iMOTION™デバイスプログラミング

モーションコントロールエンジン, パラメータおよびスクリプトのプログラミング

本書について

適用範囲と目的

本書では、第 2 世代 iMOTION™デバイス (iMOTION™) のファームウェアのプログラミングと更新について説明します。このアプリケーションノートでは、パラメータの設定とスクリプトの処理同様、モーションコントロールエンジン (MCE) それ自体のプログラミングを示します。

統合されたモーションコントロールエンジンを備えたすべての新しい iMOTION™製品 (MCE) は MCE を新しいリビジョンに更新するのと同様に、例えば、開発段階で量産製品のプログラミング用のセキュアなブートローダーメカニズムを使用します。

MCE リビジョン 1.3 までの場合、個々の iMOTION™デバイスのソフトウェア イメージは、インフィニオンテクノロジーズの Web サイト (www.infineon.com/imotion-software) で入手できます。

MCE リリース 5.x 以降では、イメージは SD pack の一部として配布され、iMOTION™ Solution Designer (iSD) 内から管理されます。iMOTION™ Solution Designer は、利用可能なアップデートについてユーザーに通知し、それぞれのダウンロードを提供します。

特定の iMOTION™デバイスのファームウェアイメージは暗号化されたファイルとして提供され、それぞれのチップタイプおよび復号化キーの組合せでのみ使用できます。したがって、これらのソフトウェアイメージは、一致するキーを持つ特定のデバイスタイプにのみインストールできます。また MCE とそれぞれのデバイスとの互換性を保証します。

システム、モーター、および力率補正 (PFC) のパラメータ設定は、暗号化されていない形式で管理されます。

同じことが、統合スクリプトエンジンのスクリプトにも当てはまります。

Note: *IMC300 シリーズのような iMOTION™製品に Arm® Cortex® コアベースのマイクロコントローラーが含まれている場合、Arm® で指定されている標準化された SWD インターフェースを使用します。デバッグコントローラとレジスタ構造は、それぞれの製品のリファレンスマニュアルに記載されます。ドキュメントの最後に参考資料を記載します。*

対象者

本書は、主にエンドオブラインプログラミング機器 (ギャングプログラマー) のプロバイダーを対象とします。

iMOTION™デバイスのプログラミングには、物理インターフェースとして UART のみが必要であるため、このドキュメントは他のユーザーがプログラミングフローを実装するためにも使用できます。

参考資料

製品のドキュメントとソフトウェアは、http://www.infineon.com/iMOTION からダウンロードできます。このドキュメントの最後にある参考資料も参照してください。

目次

本書について 1
目次 2
1 概要 4
1.1 デバイスモードの説明 5
1.2 UART 設定 7
1.3 MCE revision 1.x 9
1.3.1 ローダーファイル形式の説明 9
1.3.2 パラメータ設定ローダーファイル形式 10
1.3.3 スクリプトローダーファイル形式 11
1.3.4 結合されたローダーファイル 12
1.4 MCE revision 5.x 13
1.4.1 iMOTION™ Solution Designer ファイル処理 13
1.4.2 パラメータセットとスクリプトローダーファイル 14
2 セキュアブートローダー (SBSL) – MCE プログラミング 16
2.1 セキュアローダーコマンド設定 16
2.2 サポートされているコマンド 19
2.3 一般的なコマンドステータスの応答 19
2.4 FLASH_CHIP_RESET 20
2.5 FLASH_GET_SBSL_STATUS 20
2.6 FLASH_LOAD_DATA 22
2.7 FLASH_LOAD_CHECK_SIGNATURE 24
3 設定モード - パラメータおよびスクリプトのプログラミング (MCE 1.x) 25
3.1 サポートされているコマンド 25
3.2 CONNECT 26
3.3 CHIP_RESET 26
3.4 GET_STATUS 27
3.5 GET_PARAMETER_SET_NAME 29
3.6 GET_PARAMETER_SET_VALUE 30
3.7 GET_CHECKSUMS 31
3.8 CHANGE_BOOT_MODE 32
3.9 DOWNLOAD_PARAMETER (スクリプトにも使用されます) 33
3.10 CHECK_PARAMETER (スクリプトにも使用されます) 34
3.11 CLEAR_PARAMETER 35
4 設定モード - パラメータおよびスクリプトのプログラミング (MCE 5.x) 36
4.1 サポートされているコマンド 36
4.2 CONNECT 36
4.3 CHIP_RESET 37
4.4 GET_STATUS 38
4.5 GET_PAGE_DATA 39
4.6 GET_CHECKSUMS 40
4.7 CHANGE_BOOT_MODE 40
4.8 DOWNLOAD_DATA 42
4.9 CHECK_DATA 43
4.10 CLEAR_DATA 44
5 安全故障モード 45
5.1 サポートされているコマンド 45
5.2 CHIP_RESET 45

5.3 GET_STATUS 46
5.4 CHANGE_BOOT_MODE 47
6 アプリケーションモード- MCE 1.x コマンド設定 48
6.1 サポートされているコマンド 48
6.2 CHECK_COMMUNICATION 49
6.3 CHANGE_BOOT_MODE 50
7 アプリケーションモード- MCE 5.x コマンド設定 51
7.1 サポートされるコマンド 51
7.2 CHANGE_BOOT_MODE 52
8 キャッチ・アット・スタートアップ方式によるブートモードの変更 53
9 事例紹介 54
9.1 デバイスの SBSL ステータスの読み出し 54
9.2 製造ラインの最後でのプログラミング 55
9.3 MCE の更新の実行 56
9.4 ユーザーモードに設定された UART からの MCE の更新 56
9.5 パラメータまたはスクリプトの更新の実行 56
参考資料 57
改訂履歴 58
免責事項 59

1 概要

このドキュメントでは、モーションコントロールエンジン (MCE リビジョン 1.0 以降) に基づく iMOTION™ 製品のプログラミングと更新について説明します。

iMOTION™製品はプログラムされたソフトウェアなしで出荷されるため、生産ラインの最後に次の手順を実行する必要があります。

1. モーションコントロールエンジン (MCE) のプログラミング
2. システム、モーター、力率補正のパラメータ設定のプログラミング (PFC はオプション)
3. カスタマースクリプトのプログラミング (オプション)

