binho

アプリケーションノート

Microchip dsPIC33AK 上での CMIS I3C ターゲットの構築と Binho Supernova によるテスト

AN0019

文書情報

キーワード	AN0019, Supernova, CMIS, OIF, I3C, MIPI I3C Basic, SDR, I3C ターゲット, dsPIC33AK, dsPIC33AK256MPS306, MCC Melody, DMA, ENTDAA, IBI, GETSTATUS, 光モジュール
概要	CMIS 光モジュールは I3C ターゲットとしてホストに応答し、ホストは I2C ホストと同じ方法でそれを読み出します。 すなわち、バイトアドレスを書き込み、リピーテッド START を発行し、読み出します。dsPIC33AK256MPS306 の I3C ペリフェラルでは、 その書き込みの割り込みはリードヘッダへの応答が済んだ後に到着するため、その割り込みから応答をアームするターゲットは そのような読み出しをすべて NACK します。本アプリケーションノートでは、バイトアドレスがどこで間に合うタイミングで得られるか、 ファームウェアが代わりに受信 DMA からどのようにアームするか、そして Binho Supernova から測定した数値 （4 から 128 バイトの読み出しで 60 回中 60 回、インバンド割り込み有効時に 300 回中 300 回、リセット後も同様）を示します。 ホスト側のテストスクリプトを提供し、移植時に考慮しておく必要があるペリフェラルとドライバの挙動を列挙します。

Binho Inc.
42 Dore Street, Unit A, San Francisco, CA 94103

改訂 1.0
6 NOVEMBER 2026

検証範囲。セクション 8 のすべての結果は、セクション 3 で説明するベンチで測定したものです。ターゲットファームウェアはアセットのアーカイブに含まれていません。techsupport@binho.io までお問い合わせください。

1 はじめに

CMIS モジュールは 256 バイトのウィンドウを公開します。これは Lower Memory と Upper Memory の 128 バイトのページ 1 つから成り、 ページは `00h:127` のページバイトで選択されます。I2C では、ホストはバイトアドレスを書き込み、その後リピーテッド START で 読み出します。CMIS 向けに定義が進められている I3C 管理インタフェースはこのアクセスパターンを維持しているため、 I3C ターゲットペリフェラル上に構築したモジュールは、同じ転送の中でわずか数ビット時間前に到着したバイトアドレスに対する 読み出しに応答しなければなりません。

ペリフェラルが各バイトを到着時にファームウェアへ渡すのであれば、これは容易です。しかし dsPIC33AK256MPS306 の I3C ペリフェラルはコマンドとレスポンスに基づいて動作します。受信した転送はレスポンスとして 1 回だけ報告され、読み出しには 送信コマンドで応答しますが、その送信コマンドはリードヘッダが到着した時点ですでに有効でなければなりません。 本アプリケーションノートは、この 2 つをかみ合わせる方法について説明します。

1.1 対象範囲

- dsPIC33AK 上の CMIS レジスタモデル：ページング、セクション 8.2.15 の非サポートページの置き換え、 Appendix B.1.2 の 128 バイトセグメント内でのラップ、および Flags Summary バイト `00h:4` を運ぶ IBI。
- 2.5 MHz プッシュプルでの、書き込み + Sr + 読み出し（コンバインド）と、書き込み、STOP、読み出し（スプリット）への応答。
- ダイナミックアドレス割り当て（DAA）、RSTDAA、および Direct GET CCC。
- 本ノートのすべてのチェックを Supernova から実行するホストスクリプト。

対象外：CDB、ファームウェアのダウンロード、HDR モード、および CMIS 制御レジスタのモジュール側での意味。 レジスタイメージは、ページ選択とホストが書き込むバイトを除いて静的です。

1.2 対象デバイス

表 1. 対象デバイス

ファミリ	部品例	識別子	根拠
I3C 搭載 dsPIC33AK MPS	dsPIC33AK256MPS306（EV02E57A DIM）	PID `02 36 43 4D 49 53`	テスト済み
I3C 非搭載 dsPIC33AK MP	dsPIC33AK512MPS506 およびファミリ	-	対象外：I3C ペリフェラルなし
同じ I3C ペリフェラルを持つその他の部品	-	-	未テスト

