本ドキュメントは Cypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、「MB」から始まるシリーズ名，品名およびオーダ型格が記載されておりますが、これらはすべて「CY」から始まるシリーズ名，品名およびオーダ型格として、新規および既存のお客様に引き続き提供してまいります。

オーダ型格の調べ方について

1. www.cypress.com/pcnにアクセスしてください。
2. SEARCH PCNS フィールドに、オーダ型格などのキーワードを入力し、「Apply」をクリックしてください。
3. 該当するタイトル(Title)をクリックしてください。
4. 「Affected Parts List」ファイルを開いてください。
   当該ファイルに記載されている各種変更情報をご利用ください。

詳しいお問い合わせ先

Cypress 製品およびそのソリューションの詳細につきましては、お近くの営業所へお問い合わせください。

サイプレスについて

サイプレスは、世界で最も革新的な車載や産業機器、スマート家電、民生機器および医療機器製品向けに、最先端の組み込みシステム ソリューションを提供するリーディング カンパニーです。サイプレスのマイクロコントローラーや、アナログ IC、ワイヤレスおよび USB ベースのコネクティビティ ソリューション、高い信頼性と高性能を提供するメモリ製品は、各種機器メーカーの差異化製品の開発と早期市場参入を支援します。サイプレスは、ベストクラスのサポートと開発リソースをグローバルに提供することで、彼らが従来市場を破壊しまったく新しい製品カテゴリを歴史的なスピードで市場投入できるよう支援します。詳細はサイプレスのウェブサイト (japan.cypress.com) をご覧ください。

AN204364

New 8FX ファミリ MB95330H/630H シリーズ IEC 60730 CLASS B セルフテストライブラリ

シリーズ名	品種型格
MB95330H	MB95F334K
MB95630H	MB95F636K

このアプリケーションノートは与えられたライブラリ関数の使用法および実装法について説明しています。まず, IEC60730 CLASS B の要件を示し, 次にどのようにそれを実現するかを説明します。最後に, テスト機能を実際のシステムに統合する方法を例示します。

Contents

1 はじめに 1
1.1 目的 1
1.2 定義, 頭字語, および略語 1
1.3 文書の概要 1
1.4 参照文書 2
2 IEC60730 CLASS B の要件 2
2.1 IEC60730 について 2
2.2 IEC60730 STL のテスト項目 3
3 IEC60730 CLASS B STL 概要 4
4 IEC60730 CLASS B STL API 4
4.1 CPU レジスタテスト 4
4.2 CPU PC テスト 6
4.3 割込みテスト 7
4.4 クロックテスト 8
4.5 不揮発性メモリ (ROM) のテスト 10
4.6 揮発性メモリ (RAM) テスト 12
4.7 IO テスト 13
4.8 A/D テスト 15
5 STL の使用例 16
5.1 プロジェクトの構造 16
6 New 8FX シリーズ MCU 18
6.1 New 8FX シリーズ MCU について 18
6.2 New 8FX IEC60730 STL デモプロジェクトについて 19
改訂履歴 20
セールス, ソリューションおよび法律情報 21

1 はじめに

1.1 目的

このアプリケーションノートは与えられたライブラリ関数の使用法および実装法について説明しています。まず, IEC60730 CLASS B の要件を示し, 次にどのようにそれを実現するかを説明します。最後に, テスト機能を実際のシステムに統合する方法を例示します。

1.2 定義, 頭字語, および略語

API アプリケーションプログラミングインタフェース

STL セルフテストライブラリ

1.3 文書の概要

この文書は次のような構成になっています。

第 2 章は IEC60730 CLASS B の要件について説明しています。

第 3 章は IEC60730 CLASS B の STL の概要について説明しています。

第 4 章は IEC60730 CLASS B の STL の API について説明しています。

第 5 章は STL の使用例を説明しています。

第 6 章はこのアプリケーションでの New 8FX シリーズ MCU について説明しています。

1.4 参照文書

[1]. IEC 60730-1 Edition3.2, 2007

[2]. F^{2}MC-8FX MB95330H シリーズ ハードウェアマニュアル (CM26-10126-1J)

[3]. F^{2}MC-8FX MB95330H シリーズ データシート (DS07–12629–1J)

[4]. F^{2}MC-8FX MB95630H シリーズ ハードウェアマニュアル (MN702-00009-1v0-J)

[5]. F^{2}MC-8FX MB95630H シリーズ データシート (DS702-00009-1v0-J)

[6]. FM3 MB9B500 Series IEC60730 CLASS B 準拠 セルフテストライブラリアプリケーションノート (AN706-00037-1v0-J)

2 IEC60730 CLASS B の要件

2.1 IEC60730 について

International Electrotechnical Commission (IEC) は各国の電子技術委員会 (IEC 国内委員会) により構成された標準化のための国際組織です。2007 年 10 月以降, EU 市場向けの家電製品は IEC が定めた標準規格 IEC60730 に準拠することが必要になりました。

IEC60730 の付属書 H は特に, 「ソフトウェアを使用した制御」の要件を規定しています。付属書 H の CLASS B コントローラ向けソフトウェア要件は, 「安全ではない動作を防止することを目的とした機能を規制」します。家庭用電化製品の製造者は, これらの要件を満たすためにどのように製品を設計するかを考慮する必要があります。

付属書 H では, ソフトウェアに関連した標準項目を CLASS A, B, または C に分類しています。

- CLASS A : 湿度制御, 照明制御, タイマ等, 機器の安全に関わらないと想定される制御機能
- CLASS B : 洗濯機用の熱電開閉器またはドアロックのように, 機器にソフトウェア障害以外の障害が発生した場合の災害防止を意図したコードを含むソフトウェア
- CLASS C : 密閉式水加熱システム用の熱電開閉器等, 他の保護装置を使用せず災害を防止することを意図したコードを含むソフトウェア

2.2 IEC60730 STL のテスト項目

IEC60730 で定義された仕様では，ソフトウェア CLASS B または C に区分される機能を有する制御は，ソフトウェアに関連した障害/エラーを防止し管理するための対策を，安全に関係するデータ中およびソフトウェアの安全に関するセグメント中で実施することを要求しています。これは，当該ソフトウェアがマイクロコントローラ内外の障害を検出する試験方法を採用しなければならないことを意味します。