すべての通信とプログラミングは、次の章で説明するプロトコル定義を使用して UART インターフェースを介して行われます。

通信フローは、このアプリケーションノートでは 16 進表記で示される送信バイトと受信バイトに基づきます。

適切なプロトコルを選択するには、1.1 デバイスモードの説明で説明されるように、デバイスの現在の状態を識別することが重要です。デバイスの状態に応じて、次のプログラミング手順を実行できます。

- インフィニオンが提供する暗号化されたファームウェアイメージを使用した、セキュアブートストラップローダー (SBSL) を介した MCE のプログラミング。このセキュアローダーはデバイスの一部であり、変更または消去できません。SBSL は、短いプログラミング時間を可能にするために高いボーレートの使用をサポートします。
- MCE の再プログラミングは、例えば、新しいファームウェアリリースにアップグレードする場合に、セキュアローダーモードに再度入ることによって実行されます。セキュアローダーを介して MCE を再プログラミングすると、すべてのパラメータ設定とスクリプトを含む完全な内部フラッシュメモリが消去されます。
- パラメータとスクリプトのプログラミングは、暗号化されていないデータを使用して行われます。パラメータローダーとスクリプトローダーは MCE ファームウェアの一部であるため、最初に MCE ファームウェアをプログラムする必要があります。

ファームウェア イメージは、デバイス固有のローダー ファイル (*.ldf) の形式で配布されます。形式については、1.3.1 ローダー ファイル形式の説明の章で説明されています。

- MCE リビジョン 1.3 までの場合、個々の iMOTION™デバイスのソフトウェア イメージは、インフィニオンテクノロジーズの Web サイト (www.infineon.com/imotion-software) で入手できます。
- MCE リリース 5.x 以降では、イメージは SD pack の一部として配布され、iMOTION™ Solution Designer (iSD) 内から管理されます。iMOTION™ Solution Designer は、利用可能なアップデートについてユーザーに通知し、それぞれのダウンロードを提供します。

この概要の章では、iMOTION™デバイスのプログラミングの概要を説明し、基本的なハードウェアインターフェース設定も提供します。

次の章では、さまざまな製品モードで使用される通信プロトコルの定義について説明します。

また、初期 (end-of-line) プログラミング、およびパラメータ設定またはスクリプトの更新の例も示します。

1.1 デバイスモードの説明

iMOTION™デバイスは、サポートされるボーレートとプロトコルも定義する 4 つの異なる状態にできます。状態 1 と 2 の場合、デバイスは UART での接続要求に応答してその状態を自動的に報告し、状態はコマンド GET_STATUS を使用して要求することもできます。報告された値を以下および Figure 1 に示します。
コマンドを介した開始、あるいは Figure 1 に示す自動的な実行で、モード間を遷移できます。

1. セキュアブートストラップローダーモード 0x5D

モーションコントロールエンジン (MCE) に基づくすべてのデバイスは、デバイスの起動時にアクティブになる組込みのセキュアブートストラップローダー (SBSL) とともに工場から出荷されます。SBSL は、暗号化された MCE ソフトウェアイメージのデバイスフラッシュへの安全な転送をサポートし、ソフトウェアとデバイスタイプの互換性を保証します。MCE ファームウェアがプログラムされると、デバイスは自動的に設定モードに変わります。

2. 設定モード 0xCD

このモードは、MCE ファームウェアを (再) プログラミングした後、または応答コマンドで要求された場合に開始されます。設定モードは、パラメータやスクリプトのプログラミングに使用されます。

有効なパラメータ設定がプログラムされた後、デバイスはアプリケーションモードに移行できます。

3. アプリケーションモード 0xAD

アプリケーションモードは、モーターとオプションの PFC を実行している、デバイスの通常の動作状態です。有効なパラメータ設定がプログラムされている場合、このモードは電源投入後に開始されます。

CHANGE_BOOT_MODE コマンドを使用して、新しいパラメータまたはスクリプトをプログラミングするための設定モードに入れます。MCE 自体を更新する必要がある場合は、SBSL モードに入れます。

4. フェイルセーフモード 0xAF

クラス B 障害の結果として、デバイスは「フェイルセーフモード」に入ります。このモードでは、ユーザーは GET_STATUS コマンドと CHIP_RESET コマンドを使用できます。ファームウェアを再プログラミングするには、コマンド CHANGE_BOOT_MODE を利用して、SBSL モードにする必要があります。

あるモードから別のモードに変更すると、次のような効果があります。

- SBSL モードに変更すると、デバイスのフラッシュ全体、つまり MCE 自体、および以前の状態に関係なくパラメータ設定とスクリプトが自動的に消去されます。
- 設定モードに変更しても、既存のパラメータ設定とスクリプトは消去されません。代わりに、パラメータ設定を消去して選択的に書き込めます。スクリプトは上書きされます。つまり、新しいスクリプトプログラミングが行われると、スクリプトは自動的に消去されます。
- 障害状態に移行しても、フラッシュの内容は変更されません。