1.3 関連資料

表 2. 関連資料

資料	タイトル
OIF-CMIS-05.3	Common Management Interface Specification、リビジョン 5.3
MIPI I3C Basic v1.1.1	MIPI I3C Basic Specification、バージョン 1.1.1
DS70005539	dsPIC33AK MPS ファミリ データシート、セクション 24（I3C）
AN0009	Binho Supernova を使用した I3C 経由の CMIS 光モジュールの管理

2 システム概要

Supernova は I3C コントローラであり、バス上の唯一のコントローラです。dsPIC33AK はターゲットで、スタティックアドレスは CMIS モジュールアドレスである `0x50` です。コントローラは ENTDAA でダイナミックアドレスを割り当て（ベンチでは `0x08` を 割り当てます）、その後 SDR で読み書きします。

ファームウェアの形を決める唯一の事実は、コンバインドリードのバイトアドレスがいつ見えるようになるかです。I3C 割り込みを 無効にし、100 MHz タイマを基準にペリフェラルをポーリングし、ロジックアナライザと時刻を合わせて測定した結果は次のとおりです。

表 3. 2.5 MHz プッシュプルでのコンバインドリード、START からの時間

イベント	時間
バイトアドレスの T ビット	22.3 µs
受信 FIFO 内のバイトアドレス	23.8 µs
リピーテッド START	24.1 µs
リードヘッダの ACK スロット	~27.4 µs
書き込みのレスポンス割り込み	ACK スロットの後

書き込みのレスポンス割り込みからアームした送信コマンドは常に遅すぎ、ターゲットは読み出しを NACK します。バイトアドレス 自体は ACK スロットの 3.6 µs 前に得られ、コマンドのアームには 1.15 から 1.26 µs かかります。そこでファームウェアは 受信 DMA からアームします。受信したワードを RAM へ転送するチャネルが、I3C 割り込みより高い優先度でワードごとに割り込みを 発生させ、その割り込みが最初のバイトから送信コマンドをアームします。

その他のことはすべて、このペリフェラルとその MCC Melody ドライバの 3 つの特性から導かれます。

- 1 つの送信コマンドは 1 回の読み出しに応答します。ファームウェアはバイトアドレスからセグメントの終わりまでの 128 バイトのランをアームし、コントローラが読み出しを早めに終了します。

- 送信バッファは 8 ワード（32 バイト）を保持します。ランの残りはメインループから供給されます。32 バイトの送信に かかる時間（約 115 µs）より長くメインループを離れさせる処理があると、読み出しは 32 バイトで打ち切られます。
- 受信を再アームすると受信 FIFO がリセットされ、アーム済みの送信コマンドも破棄されます。受信は読み出しの後にのみ 再アームします。

3 ハードウェアのセットアップ

3.1 必要な機器

- Binho Supernova、ファームウェア 4.6.2、I3C HV ポート。
- EV02E57A dsPIC33AK256MPS306 DIM を装着した Microchip EV74H48A Curiosity プラットフォーム。その PKoB4 で部品の 書き込みを行い、UART も提供します。
- オプション：SCL と SDA に接続したロジックアナライザ。本ノートのキャプチャは Saleae Logic を使用し 12 MS/s で 取得しました。

3.2 接続

表 4. Supernova とターゲット間の接続

Supernova I3C ポート	dsPIC33AK DIM	必須
SCL	I3CSCL1（RC5）	はい
SDA	I3CSDA1（RC4）	はい
GND	GND	はい
I3C レール、3.3 V	接続しない。DIM は Curiosity ボードから給電	-
-	UART1 TX（RD0）、PKoB4 の仮想 COM ポートに出力	オプション

3.3 バスに関する考慮事項

バスは 3.3 V で動作します。プッシュプル 2.5 MHz は Supernova が提供する最低のレートであり、ここで測定した唯一のレート です。より速いレートでは表 3 の 3.6 µs のウィンドウが短くなり、12.5 MHz ではコマンドのアームにかかる時間より短く なります。

4 ソフトウェアのセットアップ

ターゲットファームウェア：

- MPLAB X v6.35 と MPLAB IPE、XC-DSC v4.00、dsPIC33AK-MP DFP 1.4.260。
- MCC Melody I3C ターゲットドライバ（セクション 5.6 のローカル変更を適用）。