New 8FX IEC60730 セルフテストライブラリ (STL) は，MB95F330H シリーズ MCU および MB95630H シリーズ MCU に対するソフトウェア CLASS B の要件 (これは標準に示された IEC60730 の要件の大部分をカバーしています) に重点を置いています。CLASS B のコントローラに対しては，試験すべきエレメント，採用すべき手法および実現すべき定義を，付属書 H の表 H.11.12.7 のまとめとして次の表に示してあります。

表 2-1. IEC60730 STL テスト項目

コンポーネント	障害/エラー	STL で使用する手法	定義	STL での有無
1. CPU 1.1 レジスタ 1.2 プログラムカウンタ	 スタックエラー スタックエラー	 スタティックメモリテスト プログラムシーケンスの論理監視	 H. 2.19.6 H.2.18.10.2	 ○ ○
2. 割込み	割込みせず，または頻繁すぎる割込み	タイムスロット監視	H.2.18.10.4	○
3. クロック	誤った周波数	周波数監視	H.2.18.10.1	○
4. メモリ 4.1. 不揮発性メモリ 4.2. 揮発性メモリ 4.3. アドレス [1]	 全 1 ビット不良 DC 不良 スタックエラー	 冗長性チェック スタティックメモリテスト 冗長性チェック	 H.2.19.3.2 H.2.19.6 -	 ○ ○ ○
5. 内部データパス [2]	スタックエラー	-	-	×
6.外部通信	ハミング距離 3	冗長性チェック	H.2.19.4.2	×
7. 入出力周辺機器 7.1 デジタル I/O 7.2 A/D	 機能エラー 機能エラー	 出力照合 入力比較	 H.2.18.12 H.2.18.8	 ○ ○

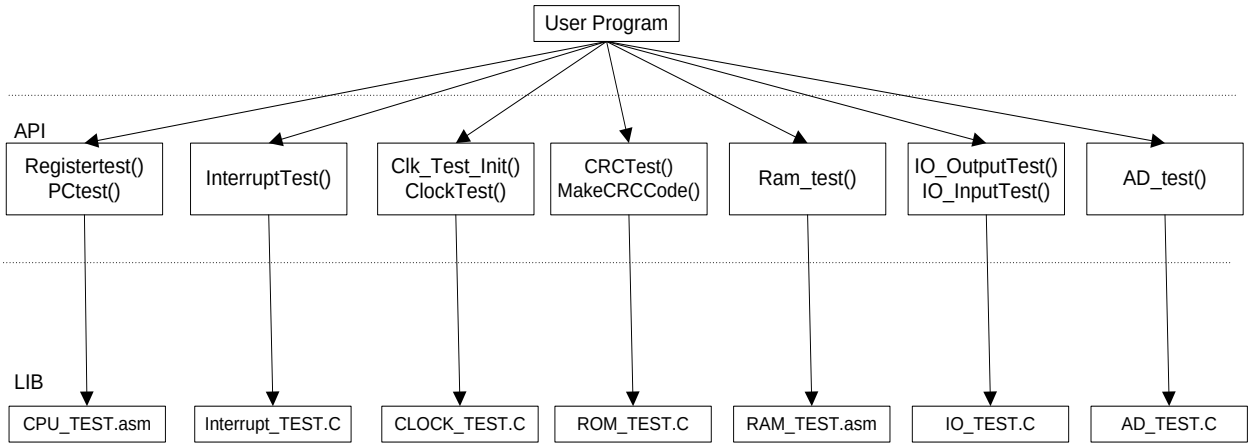
注意:

1. アドレス試験は不揮発性および揮発性メモリの試験法により部分的にカバーすることが可能です。たとえば，2 個のセルが同じアドレスにマッピングされるエラーはCRCテストで不揮発性メモリをテストする際に確認できます。
2. 内部データパスは外部メモリ使用時にのみテストされます。
3. 本 STL では外部通信テストは含まれていません。しかし，外部通信データは不揮発性メモリテストと同様の方法でテスト可能です。

3 IEC60730 CLASS B STL 概要

次の図に示すように, STL のブロック図には CPU, 割込み, クロック, メモリおよび入出力周辺モジュールが含まれています。この図は STL のファイル構造とソフトウェア API を表しています。STL は C 言語およびアセンブリ言語でコーディングされています。

図 3-1. IEC60730 CLASS B STL のブロック図



4 IEC60730 CLASS B STL API

4.1 CPU レジスタテスト

New 8FX MCU には 13 個のコアレジスタがあり, 読み書きが可能です。これらのレジスタをテストする必要があります。

表 4-1. New 8FX MCU レジスタテスト

レジスタ名	テストするビット
R0～R7	[7:0]
A (アキュームレータ)	[15:0]
T (テンポラリアキュームレータ)	[15:0]
IX (インデックスレジスタ)	[15:0]
EP (エクストラポインタ)	[15:0]
SP (スタックポインタ)	[15:0]

4.1.1 テストの説明

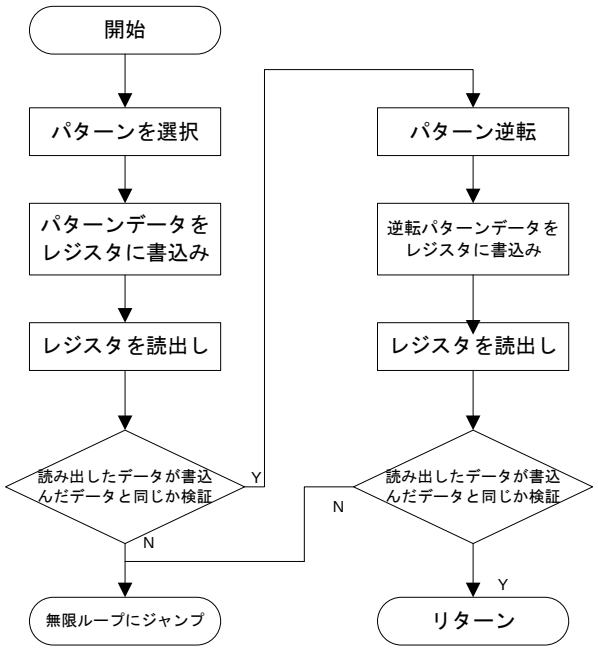
IEC60730-1 の表 H.11.12.7 に示すように, レジスタは『スタックエラー』のチェックを受けなければなりません。レジスタテストの実行には単純なチェッカーボード法を使用しますが, これはスタックエラー検出に有効な方法です。