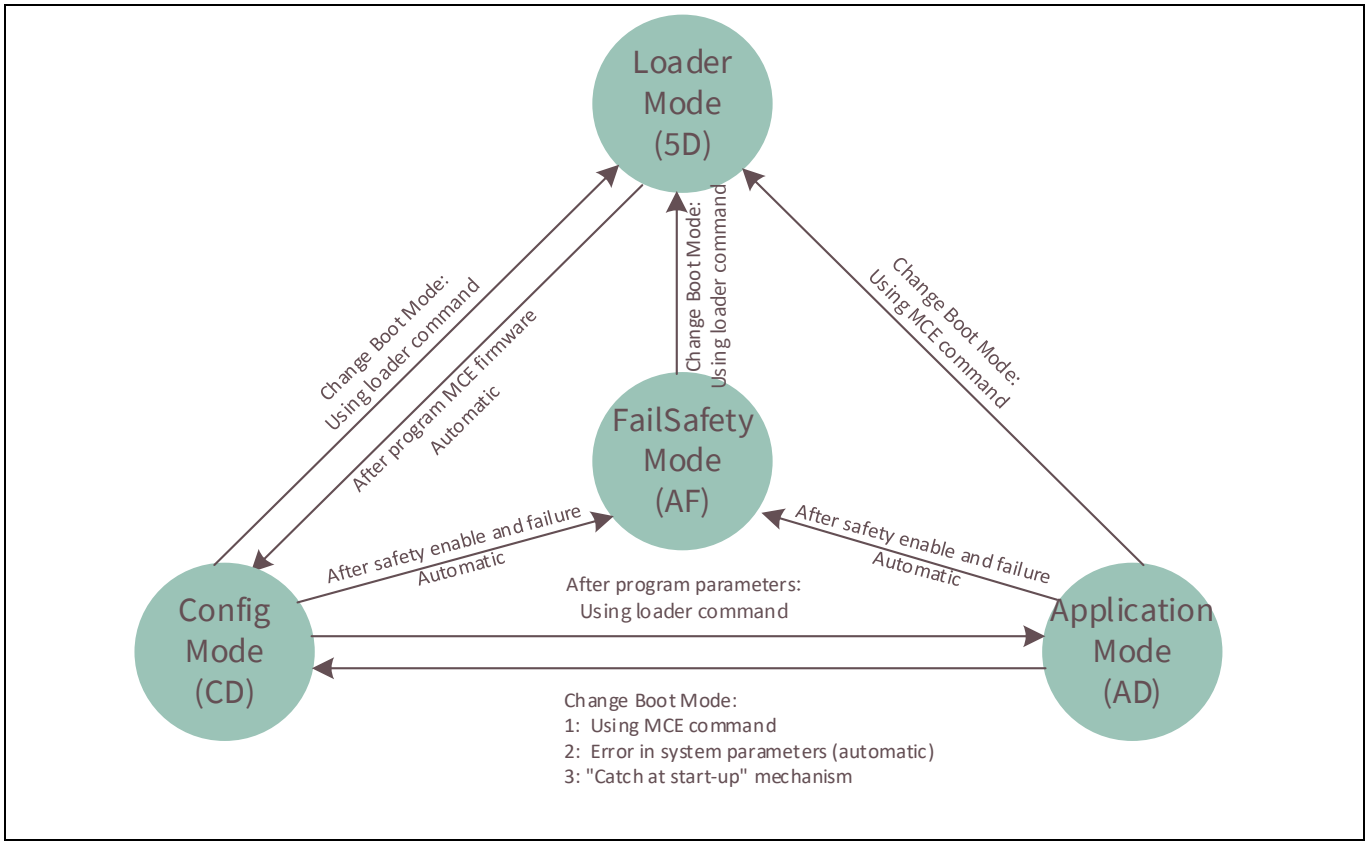


Figure 1 さまざまなデバイスモードのフローチャート

1.2 UART 設定

複数の UART を提供するデバイスの場合、それぞれのデータシートで「デバイスプログラミング」とマークされるポートは、MCE と、パラメータおよびスクリプトプログラミングの両方に使用する必要があります。

UART1 は、速度を制御し、ドライブのステータスを報告するコマンドを送受信するなど、ユーザー通信専用です。UART インターフェースの詳細については、iMOTION™製品の AN2020-07 インターフェースを参照してください。[2]

iMOTION™デバイスは、プログラミングポートの次の設定を使用します。

- 8 データビット
- 1 ストップビット
- パリティなし
- LSB ファースト

自動ボーレート検出

iMOTION™デバイスは、300～115,200 Baud の範囲で自動ボーレート検出を実装します。

ボーレート検出は、CONNECT コマンド「0x00 0x6C」を送信するダウンロードツールによってアクティブ化されます。ボーレートが正しく検出されると、iMOTION™デバイスは 1.1 デバイスモードの説明に記載される現在のモードで応答します。新しいデバイスの場合、これは SBSL モード「0x5D」になります。デバイスの状態以外の応答が返された場合、デバイスはダウンロードツールで使用されるボーレートを認識できません。

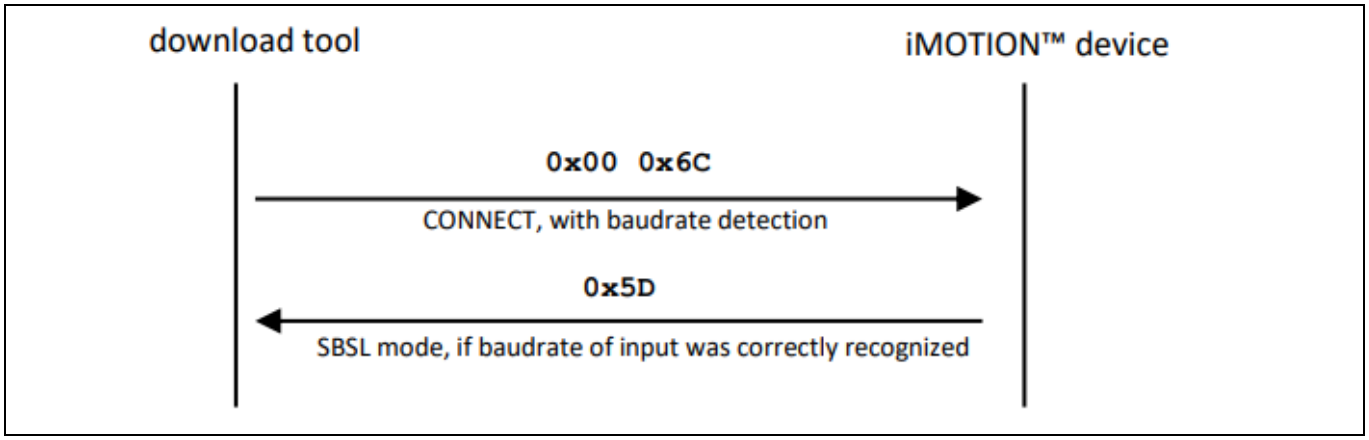


Figure 2 ボーレート検出を備えたプロトコルフロー

拡張ボーレートモード

製造における高速なエンドオブラインプログラミングを可能にするために、セキュアブートローダー (SBSL) はさらに拡張ボーレートモードをサポートします。このモードでは、自動ボーレート検出も実装されており、最大 5.994 MHz のボーレートを使用できます。拡張ボーレートモードに入るフローは以下のとおりであり、Figure 3 拡張ボーレートモードでのプロトコルフローに示されます。