ホスト：

- Python 3 と pycosmicsdk（最新リリース）、および本ノートのアセットに含まれる `cmis_target_check.py`。

```
pip install --upgrade pycosmicsdk
```

5 ターゲットファームウェアの設計

5.1 レジスタモデル

Lower Memory と Upper Memory のページ `00h` は、CMIS 5.3 Appendix C の例のイメージに基づいています。`00h:127` の ページ選択は `00h` と `01h` を受け付け、それ以外の値は `00h` に置き換えられます（セクション 8.2.15）。各セグメントは 連続して 2 回格納され、さらに 1 から 3 バイトずらしたコピーも格納されるため、任意のバイトアドレスからの 128 バイトの ランは 1 つのアラインされたバッファとなり、セグメント内の B.1.2 ラップにはコードが不要です。

5.2 識別情報

表 5. コントローラから見える識別情報

項目	値
スタティックアドレス	`0x50`
PID	`02 36 43 4D 49 53`（MIPI メーカー `0x11B`、部品 `"CMIS"`）
BCR	`0x17`：IBI 対応、IBI は 1 バイトを運ぶ、最大データ速度制限あり
DCR	`0x00`

5.3 読み出しのアーム

1. 受信 DMA チャネルは受信した各ワードを `rxbuf` に転送します。その完了割り込みは有効で、優先度は 5 であり、 優先度 4 の I3C 割り込みより高くなっています。
2. 書き込みの最初のワードで、DMA 割り込みは `rxbuf[0]` をバイトアドレスとして取得し、そのアドレスからのランの 送信コマンドをアームします。
3. 書き込みのレスポンスが到着し、それがアドレスのみを運んでいた場合、コマンドはアームされたままとなり、受信は 読み出しの完了後に再アームされます。
4. 書き込みがデータを運んでいた場合、拡張送信バッファをリセットしてコマンドを取り消し、受信を直ちに再アームします。
5. リードヘッダの時点でまだ何もアームされていない場合、ペリフェラルはリードリクエストを発生させます。ファームウェアは 遅れてアームされたものをすべて破棄し、`I3C1_Target_ReceiveCancel()` で受信を再アームします。この場合、書き込みの レスポンスは決して来ないためです。

5.4 長い読み出しへの供給

送信バッファにはあらかじめ 8 ワードを充填しておきます。`I3C1_Target_TxRefill()` が残りをメインループから供給し、 その実行にかかる数命令の間は I3C 割り込みと DMA 割り込みをマスクします。ブリングアップ用ファームウェアの UART ステータス行はデフォルトでオフです。9600 baud でのブロッキング `printf` が 1 秒あたり約 110 ms メインループを占有し、 128 バイトの読み出しを 32 バイトで打ち切っていたためです。

5.5 イベント

IBI が有効な間、IBI は 1 秒に 1 回、`00h:4` を必須のデータバイトとして運びます。ペリフェラルは I3C Basic の規定どおり IBI が有効な状態でリセットから復帰し、そのイネーブルに書き込まれた 0 を保持しません。IBI を無効にするには DISEC を 使用します。

5.6 Melody ドライバへの変更

Table: `mcc/` 内のローカル変更（`binho:` でマーク）

変更	理由
`BufferTransmit` が 8 ワードを事前充填し、`I3C1_Target_TxRefill()` が残りを供給	8 ワードを超えると DMA パスがワード 7 を 2 回書き込んで先へ進まず、コマンドが拒否されていたため
`I3C1_Target_ReceiveCancel()`	書き込み + Sr + NACK されたリードヘッダの後、書き込みのレスポンスは来ない。ドライバは受信を処理中としてマークしたままにし、以降のすべての受信を拒否していた
完了コールバックの前に `rxInProgress` をクリア	コールバックが受信を再アームできるようにするため
スタティックアドレス `0x50`	CMIS モジュールアドレス

6 手順

ステップ 1：ファームウェアフォルダからターゲットをビルドし、書き込みます。

```
make flash
```

ビルドは MPLAB IPE の `Program Succeeded.` で終了します。

ステップ 2：表 4 のとおり、Supernova の I3C ポートを DIM に配線します。

ステップ 3：書き込み + Sr + 読み出しで 128 バイトを 60 回読み出します。

```
python cmis_target_check.py reads --length 128 --count 60
```

```
init bus: FW_I3C_BUS_INIT_NACK_RSTDAA
target DA 0x08, PID 02 36 43 4d 49 53
..........................................................
reads 60/60  (combined, 128 B, IBI off)
```