このテストは破壊的なため、システムがリセットされる際にスタートアップファイルで呼び出される必要があります。

直接にレジスタにアクセスするため, レジスタテストにはアセンブリ言語が使用されています。またこのテストは非常に重要なので, レジスタテストでエラーが検出された場合は,プログラムが無限ループに入るように設計されています。

レジスタテストのフローチャートを次の図に示します。

図 4-1. レジスタテストのフローチャート



4.1.2 API の定義

名称	Registertest
パラメータ	なし
リターン	なし

説明:

Registertest()は CPU レジスタの書き込みと読み出しのテストを行います。アセンブリ言語により, 2 種類のテストパターンをレジスタに書き込みおよび読み出しを行います。この関数は R0～R7 (ビット 0～ビット 7) および特殊レジスタ (A, T, IX, EP, SP) を含むすべてのレジスタのテストを行います。

4.2 CPU PC テスト

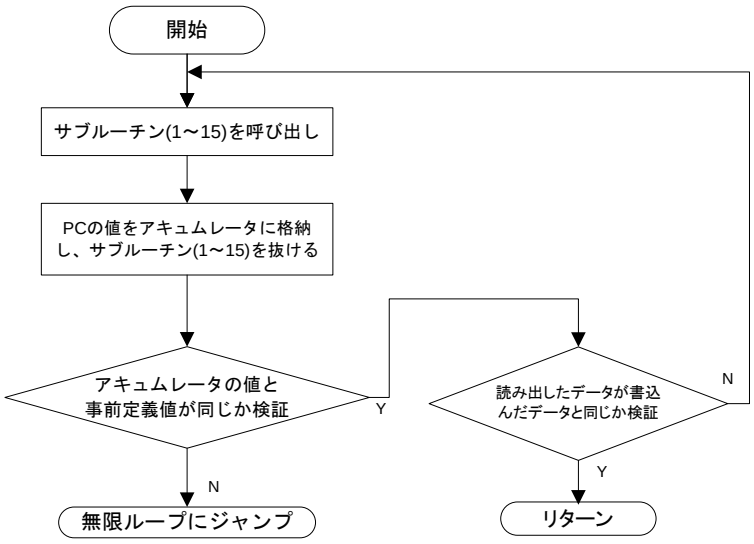
4.2.1 テストの説明

IEC60730-1 ファイルの表 H.11.12.7 に示すように, PC は『スタックエラー』のチェックを受けなければなりません。PC テストは 15 個のサブルーチンを使用し, 各サブルーチンから得られる PC の値が事前定義値と同じであるかどうかを検証します。このテストは破壊的なため, システムがリセットされた際にスタートアップファイルで呼び出される必要があります。

PC レジスタに直接アクセスするために, PC テストにはアセンブリ言語が使用されています。またこのテストは非常に重要なので, PC テストでエラーが検出された場合プログラムが無限ループに入るように設計されています。

PC テストのフローチャートを次の図に示します。

図 4-2. PC テストのフローチャート



4.2.2 API の定義

名称	PCtest
パラメータ	なし
リターン	なし

説明:

PCtest() は 15 個のサブルーチン (MB95330H シリーズ向け STL のサブルーチンは 6 個) を使用します。さまざまな領域でサブルーチンにジャンプしてサブルーチンアドレスを取得し, 得られた PC アドレスが事前定義値と同じかどうかを検証します。

4.3 割込みテスト

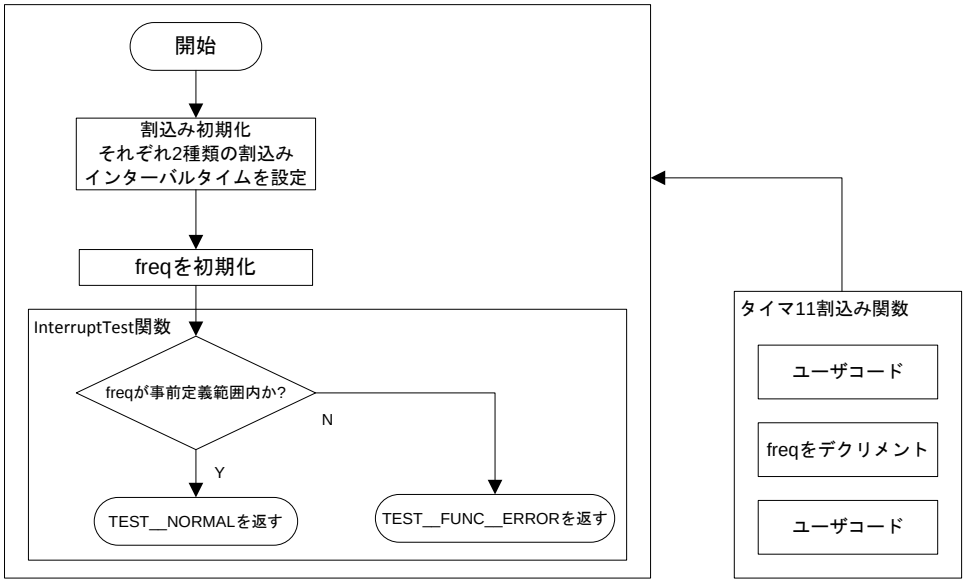
4.3.1 テストの説明

CLASS B の要件を満たすためには, 割込みに対して『間違った回数』のチェックを行う必要があります。このテストはシステムに高度に依存したタスクであり, そのため STL はラップアップハンドルを提供することしかできません。つまり, いくつかの特定の割込みが定義されている回数発生したこと (これより少なかったり多かったりしないか) をチェックします。

InterruptTest (割込みテスト関数) が特定の間隔 (たとえば, タイマまたは線周波数割込みでトリガーされる) で呼び出されることを前提としています。監視するすべての特定の割込みハンドラは, 専用のグローバル変数 (freq) をデクリメントする必要があります。InterruptTest() はその変数を事前に定義された上限および下限と比較して, 制限を超えている場合はエラーを返します。