拡張ボーレートモードは、ダウンロードツールによって「0x00 0x93」CONNECT コマンドを送信することによって要求されます。

デバイスは、同じボーレートで「0xA2」と「PDIV」の値で応答します。

概要

ダウンロードツールは、要求された「最終ボーレート」を定義する「STEP」値の計算に「PDIV」値を使用し、STEP 値をデバイスに送信します。(STEP は最初に MSB で送信されます)

iMOTION™デバイスは、初期ボーレートで「0xF0」を確認し、ターゲットボーレートに切り替えます。ターゲットボーレートに変更した後、ダウンロードツールは確認応答「0xF0」も送信します。

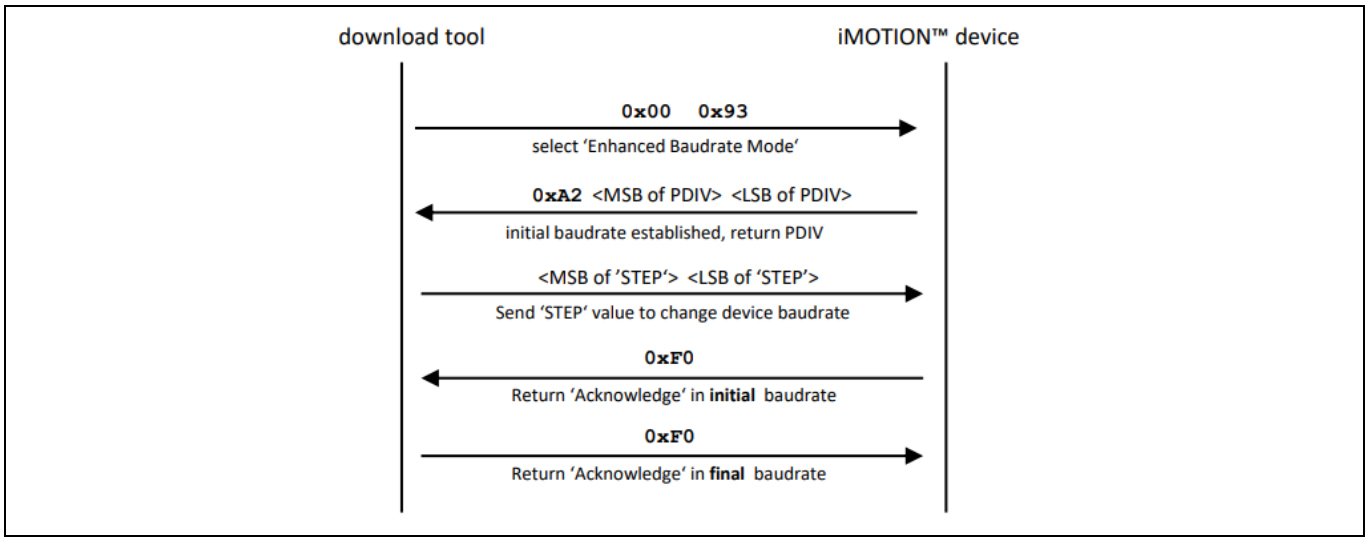


Figure 3　拡張ボーレートモードでのプロトコルフロー

STEP 値の計算

STEP 値は、次の式に従って最終ボーレートを調整するために使用されます。

ステップ= 1024 × (ターゲットボーレート/初期ボーレート) / (PDIV + 1)

例

ボーレートを最初の 115,200 から 1Mbaud に変更します。

- 初期ボーレート= 115,200 Baud
- 目標ボーレート= 1,000,000 Baud
- PDIV = 0x00 0x34 = 0x0034 = 52

STEP = 1024 × (1,000,000 / 115,200) / (52+1) = 167.7 ~ 168 = 0x00A8 = 0x00 0xA8

Note:　強化されたボーレートの使用は、ギャングプログラマーのような専用ツールにのみ推奨されます。PC 用の一般的な USB-UART ('VCOM') インターフェースでは、これらの高いボーレートを確実に達成するのに問題がある可能性があります。

1.3 MCE revision 1.x

1.3.1 ローダーファイル形式の説明

インフィニオンは、Web サイトでファイル拡張子*.ldf のローダーファイルでモーションコントロールエンジンを提供します。

これらのファイルには、それぞれの暗号化されたファームウェアイメージが ASCII ファイルの形式で含まれており、個々のデータバイトは 16 進表現で示されます。

このローダーファイルは、*.zip ファイルにパッケージ化されて提供されます。パッケージには、実際のファームウェアに加えて、使用許諾契約, リリースノート, および基本的な使用法を説明する短い readme が含まれます。

例: Infineon-MCE_IMC101T-T038-Firmware-v01_03-EN.zip

- IMC101T-T038_A_V1.03.03.ldf -デバイス IMC101T-T038 用の MCE rev.1.3 ファームウェア
- IMC101T-T038_Parameter.txt -初期パラメータファイル (テキスト形式)
- IMC101T_V1.03.03_Default.map -初期マップファイル
- IMC101T_V1.03.03.irc -初期*.irc ファイル
- IMC101T-T038_V1.03.03_ReleaseNotes.pdf -MCE バージョンリリースノート
- iMOTION_License_Agreement.pdf -iMOTION™使用ライセンス契約
- Readme.txt -使用法の簡単な説明

Attention: 暗号化されたローダーファイルはタイプ固有です。間違ったファイルをデバイスにプログラムしようとすると、フラッシュダウンロードトライアルカウンター(FDTC) が減少します。FDTC がゼロに達すると、デバイスは使用できなくなります。 (2.5 を参照)

プログラミングには、ローダーファイルのみが必要です。ローダーファイルの簡略例を以下に示します。

Table 1 MCE ローダーファイルの例 (省略形)

`# DEVICE: IMC101T-T038`
`# RELEASE: A_V1.03.03`
`# Date: 24.07.2020`
`# LIP: 70A0BD00`
	ファイルヘッダー、
	'#'で始まる行はコメント

`A0 20 00 00 82 A0 00 00 01 5C 67 20 ... A1 9E`
`A0 20 00 00 82 98 B1 34 BC 47 C9 BD ... 5D E0`
`...`
`A0 20 00 00 82 FB D5 66 EA 4C BA C3 ... 4D 64`
	MCE イメージ
	各行は長さバイト (0x82 = 130 バイト) を含む FLASH_LOAD_DATA コマンドで始まり、ペイロード (MCE)が続きます。