`init bus` の行は、リセット後の最初の RSTDAA が NACK されたことを示します（セクション 8.3）。スクリプトは再度 初期化します。

ステップ 4：IBI を有効にして同じことを行います（読み出しの前に ENEC）。

```
python cmis_target_check.py reads --length 128 --count 300 --ibi
```

```
reads 299/300  (combined, 128 B, IBI on)
```

ドットの行の `F` はエラーが発生した読み出しを、`B` は誤ったデータを返した読み出しを示します。

ステップ 5：データ書き込みと読み出しの混在：各ラウンドで `7Fh=00h` と `7Eh=00h` を書き込み、その後 `7Fh` （1 バイト）と `80h`（16 バイト）を読み出し、イメージと照合します。

```
python cmis_target_check.py mixed --count 60
```

```
rounds 60/60
```

ステップ 6：アドレスのリセットと再割り当て：

```
python cmis_target_check.py daa --cycles 10
```

```
0: RSTDAA SUCCESS, GETSTATUS at old DA FW_I3C_NACK_ADDRESS, read after ENTDAA ok
...
cycles 10/10
```

ステップ 7：識別情報と Direct GET CCC：

```
python cmis_target_check.py ccc --getstatus 100
```

```
GETBCR   SUCCESS               17
GETDCR   SUCCESS               00
GETPID   SUCCESS               02 36 43 4d 49 53
GETMXDS  SUCCESS               00 00 00 00 01
GETMRL   FW_I3C_NACK_ADDRESS
GETMWL   SUCCESS               ff
GETSTATUS 100/100  {'SUCCESS': 100}
```

Direct GET は NACK された `7Eh` ヘッダで失敗することがあります（セクション 8.3）。ここでは GETMRL が失敗しました。 もう一度実行してください。

7 リファレンス

Table: `cmis_target_check.py` のコマンド

コマンド	動作
`reads --length N --count N [--split] [--ibi]`	`00h`、`10h`、`7Ch`、`80h`、`A0h`、`FCh` を順に読み出し、イメージと比較します。 `--split` を指定しない限りコンバインド。 `--ibi` は先に ENEC を送り、指定しなければ DISEC を送ります
`mixed --count N`	`7Fh` と `7Eh` にデータを書き込み、その後 `7Fh` と `80h` をコンバインドリードして照合します
`ccc [--getstatus N]`	GETBCR、GETDCR、GETPID、GETMXDS、GETMRL、GETMWL、続いて GETSTATUS を N 回
`daa --cycles N`	RSTDAA、旧アドレスが NACK することを確認、ENTDAA、読み出しを確認
`i2c`	アドレス割り当ての前に、I2C モードで `0x50` の `00h:0-3` と `00h:46-48` を読み出します

共通オプション：`--serial`、`--push-pull`（デフォルト `PUSH_PULL_2_5_MHZ_DC_25`）、`--voltage-mv`（デフォルト 3300）、`--image`（デフォルトはスクリプトと同じ場所の `cmis_image.h`）。各コマンドは RSTDAA と ENTDAA から始まり、PID 内の `"CMIS"` によってターゲットを見つけます。

8 測定結果

8.1 テスト構成

表 6. テスト構成

項目	バージョン
Supernova	シリアル 4F970418、ファームウェア 4.6.2
ターゲット	EV74H48A 上の EV02E57A DIM
ツールチェーン	XC-DSC v4.00、DFP 1.4.260、MPLAB X v6.35
ホスト	Ubuntu 26.04。セッション 1 と 2：Python 3.13、binhosupernova 4.2.0。セッション 3：Python 3.14.4、pycosmicsdk 1.9.1
バス	3.3 V、プッシュプル 2.5 MHz、オープンドレイン 1 MHz

8.2 結果

3 つのセッションを実施しました。セッション 2 は書き込み（リセット）の後に開始し、セッション 3（2026 年 11 月 6 日）は 各テストの前にターゲットを書き込み、ホストには pycosmicsdk を使用しました。