割込みテストのフローチャートを次の図に示します。

図 4-3. 割込みテストのフローチャート



4.3.2 API の定義

名称	InterruptTest
パラメータ	なし
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

InterruptTest()は発生した割込み回数が事前定義範囲内であるかどうかをチェックします。

また, 割込みが時間内に処理されたかどうかも検証します。割込みテストのエラーを検出すると, プログラムはエラーを返します。

4.4 クロックテスト

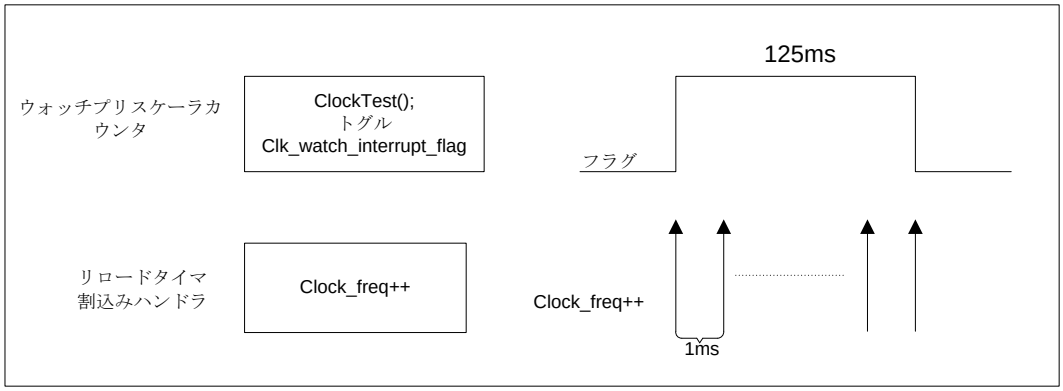
4.4.1 テストの説明

CLASS B の要件を満たすためには, CPU クロックに対して『間違った回数』のチェックを行う必要があります。これには, クロックテスト用の標準クロックとして 2 個の独立したクロックが必要です。まず, New-8FX MCU にはウォッチプリスケーラカウンタが集積されており, これは外部サブクロック (32.768 kHz の発振器) で駆動可能です。このテストはウォッチプリスケーラカウンタを標準クロックとして使用し, タイマ割込みでカウントされるタイムチックで検証することにより, CPU クロック周波数が許容範囲内にあるかどうかをテストします。タイマ割込みのソースクロックは CPU クロックと同一でなければなりません。CPU クロックがサブクロックで駆動されている場合はテストできません。32.768 kHz の発振器を正確と想定するからです。

クロックテストでは, 1 MHz, 8 MHz, 10 MHz, と 12.5 MHz の 4 種類のメイン CR 周波数がテスト可能です。要件に応じて必要なクロック周波数を選択し, テストを実行できます。システムクロックの初期化時に, clock_init 関数を使用してパラメータ (クロック周波数値) を送信し, このクロック周波数に従ってリロードタイマの 1 ms 内部時間に対応する設定を行います。

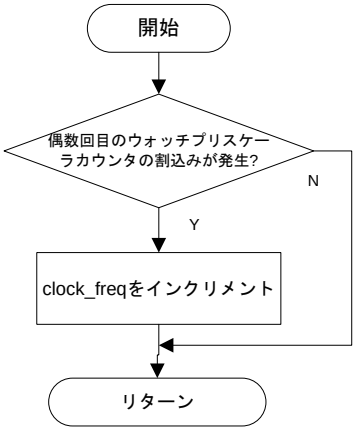
クロックテストのブロック図を次に示します。

図 4-4. クロックテストのブロック図



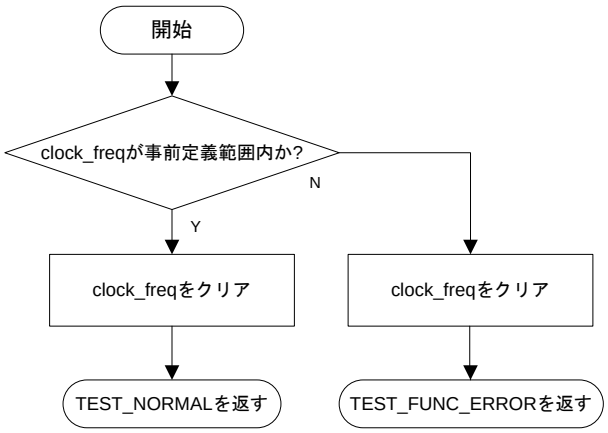
リロードタイマ割込みハンドラはグローバル変数『clock_freq』をカウントするために使用され, タイマのソースクロックは CPU クロックと同じでなければなりません。フローチャートを次の図に示します。

図 4-5. クロックカウンタのフローチャート



クロックテストのフローチャートを次の図に示します。

図 4-6. クロックテストのフローチャート



4.4.2 API の定義

名称	ClockTest
パラメータ	なし
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

クロックテストの標準クロックとしての外部サブクロック (32.768 kHz) で駆動されるウォッチカウンタ割込みハンドラ内で ClockTest() が呼び出されます。発生した割り込みの数 (タイマ割込みの駆動クロックは CPU クロックと同じでなければなりません) が事前定義範囲内かどうかをチェックします。クロック周波数が事前定義範囲内であるかどうかを検証し, かつ割込みが時間内に処理されたかも検証します。クロックテストのエラーが検出されるとプログラムは無限ループに入ります。

名称	Clk_Test_Init
パラメータ	なし
リターン	なし

説明:

Clk_Test_Init() はクロックテストに対応する変数およびフラグ (上限/下限周波数値, 等) の設定を行います。この API はクロックテスト開始前のシステム初期化時に呼び出さなければなりません。変数の値 (上限/下限周波数値) は実際の例に従って設定しなければなりません。たとえば, 1 ms タイマ割込みをモニタするためにウォッチカウンタの 125 ms のインターバルを使用する場合は, clock_freq_lower = 120 と clock_freq_upper = 130 の値をタイマクロック回数の限界値として設定します。この周波数値の標準値は 125 です。