`A0 20 00 00 12 D3 69 49 41 F6 10 76 ... BC D2`	通常、最後の行は短く、ここでは 0x12 –18 バイトです。
`A0 21 00 00 00`	フラッシュプログラミングを完了するために必要な FLASH_LOAD_CHECK_SIGNATURE コマンド

1.3.2 パラメータ設定ローダーファイル形式

ダウンロードに使用されるパラメータファイルのファイル拡張子は*.ldf で、MCE ローダーファイルと同じ形式でヘッダー情報が追加されます。 (人間が読めるテキスト表示のパラメータファイルの拡張子は *.txt です。)

パラメータは MCE に統合されたローダーを使用してプログラムされるため、コマンドは 3 設定モード - パラメータおよびスクリプトのプログラミングで指定されたとおりになります。

モーターごとに 1 つのパラメータブロックと PFC パラメータ用に 1 つのパラメータブロックがあるため、通常、プログラムするパラメータブロックは複数あります。

Table 2 パラメータローダーファイルの例 (省略形)

%:Parameters Data Section Begin %-------------------- % Page 00 - AppID 01 %--------------------	パラメータ設定ヘッダー、 '%'で始まる行はコメント
% Erase Parameter Set a0 22 00 01 00	パラメータ設定を消去するための CLEAR_PARAMETER コマンド
% Program Parameter Set a0 20 00 01 40 99 9b 00 00 01 ... 33 03 a0 20 00 01 40 00 00 00 00 33 ... 3f 00	パラメータデータイメージ。各行は、長さバイト (0x40 = 64 バイト) を含みます DOWNLOAD_PARAMETER コマンドで始まり、その後にペイロードが続きます
% Check Parameter Set a0 21 00 01 00	チェックサム計算に必要な CHECK_PARAMETER コマンド

1.3.3 スクリプトローダーファイル形式

スクリプトはバイトコードにコンパイルされ、デバイスにロードされ、統合されたスクリプトエンジンによって実行されます。ダウンロードに使用するファイルの拡張子は*.ldf で、ヘッダー情報が異なるパラメータローダーファイルと同じ形式を使用します。 (スクリプトソースファイルは拡張子*.mcs を使用します。)

パラメータと同様に、スクリプトは MCE に統合されたローダーを使用してプログラムされ、コマンドは 3 設定モード - パラメータおよびスクリプトのプログラミングで指定されたとおりです。

Table 3 スクリプトローダーファイルの例 (省略形)

%------------------------------------- % Script Compiler Version : V1.0.14 %------------------------------------- % Script Object File %------------------------------------- % SCRIPT_USER_VERSION : 001.000 % SIZE : 453 Bytes % Total Number of Lines : 107	スクリプトコードヘッダー、 '%'で始まる行はコメント バージョン、サイズ、合計行数の概要
% Task 0 – Number of Instructions : 20 % Task 1 - Number of Instructions : 13 %-------------------------------------	タスクごとの命令の数、 スクリプトタスク実行の分割に使用できます ([3]を参照)
MCEDesigner ID for Script Global Variables, Script variables are % Part of System group %------------------------------------- % User Variable Name Variable ID %------------------------------------- % sVar0_L (Low 16bit) 130 % sVar0_H (High 16bit) 131 %-------------------------------------	グローバル変数とその ID のリスト、 グローバル変数は、他のすべての MCE パラメータと並行して外部ツールで監視できます。
% Program Script a0 20 00 02 40 72 aa ff ff 00 … 88 13 a0 20 00 02 40 00 00 29 29 47 … 3e 4a …	スクリプトバイトコードイメージ。各行は、長さバイト (0x40 = 64 バイト) を含む DOWNLOAD_PARAMETER コマンドで始まり、その後にペイロードが続きます。
% Verify Script a0 21 00 02 00	チェックサム計算用の verify スクリプトコマンド

1.3.4 結合されたローダーファイル

使いやすくするために、上記の 3 つのデータセクション (MCE ファームウェア、パラメータおよびスクリプト) を 1 つのローダーファイル (*.ldf) に組み合わせられます。

ローダーファイルの 3 つのセクションは、それぞれのセクションのマーカーで区切られます。

Table 4 結合されたローダーファイルの例 (省略形)

%:Combined file 16-BITS CRC result: 0xA97F **%:Firmware Data Section Begin** # DEVICE: IMC101T-T038 ... A0 20 00 00 12 00 80 AE 9A 6B 55 ... B5 24 A0 21 00 00 00 **%:Firmware Data Section End**	説明とチェックサムを含むファイルヘッダー MCE ファームウェアセクションの始まり MCE セクションの終わり
%:Parameters Data Section Begin %---------- % Page 00 - AppID 01 %---------- … % Check Parameter Set a0 21 0f 01 00 **%:Parameters Data Section End**	パラメータセクションの始まり (パラメータセクションには複数のパラメータ設定を含められることに注意してください) パラメータセクションの終わり
%:Script Data Section Begin %---------- % Script Compiler Version : V1.0.14 %---------- ... % Verify Script a0 21 00 02 00 **%:Script Data Section End**	スクリプトセクションの始まり スクリプトセクションの終わり

1.4 MCE revision 5.x

1.4.1 iMOTION™ Solution Designer ファイル処理

モーションコントロールエンジン (MCE) リビジョン 5 と PC 用ツール iMOTION™ Solution Designer (iSD) のリリースにより、ファイルの処理は大幅に改善されました。

iSD には、SD pack で動作するパッケージマネージャーが含まれています。新しいリリースの可用性についてユーザーに通知し、それらをインストールします。SD pack には、サポートされているすべての iMOTION™製品のファームウェア画像が含まれています。

iSD は、プロジェクトの作成から最終チューニングとスクリプト生成までの完全なワークフローをサポートしています。詳細については、iMOTION™ Web ページを参照してください。[1] [4]

プロジェクトの最終化の後、ローダーファイルは現在のプロジェクトの"generated"サブフォルダに保存されます。実際の MCE ファームウェアは、選択したデバイスとして名前が付けられ、ファームウェアリビジョンとファイル‘out.ldf’には、生成されたパラメータセットとオプションのスクリプトが含まれています。