表 7. 結果

テスト	セッション 1	セッション 2	セッション 3
コンバインドリード、4 / 16 / 32 / 128 B、IBI オフ	各 60/60	60/60（4 および 128 B）	各 60/60
コンバインドリード、128 B、IBI オン	300/300	300/300	299/300
スプリットリード、4 / 128 B	各 60/60	60/60（128 B）	60/60（128 B）
データ書き込みとコンバインドリード、データ照合あり	60/60 ラウンド	60/60 ラウンド	60/60 ラウンド
RSTDAA、旧アドレスが NACK、ENTDAA、読み出し	10/10	10/10	10/10
GETSTATUS	97/100	94/100	100/100

受信 DMA からのアームを導入する前は、同じコンバインドリードは 60 回中 0 回でした。

8.3 ホストが想定すべき挙動

- **ターゲットのリセット後の最初の転送は NACK されます**。5 回のリセットのそれぞれの後、最初の `7Eh` ヘッダは NACK され、 2 回目は ACK されました。DS70005539 のセクション 24.4.5

には、モジュールは SCL クロックを検出して初めてリセットから 抜けると記載されています。1 回だけプローブして NACK を「I3C ターゲットなし」と判断するホストは、このモジュールを 見落とします。プローブを再試行してください。

- **一部の Direct GET CCC は NACK された `7Eh` で失敗します**。あるセッションでは、Direct GET に続く `7Eh` ヘッダの 約 7 回に 1 回が NACK されました（1200 回の GET で 173 回失敗し、173 回すべてがワイヤ上で `7E/W NACK` として 確認できました）。別のセッションでは 500 回中 500 回が成功しました。プライベートリードには影響しませんでした。 NACK で失敗した GET は再試行してください。
- **GETMRL は 255 を報告し、SETMRL は適用されません**。ファームウェアは 128 を設定しますが、ペリフェラルが自ら GETMRL に 255 で応答します。SETMRL 16 の後も、32 バイトの読み出しは 32 バイトを返します。

8.4 アドレス割り当て前の I2C 読み出し

ENTDAA の前、ターゲットは `0x50` で I2C に応答します。Supernova の I2C ポート（3.3 V で 2.2 kOhm のプルアップを持つ 通常の I2C コントローラ）から、各実行の前にターゲットを書き込んだうえで、`00h`、`10h`、`7Ch`、`80h`、`A0h`、`FCh` で 3、4、16、128 バイトを読み出したところ、100 kHz と 400 kHz のどちらでも各 30/30 で、データはイメージと一致しました。 各転送を `7Eh` ヘッダで開始する I2C モードの I3C ポートからは、セッション 1 では 30 回中 8 回が正しく、セッション 3 では リセット後の最初の読み出しが NACK され、2 回目は `00h:46-48` から `80 00 00` を返しました。

9 トラブルシューティング

表 8. トラブルシューティング

症状	考えられる原因と対処
すべてのコンバインドリードが `FW_I3C_NACK_ADDRESS` で失敗する	コマンドのアームが遅すぎます。DMA 割り込みの優先度が I3C 割り込みより高いこと、およびその中で時間のかかる処理が実行されていないことを確認してください
128 バイトの読み出しが失敗し、アナライザに 308 個の SCL クロックが表示される	読み出しが 32 バイトで打ち切られています。メインループが送信バッファに供給していません。ブロッキング `printf` などを取り除いてください
読み出しの失敗後、リセットするまでターゲットが応答しなくなる	受信が処理中としてマークされたままです。ファームウェアに `I3C1_Target_ReceiveCancel()` が必要です
電源投入後の最初の初期化が `FW_I3C_BUS_INIT_NACK_RSTDAA` で失敗する	リセット後の最初の転送は NACK されます。再度初期化してください
MPLAB IPE が "Connection Failed" または `Transmission on endpoint 2 failed (err = -150)` を報告する	PKoB4 の USB を抜き差しする必要があります

10 略語

表 9. 略語

用語	意味
BCR	Bus Characteristics Register（バス特性レジスタ）
CCC	Common Command Code（共通コマンドコード）
CMIS	Common Management Interface Specification（共通管理インタフェース仕様）
DA	Dynamic Address（ダイナミックアドレス）
DCR	Device Characteristics Register（デバイス特性レジスタ）
DMA	Direct Memory Access（ダイレクトメモリアクセス）
ENTDAA	Enter Dynamic Address Assignment（ダイナミックアドレス割り当て開始）
IBI	In-Band Interrupt（インバンド割り込み）
MRL	Max Read Length（最大読み出し長）
PID	Provisioned ID（プロビジョンド ID）
RSTDAA	Reset Dynamic Address Assignment（ダイナミックアドレス割り当てのリセット）
SDR	Single Data Rate（シングルデータレート）
Sr	Repeated START（リピーテッド START）