4.5 不揮発性メモリ (ROM) のテスト

フラッシュのサイズは表 6-1:MB95330H シリーズ製品リスト表 6-1, 表 6-1 に示すように, さまざまな製品に応じて異なります。

CRC (巡回冗長検査) はエラー検出システムです。CRC コードは入力データ文字列を高次多項式と仮定して, それを定義済みの生成多項式で除算した後の剰余です。通常, データ文字列は送信時に CRC コードがサフィックスとして付けられ, 受信したデータは上と同様に生成多項式で除算されます。受信データが割り切れれば正しく受信できたと判断します。このモジュールでは, 生成多項式はこのモード用の数値に固定されています。

- CCITT CRC16 生成多項式: 0x1021

4.5.1 テストの説明

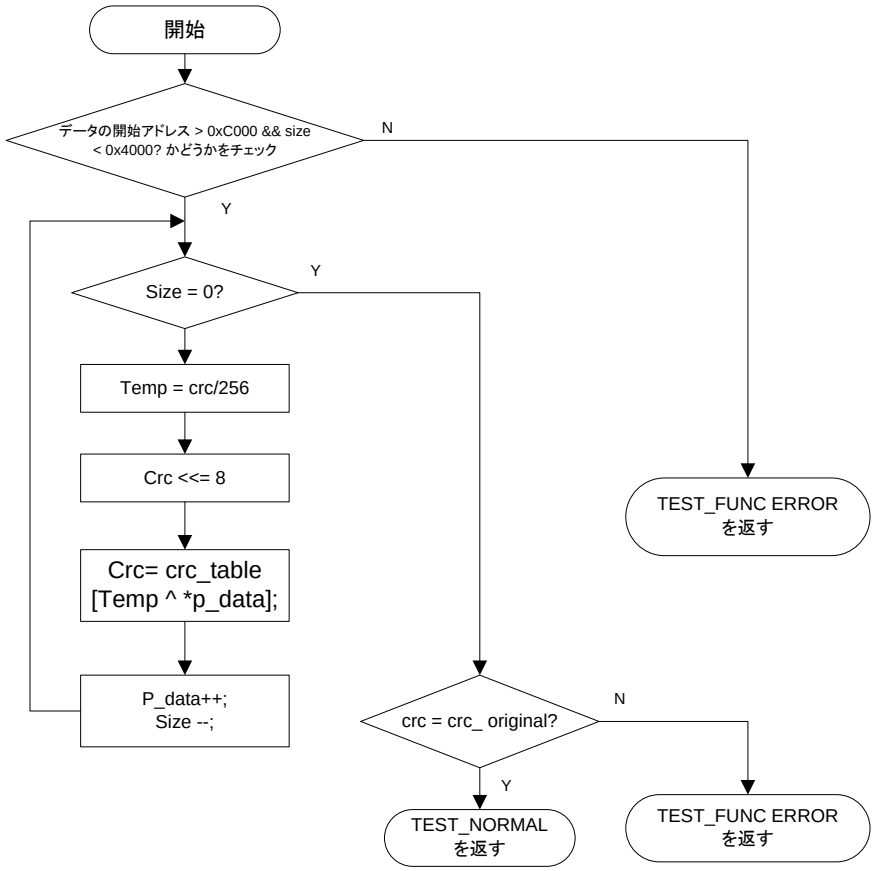
CLASS B の要件を満たすためには, フラッシュテストは『単一ビット不良』のチェックを行う必要があります。このテストは CRC16 テストとして実行できます。

このテストはスタートアップ手続き時に実行してコード領域全体をテストできます。

サンプルソースコードは、期待値の CRC コードを API を使用して生成していますが、期待値の CRC コードは予め計算された即値を使用することを推奨します。これには、期待値の CRC コードを保存した領域を、CRC チェックを行うコード領域から外すことが必要です。

不揮発性メモリテストのフローチャートを次の図に示します。

図 4-7. 不揮発性メモリテストのフローチャート



ソフトウェアによる CRC16 生成コードと CRC16 の表を次の図に示します。

図 4-8. ソフトウェアによる CRC16 生成のソースコード

```
unsigned int MakeCRCCode(unsigned char *p_data, unsigned int size)
{
          unsigned char temp;
          unsigned int crc = 0xFFFF;
          while(size-- != 0)
          {      temp = ((unsigned char)(crc/256))/16;
                 crc <<= 4;
                 crc ^= crc_table[temp ^ (*p_data/16)];
                 temp = ((unsigned char)(crc/256))/16;
                 crc <<= 4;
                 crc ^= crc_table[temp^(*p_data & 0x0F)];
                 p_data ++;
          }
          return crc;
}
```

図 4-9. CRC 表

```
const unsigned int crc_table[16]={
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50A5, 0x60C6, 0x70E7,
    0x8108, 0x9129, 0xA14A, 0xB16B, 0xC18C, 0xD1AD, 0xE1CE, 0xF1EF,
};
```

4.5.2 API の定義

名称	MakeCRCCode
パラメータ	p_data: テストデータのアドレス size: データサイズ
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

MakeCRCCode() はソフトウェア CRC 演算により CRC16 生成を行います。CRC テーブル参照法を使用します。これはオンチップ CRC をソフトウェア的に実装する際の参照 CRC 法となります。

名称	CRCTest
パラメータ	p_data: テストデータのアドレス size: データサイズ crc_original: 期待される CRC コード
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

CRCTest()は, 内部ソフトウェア CRC ジェネレータによって発生された CRC コードが期待される CRC コードと同一かどうかをチェックします。書き込みまたは送信後も冗長データが同じかどうかをチェックします。もし CRC コードエラーが発生すると, プログラムはエラーを返します。

4.6 揮発性メモリ (RAM) テスト

4.6.1 テストの説明

CLASS B の要件を満たすためには, RAM テストは『DC 不良』のチェックを行う必要があります。この RAM テストの実行には単純なチェッカーボード法を使用します。