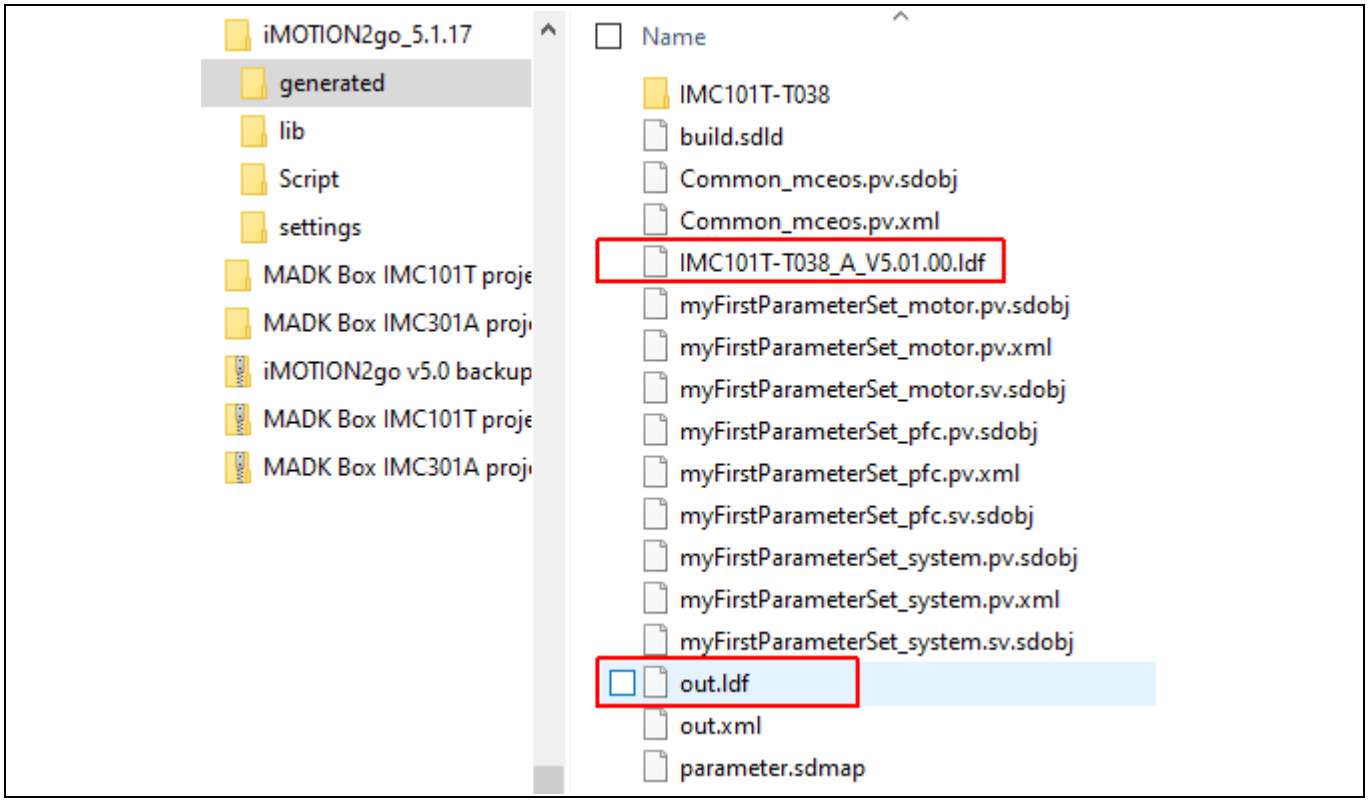


Figure 4 iMOTION™ Solution Designer のファームウェアおよびパラメータファイル

ファイル‘out.ldf’には、設定されたすべてのパラメータセット、選択したスクリプト、および必要なすべてのプログラミングコマンドのコンパイルが含まれています。以下の Table 5 に、パラメータセットとスクリプトを備えた出力ファイルの例を示します。

1.4.2 パラメータセットとスクリプトローダーファイル

Table 5 iMotion™ソリューションデザイナーによって生成されたローダーファイルの例 (省略形)

% iMOTION configuration file % iMOTIONld version: V1.1.0 - 2022-10-11	ファイルヘッダー、 '%'で始まる行はコメント
%linker input file %filename: C:\...\generated\build.sdld %built: 2023-02-06, 14:20:50	iSD のビルド情報
% Erase Static Parameters a0 22 0f 01 00 % Static Parameters a0 20 0f 01 40 fe 04 00 00 … 00 00 … a0 20 0f 01 40 00 44 00 44 … 00 00 % Verify Static Parameters a0 21 0f 01 00	システムパラメータの消去、プログラミングおよび検証
% Parameter set: myFirstParameterSet % Erase Parameter Set a0 22 00 04 00 % Program Parameter Set a0 20 00 04 30 00 00 00 00 … 00 00 … a0 20 00 04 1c 02 22 00 20 … 20 8a % Verify Parameter Set a0 21 00 04 00	パラメータセットの消去、プログラミングおよび検証 (パラメータセクションには複数のパラメータセットを含むことができることに注意してください)
%---------- % Project Name : SpeedProfile %# Build Successful - 0 Error, 0 Warning %# Build Date and Time : 2023-02-06 14:20:31 %---------- % Script Translator Version : 2.00.00 %---------- % Script Object File %---------- % SCRIPT_USER_VERSION : 001.000	スクリプトコードヘッダー、 '%'で始まる行はコメント バージョン、サイズおよび行の総数の概要

%# Script Code Memory Size : 304 Bytes of 16 kBytes % Total Number of Lines : 122 %# Number of Global Variable(s) : 2, Number of Flash Variable(s) : 0, Data Memory Usage : 5 Bytes of 256 Bytes %# Task0 - Number of Instruction(s) : 23, Number of Variable(s) : 3, Data Memory Usage : 7 Bytes of 128 Bytes %# Task1 - Number of Instruction(s) : 5, Number of Variable(s) : 0, Data Memory Usage : 0 Bytes of 128 Bytes %----------	タスクごとの命令数、 スクリプトタスクの実行の分割に使用できます（[3]を参照）
% Erase Script memory a0 22 00 05 00 % Program Script a0 20 00 05 40 65 c3 ff … 56 00 … a0 20 00 05 30 66 01 3a … fb 1f % Verify Script a0 21 00 05 00	スクリプトの消去、プログラミングおよび検証

