに関する更に詳しい情報は http://www.jedec.org/ をご覧下さい。
JEDECがサポートされた特殊なデバイスに付いての詳細はデータシートをご覧下さい。
ノート: 選択されたプログラマーのみでオプションが利用出来ます。

CHECK INVALID BLOCKS[チェック無効ブロック]コマンド:
このコマンドは無効ブロック解析のために便利です。デバイスの全ブロックの状態をチェックし、
出力ファイルにデバイス内のチップ全体の無効のブロック配置を表にした概要を生成します。
ノート: 選択されたプログラマーのみでオプションが利用出来ます

Load / Save partition table(menu File)[ロード/セーブ・パーティション・テーブル](メニュー・ファイル):

ソフトウエアは幾つかのパーティション・テーブル定義ファイル形式を認識:

- **Qualcommパーティション定義ファイル(*.MBN):** このフォーマットはQualcomm Incorporated (U.S.A.)によって紹介されました。

これは各パーティションに対してパーティション開始、パーティション終了、使用されているパーティションの

サイズとコメント(項目ごとに4バイト)を指定するために16バイトを使用しています。詳しくはQualcomm社の関連ドキュメントを見て下さい。

- **カンマ・セパレート値(*.CSV):** このフォーマットは1つのパーティションを
指定するために1つのテキスト行を使用します。その行でパーティション開始、パーティション終了、
使用されるパーティションサイズ、スペシャル・オプテョンとコメントを指定することが出来ます。
各項目はカンマ、又は、セミコロンを使って分離されなければいけません。

- **Group Definition file(*.DEF)[グループ定義ファイル(*.DEF)]:**
これは実験的なサポートのみです。ユーザーからのフレグメント仕様に基づいてインプリメントされます。
ユーザーが何をするかを確実に理解出来るまでDEFフォーマットは使用しないで下さい。

最新バージョンのNANDフラッシュ関連ドキュメントのために定期的に (http://www.elnec.com/support/application-notes/
)をチェックして下さい。それらは更に有益な情報を含んでいます。