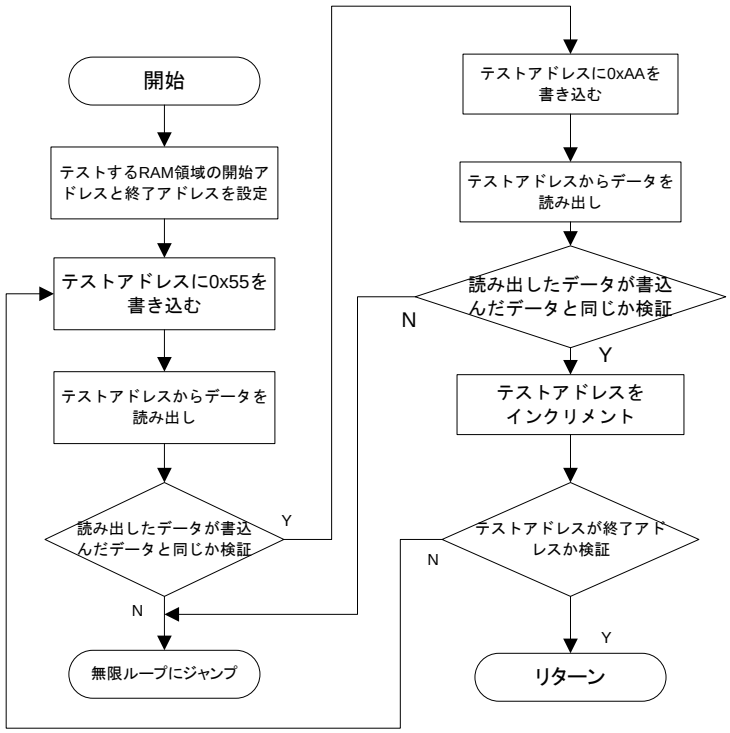
このテストはスタートアップ手続き時に実行して, RAM 領域全体をテストできます。この試験終了時にはデータが破壊されるので注意が必要です。

このテストはすべての RAM 領域が対象なので, このテストでは変数を使用しないことが推奨されています。そのため, レジスタテストの実装にはアセンブリ言語が使用されています。

またこのテストは非常に重要なので, RAM テストでエラーが検出されるとプログラムが無限ループに入るように設計されています。

揮発性メモリテストのフローチャートを次の図に示します。

図 4-10. 揮発性メモリテストのフローチャート



4.6.2 API の定義

名称	Ram_test
パラメータ	なし
リターン	なし

説明:

Ram_test ()は指定した RAM 全域を破壊的にテストします。

4.6.3 define 定義

名称	RAM_TEST_START_ADDRESS
値	RAM テストのスタートアドレス
備考	設定したアドレスから、RAM テストを開始します。値はユーザが設定可能です。

名称	RAM_TEST_END_ADDRESS
値	RAM テストのエンドアドレス
備考	設定したアドレスまで、RAM テストを実行します。テストは、本 define で設定したアドレスを含みます。値はユーザが設定可能です。

4.7 IO テスト

4.7.1 テストの説明

CLASS B の要件を満たすためには, GPIO に対して『機能エラー』のチェックを行う必要があります。したがって, 入力, 出力機能の双方に対して機能テストが行われます。

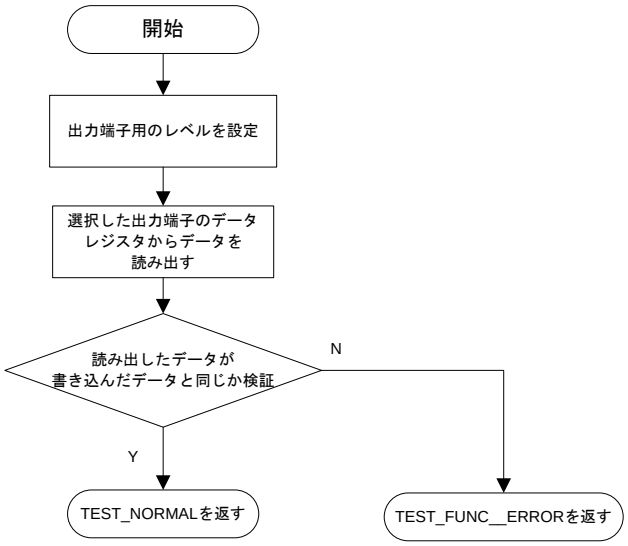
IO 出力テストのフローチャートを次の図に示します。

ただし、GPIO には下記の特徴があります。

- 出力モードの際 PDR レジスタをリードすると、PDR レジスタの値がリードされる
- 入力モードの際 PDR レジスタをリードすると、端子の値がリードされる

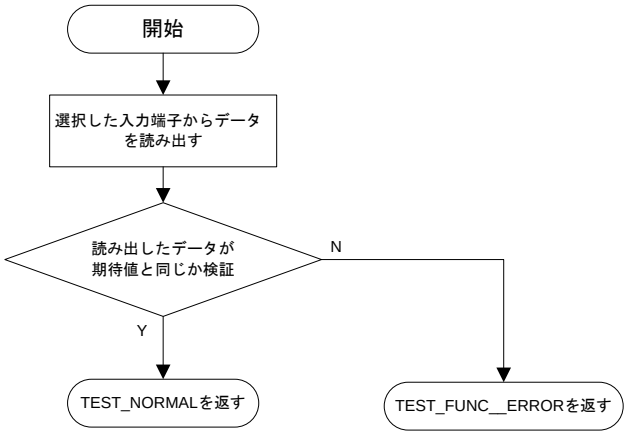
つまり、出力モードの際は PDR レジスタの値と端子の値が一致しない可能性があります。

図 4-11. 出力テストのフローチャート



IO 入力テストのフローチャートを次の図に示します。

図 4-12. IO 入力テストのフローチャート



4.7.2 API の定義

名称	IO_Input_Test
パラメータ	port : ポートアドレス (PDR レジスタアドレスに対応する) bit : ビット番号 pull : プルレジスタアドレス value : 期待される端子レベル
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

IO_Input_Test () は, 選択した IO 入力値が期待値と同じかどうかをチェックします。ユーザコードの初期化前に実行しなければなりません。

名称	IO_Output_Test
パラメータ	port : ポートアドレス (PDR レジスタアドレスに対応する) bit : ビット番号 value : 出力レベル
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