2 セキュアブートローダー (SBSL) – MCE プログラミング

SBSL は、暗号化された MCE ソフトウェアイメージのデバイスフラッシュへの安全な転送をサポートし、ソフトウェアとデバイスタイプの互換性を保証します。暗号化は、128 ビットのキーサイズを持つ Advanced Encryption Standard (AES) に基づきます。

2.1 セキュアローダーコマンド設定

セキュアローダーコマンドは、ファームウェア、パラメータおよびスクリプトコードをターゲットデバイスにプログラムし、ステータスを読み出すために使用されます。ローダープロトコルは、ISO7816 T=0 プロトコルを使用しており、通信フローを Figure 5 に示します。

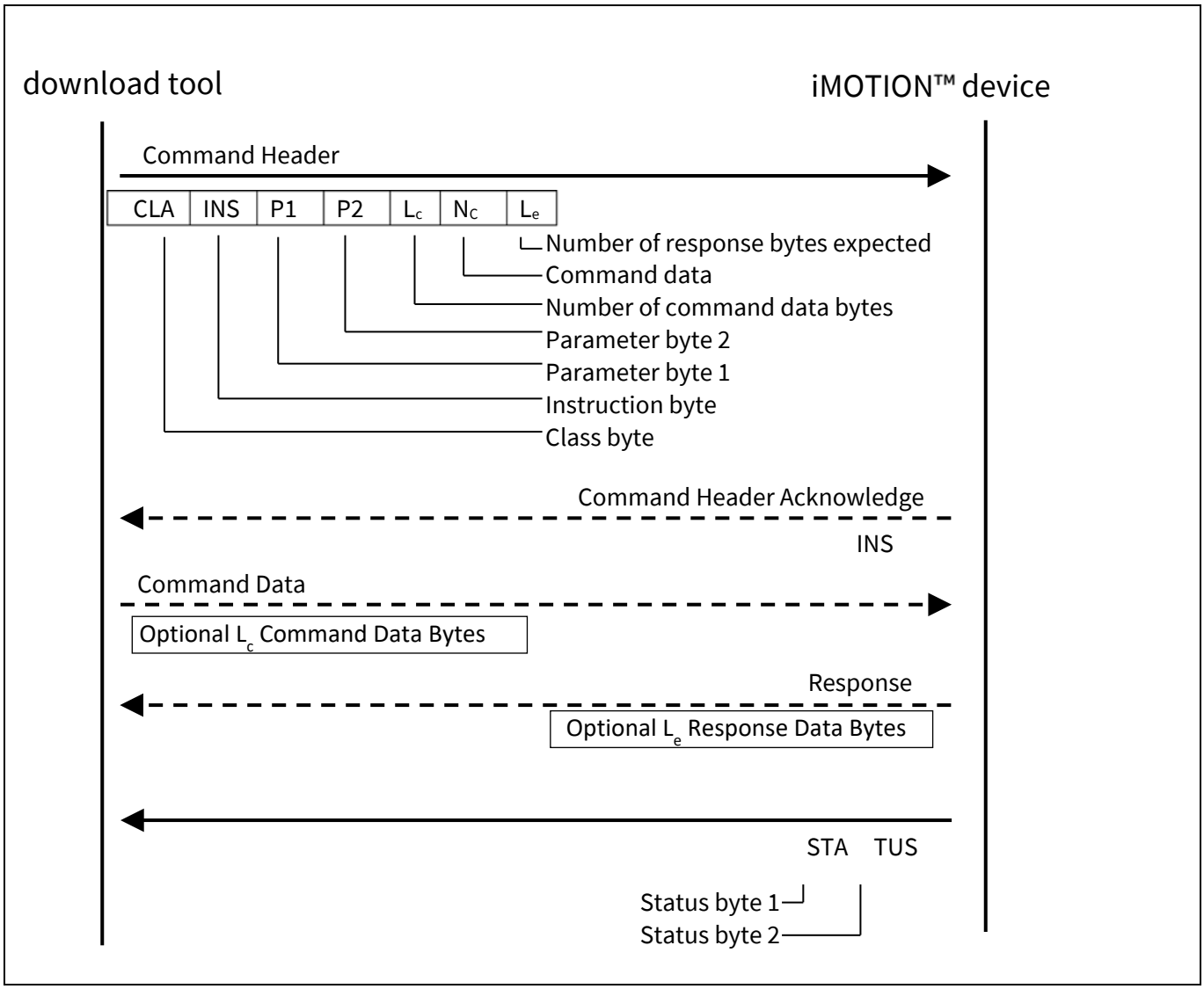


Figure 5 セキュアローダープロトコルのデータフローチャート

セキュアローダーコマンドについて

セキュアローダーコマンドは、コマンドクラス CLA とコマンド命令 INS を表す 2 つの 8 ビット整数によって識別されます。さらに、次のものが含まれます。

- 2 つの 8 ビットコマンドパラメータ P1 および P2
- 次のコマンドデータのバイト数を示す 8 ビットフィールド L_c
- N_c バイトのコマンドデータ
- 予想される応答バイトの最大数を示す 8 ビットフィールド L_e

コマンドは、ペイロードデータを一方向にのみ転送するように設計されています。つまり、コマンドのデータフィールドのローダー、または応答のデータフィールドのローダーに送信されますが、両方に同時に送信されることはありません。

フィールド L_c、N_c および L_e は、コマンドに応じたオプションです。L_e または L_C のいずれかが使用される場合、SBSL デバイスは受信した「INS」バイトを「ヘッダー確認応答」として最初に返します。

例えば、FLASH_CHIP_RESET コマンドの場合、L_c は 0x00 であるため、デバイスからの「ヘッダー確認応答」はありません。

FLASH_GET_SBSL_STATUS コマンドの場合、デバイスは最初に INS バイト (0x10) で応答し、その後に実際のデータが続き、最後に「成功」のステータス バイト (0x90 0x00) を送信します。

セキュアローダータイミングについて

ISO7816 規格では、T=0 は、通信が半二重であること、および通信回線の明確な「所有者」がいつでも存在することを定義しています。これは、データの送信を許可されているのはマスター (プログラマー) またはスレーブ (デバイス) のいずれかであることを意味します。したがって、プログラマーの実装はタイムアウトに依存するのではなく、それぞれのコマンドに対するデバイスからの応答をチェックする必要があります。