11 本書のソースコードに関する注記

本書に示し、付属アーカイブで提供するサンプルコードは、評価および参考を目的として提供されます。MIT ライセンスの下で 公開されており、その条文はアーカイブに含まれています。

アーカイブ `AN0019-assets.zip` には `cmis_target_check.py` と `cmis_image.h` が含まれています。

12 改訂履歴

表 10. 改訂履歴

リビジョン	日付	説明
0.1	29 October 2026	社内レビュー用ドラフト。
0.2	6 November 2026	ホストスクリプトを pycosmicsdk に移行。セッション 3 の結果。アドレス割り当て前の I2C 読み出しを通常の I2C コントローラから測定。
1.0	6 November 2026	公開。

法的情報（英語）

Definitions

Draft

A document marked draft is under internal review and subject to formal approval. Binho Inc. makes no representation or warranty as to its accuracy or completeness and accepts no liability arising from its use.

Disclaimers

Limited warranty and liability

Information in this document is believed to be accurate and reliable. Binho Inc. makes no representation or warranty, express or implied, as to its accuracy or completeness, accepts no liability for the consequences of its use, and takes no responsibility for content originating outside Binho Inc.

In no event shall Binho Inc. be liable for any indirect, incidental, punitive, special or consequential damages, including without limitation lost profits, lost savings, business interruption, or costs of removing or replacing any product, whether or not based on tort, warranty, breach of contract or any other legal theory. Notwithstanding any damages a customer might incur, the aggregate and cumulative liability of Binho Inc. is limited in accordance with its terms and conditions of commercial sale.

Right to make changes

Binho Inc. reserves the right to change the information in this document, including specifications and product descriptions, at any time and without notice. This document supersedes all information supplied prior to its publication.

Suitability for use

Binho Inc. products are not designed, authorized or warranted for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure of a Binho Inc. product can reasonably be expected to result in personal injury, death, or severe property or environmental damage. Any such inclusion or use is at the customer's own risk and Binho Inc. accepts no liability for it.

Applications

Applications described here are illustrative only. Binho Inc. makes no warranty that they are suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their own applications and products, for appropriate design and operating safeguards, and for determining whether a Binho Inc. product is suitable and fit for their purpose. Binho Inc. accepts no liability for assistance with applications or customer product design.

Third-party products and specifications

This document refers to products, boards, specifications and documentation supplied by third parties, which remain the responsibility of their respective owners. Binho Inc. makes no warranty regarding them and their behavior may change without notice. Register maps, protocol details and device identifiers reproduced here were observed on the specific hardware and firmware versions stated in this document and must be confirmed against the vendor's own documentation for the reader's part.

Example code

Source code supplied with this document is provided as an example, for evaluation and reference. It is not qualified for production use, has not been developed under a functional safety or security process, and is licensed under the terms accompanying it in the distribution archive.

Terms and conditions of commercial sale

Binho Inc. products are sold subject to the general terms and conditions of commercial sale published by Binho Inc., unless otherwise agreed in a valid written individual agreement, in which case only the terms of that agreement apply.

Export control

This document and the items it describes may be subject to export control regulations. Export may require prior authorization from the competent authorities.

Security

All products may be subject to unidentified vulnerabilities or may support security standards with known limitations. The customer is responsible for the design and operation of its applications and products throughout their lifecycle to reduce the effect of such vulnerabilities, and for compliance with all legal, regulatory and security requirements concerning its products.

Trademarks

All referenced brands, product names, service names and trademarks are the property of their respective owners. **Binho** and the Binho logo are trademarks of Binho Inc. **I3C** および **MIPI** は MIPI Alliance, Inc. の商標または登録商標です。 **CMIS** および **OIF** は Optical Internetworking Forum の商標または登録商標です。 **Microchip**、**dsPIC**、**MPLAB** および **Curiosity** は Microchip Technology Incorporated の商標または登録商標です。**Saleae** および **Logic** は Saleae, Inc. の商標です。