IO_Output_Test () は出力値が正しいかどうかチェックします。ユーザコードの初期化前に実行しなければなりません。

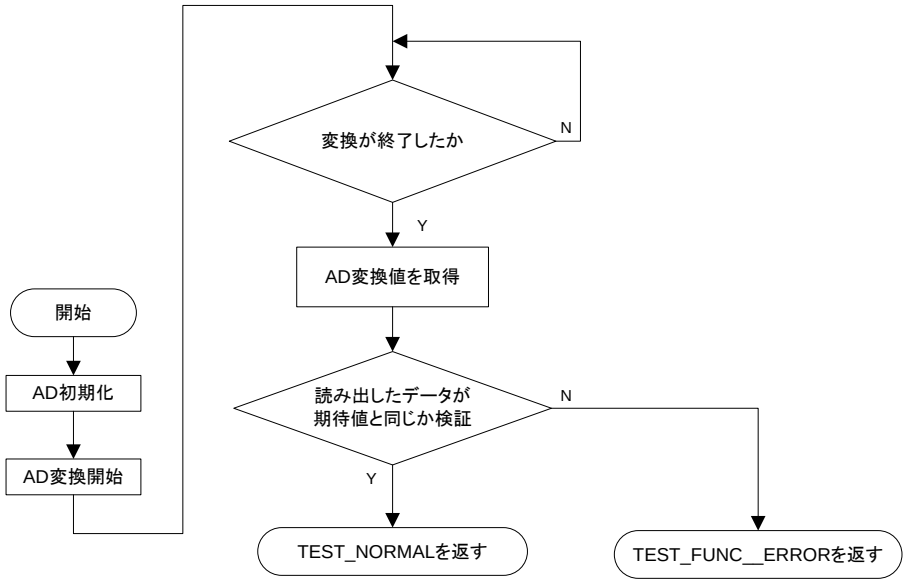
4.8 A/D テスト

4.8.1 テストの説明

CLASS B の要件を満たすためには, A/D に対して『機能エラー』のチェックを行う必要があります。このテストは, 選択された A/D チャネルから A/D 信号を取得し, A/D 変換された値が事前定義範囲内であるかをチェックします。

A/D テストのフローチャートを次の図に示します。

図 4-13. A/D テストのフローチャート



4.8.2 API の定義

名称	AD_test
パラメータ	ad_type: 8/10 ビット精度 pin_num: A/D チャネル番号 value_low: 期待される下限値 value_upper: 期待される上限値
リターン	0: TEST_NORMAL 1: TEST_FUNC_ERROR

説明:

AD_test() は, 選択した A/D チャネルから A/D 信号を取得し, A/D 変換された値が事前定義範囲内であるかをチェックします。その範囲内に収まらなければ, プログラムは無限ループに入ります。

5 STL の使用例

このプロジェクトは, IEC60730 STL を実際のシステムに統合する方法を示しています。

5.1 プロジェクトの構造

CLASS B の STL ルーチンは 2 つのメインプロセスに分割されます。スタートアップテストと定期テストです。定期テストは適用されるセットアップブロックにより初期化されなければなりません。

5.1.1 スタートアップテスト

PC, レジスタ, RAM テストはすべて破壊的なテストであるため, リセットハンドラで呼び出されなければなりません。

ROM, A/D, IO は, プログラムがメインファンクションにジャンプした後, システムクロックの初期化後にテストできます。

5.1.2 定期テストの初期化

割り込みおよびクロックテストはテスト開始前に初期化されなければなりません。

- 割り込みテスト初期化

8/16timer00 割込みは 8/16timer11 をモニタするように設計されています。初期化設定パラメータの例を次の表に示します。

事前定義範囲は、使用環境に応じて適切な値を設定してください。設定箇所は、Initialization.c ファイル IntTestInit 関数内の freq_lower / freq_upper です。

割込み名称	ハンドラ名	割込み間隔	8/16 ビットタイマ 11 の割込み間隔	標準周波	事前定義範囲
8/16 ビット複合タイマ ch. 0 (下側)	timer00_interrupt0	100 us	1 ms	10	[8,12]

- クロックテストの初期化

初期化設定パラメータの例を次の表に示します。リロードタイマの割込み間隔は 1 ms, ウォッチプリスケーラタイマの割込み間隔は 125 ms になっているので, リロードタイマの標準周波数は 125 であり事前定義範囲は 120 から 130 の間に設定されています。

事前定義範囲は、使用環境に応じて適切な値を設定してください。設定箇所は、CLOCK_TEST.c ファイル Clk_Test_Init 関数内の clock_freq_lower/clock_freq_upper です。

カウンタ名	ハンドラ名	インターバルタイマ	ウォッチプリスケーラのインターバルタイム	標準周波数	事前定義範囲
16 ビットリロードタイマ ch. 1	ReloadTimer	1 ms	125 ms	125	[120,130]

5.1.3 定期テスト

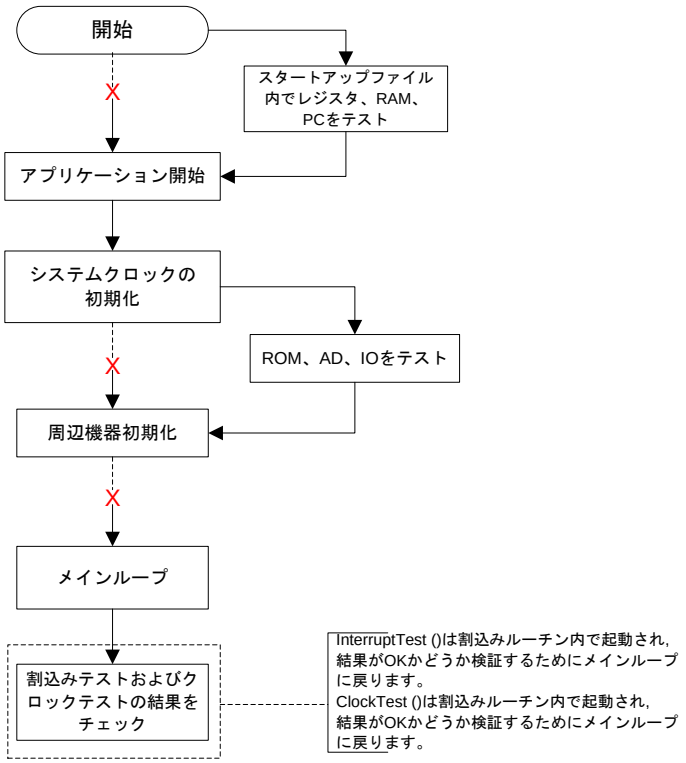
割込みおよびクロックテストはコード実行中に定期的に実行出来ます。

InterruptTest () を 8/16timer11 割込みに統合し，各 8/16timer00 割込みの割込み回数をカウントし，メインループ中の InterruptTest () 関数から返される result_interrupt フラグをチェックします。

ClockTest() をウォッチプリスケーラ割込みに統合し，リロードタイマ割込みの割込み回数をカウントし，メインループ中の ClockTest() 関数から返される result_clock フラグをチェックします。

図 5-1 に CLASS B ソフトウェアパッケージをアプリケーションソフトウェアに統合する上での基本原則を示します。

図 5-1.プロジェクトの構造



6 New 8FX シリーズ MCU

6.1 New 8FX シリーズ MCU について

本アプリケーションでは MB95330H シリーズと MB95630H シリーズが使用されています。MB95330H シリーズと MB95630H シリーズは, どちらも汎用のシングルチップマイクロコントローラのシリーズです。コンパクトな命令セットに加え, 本シリーズのマイクロコントローラはさまざまな周辺機器リソースを備えています。

表 6-1. MB95330H シリーズ製品リスト

製品	ROM	RAM	パッケージ
MB95F332H/K	8 K バイト	240 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19
MB95F333H/K	12 K バイト	496 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19
MB95F334H/K	20 K バイト	1008 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19

表 6-2. MB95630H シリーズ製品リスト

製品	ROM	RAM	パッケージ
MB95F632H/K	8 K バイト	256 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19
MB95F633H/K	12 K バイト	512 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19
MB95F634H/K	20 K バイト	1024 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19
MB95F634H/K	36 K バイト	1024 バイト	FPT-32P-M30 DIP-32P-M06 LCC-32P-M19

6.2 New 8FX IEC60730 STL デモプロジェクトについて

IEC60730 STL デモプロジェクトには 2 種類のプロジェクトが含まれています。1 つは MB95330H シリーズ用で，もう 1 つは MB95630H シリーズ用です。どちらも New 8FXIEC60730 セルフテストライブラリの使用法を説明するためのサンプルプロジェクトです。

表 6-3.プロジェクトで使用したツール

	MB95330H シリーズ	MB95630H シリーズ
MCU	MB95F334K	MB95F636K
評価ボード	MB2146-442-E	FMCDC-8FX-PGMA-07032080
SOFTUNE	F²MC-8L/8FX ファミリ SOFTUNE ワークベンチ V30L32R05	F²MC-8L/8FX ファミリ SOFTUNE ワークベンチ V30L33
BGMA	MB2146-08-E	MB2146-07-E

使用 ROM/RAM サイズ:

- ROM: 2K
- RAM:
  - MB95F334K: 1008 バイト
  - MB95F636K: 1024 バイト

本アプリケーションで使用する MCU の周辺機能:

- 8/16 ビット複合タイマ
- 16 ビットリロードタイマ
- ウォッチプレスケーラ
- I/O ポート
- 8/10 ビット A/D コンバータ

改訂履歴

文書名: AN204364 - New 8FX ファミリ MB95330H/630H シリーズ IEC 60730 CLASS B セルフテストライブラリ

文書番号: 002-04365

版	ECN 番号	変更者	発行日	変更内容
**	-	MIQI	07/13/2012	新規作成 スパンションアプリケーションノート AN702-00006-1v0-J をドキュメントコード 002-04365 に登録しました。
*A	5697552	MIQI	04/17/2017	最新のテンプレートへ更新しました。

セールス, ソリューションおよび法律情報

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューション センター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、サイプレスのロケーション ページをご覧ください。

製品

ARM® Cortex® Microcontrollers	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmic
タッチ センシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス／RF	cypress.com/wireless

PSoC® ソリューション

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

サイプレス開発者コミュニティ

フォーラム | WICED IOT Forums | Projects | ビデオ | ブログ | トレーニング | Components

テクニカルサポート

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2012-2017. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社（以下、「Cypress」という。）に帰属する財産である。本書面（本書面に含まれ又は言及されているあらゆるソフトウェア又はファームウェア（以下、「本ソフトウェア」という。）を含む）は、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき、Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、また、本段落で特に記載されているものを除き、Cypress の特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾していない。本ソフトウェアにライセンス契約書が伴っておらず、かつ、あなたが Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意をしていない場合、Cypress は、あなたに対して、（1）本ソフトウェアの著作権に基づき、（a）ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためにのみ、組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに（b）Cypress のハードウェア製品ユニットに用いるためにのみ、（直接又は再販売者及び販売代理店を介して間接のいずれかで）エンドユーザーに対して、バイナリーコード形式で本ソフトウェアを外部に配布すること、並びに（2）本ソフトウェア（Cypress により提供され、修正がなされていないもの）に抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためにのみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス（サブライセンスの権利を除く）を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェアに関しても、明示又は黙示をとわず、いかなる保証（商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない）も行わない。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のあるいかなる製品又は回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報（あらゆるサンプルデザイン情報又はプログラムコードを含む）は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計し、プログラムし、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分として用いるため、又はシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせることになるその他の使用（以下、「本目的外使用」という。）のためには、設計、意図又は承認されていない。重要な構成部分とは、装置又はシステムのその構成部分の不具合が、その装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できる、機器又はシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部をとわず一切の責任を負わず、かつ、あなたは Cypress をそれら一切から免除するものとし、本書により免除する。あなたは、Cypress 製品の本目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任（人身傷害又は死亡に基づく請求を含む）から Cypress を免責補償する。

Cypress、Cypress のロゴ、Spansion、Spansion のロゴ及びこれらの組み合わせ、WICED、PSoC、CapsSense、EZ-USB、F-RAM、及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress の商標のより完全なリストは、cypress.com を参照のこと。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。