コマンドの受信後、スレーブは成功またはエラーのいずれかで「すぐに」反応します。セキュア ローダーがより多くの時間を必要とする場合、セキュア ローダー コマンド「FLASH_LOAD_CHECK_SIGNATURE」(「A0 21 00 00 00」) に対して、Figure 6 に示すように、1 つ以上の「待機時間延長リクエスト」を送信します。

この場合、スレーブはまだ通信回線を「所有」しており、マスターはスレーブが最終的に成功またはエラーを報告するまで待機する必要があります。

タイムアウトを実装して、回線の切断や通信の切断などのエラーから回復できます。具体的な値は、使用するプログラミング デバイスとボーレートに適合させる必要があります。
推奨開始値は次のとおりです。

- 50 ミリ秒　スレーブがすぐに応答できる場合 (例: FLASH_GET_SBSL_STATUS)
- 0.5 ... 1 秒　スレーブがフラッシュのプログラミング/消去または状態変更などのアクションを実行する必要がある場合

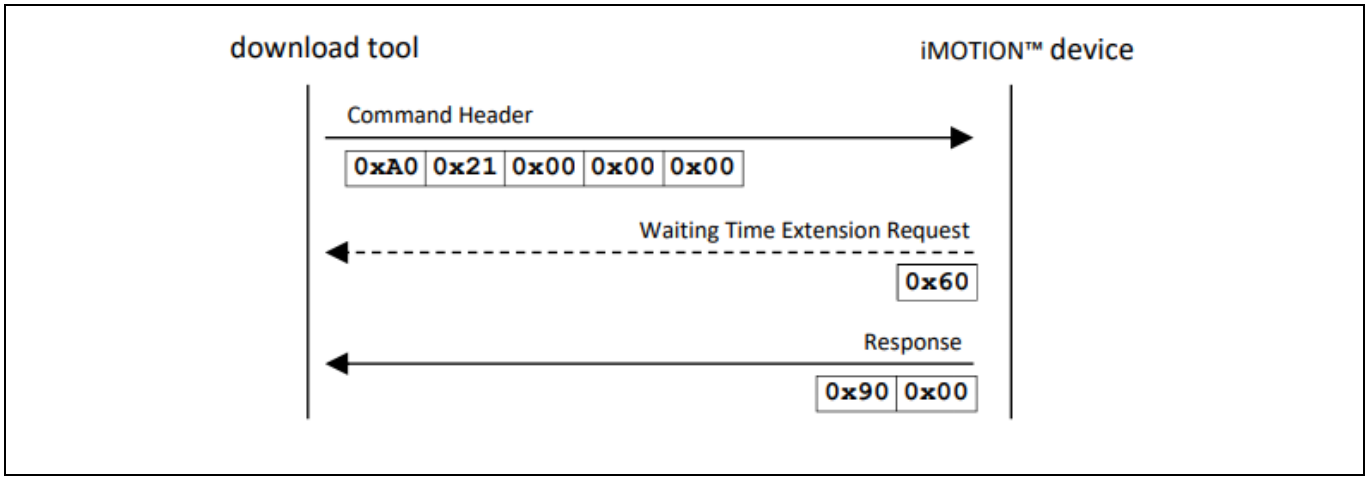


Figure 6　待機時間延長リクエストのデータフローチャート

2.2 サポートされているコマンド

以下の表に、モーションコントロールソフトウェアイメージをプログラミングするためにセキュアローダーでサポートされるコマンドを示します。

Table 6 SBSL コマンド

CLA	INS	名前	説明
0xA0	0x00	FLASH_CHIP_RESET	チップリセットをトリガーします
0xA0	0x10	FLASH_GET_SBSL_STATUS	39 バイトの SBSL ステータス情報 *rSbslStatus* を取得します
0xA0	0x12	FLASH_CHANGE_KEY	IP キー、K_{IP} およびそのラベル L_{IP} を更新します
0xA0	0x20	FLASH_LOAD_DATA	フラッシュメモリにデータをロードします
0xA0	0x21	FLASH_LOAD_CHECK_SIGNATURE	ダウンロードしたデータのチェックサムを確認します

2.3 一般的なコマンドステータスの応答

SBSL は、SBSL コマンドに応答して、常に 2 バイトのステータスワード、SW1 と SW2、およびデータ(該当する場合) を返します。

次の表に、一般的なコマンドステータスの応答値を示します。追加のコマンド固有の応答は、それぞれのコマンドの説明にリストされています。

Table 7 コマンドステータス応答値 SW1-SW2

SW1	SW2	意味	処理状態
0x90	0x00	成功 (Success)	通常
0x64	0x00	実行エラー: NVM は変更されていません	実行エラー
0x65	0x00	実行エラー: NVM が変更されました	
0x65	0x81	NVM が変更されました; メモリ障害	
0x67	0x00	間違った長さ (L_c または L_e)	確認エラー
0x69	0x82	不十分なセキュリティ状態	
0x69	0x83	認証方法がブロックされました	
0x69	0x84	参照データは使用できません	
0x69	0x85	使用条件が満たされていない	
0x6A	0x00	間違ったパラメータ P1 と P2	
0x6A	0x86	間違ったパラメータ P1 と P2	
0x6C	L'_e	間違った長さ L_e、SW2 は予想される長さ L'_e を示します	
0x6D	0x00	無効な命令バイト (INS)	
0x6E	0x00	無効なクラスバイト (CLA)	

2.4 FLASH_CHIP_RESET

このコマンドは、チップリセットをトリガーします。応答が返され、チップリセットが行われます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 8 FLASH_CHIP_RESET syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x00	0x00	0x00	0x00	-	-

応答

Table 9 FLASH_CHIP_RESET の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)

戻り値

コマンドは常に成功を報告します。

通常のリセットは、リセットコマンドの送信後の時間に発生

応答が返され、100 ms 後にチップリセットが行われます。

2.5 FLASH_GET_SBSL_STATUS

このコマンドは、39 バイトの SBSL ステータス情報 *rSbslStatus* をチップから取得します。*rSbslStatus* は、SBSL を処理するための追加の手順の決定や、パーソナライズ中に予想される状態との比較に役立ちます。

SBSL ステータスの準備中に、SBSL が以前に再アクティブ化されていて、ユーザーフラッシュ領域が消去されるようにスケジュールされている場合、SBSL はフラッシュ消去手順を実行します。待機時間延長 (WTX) リクエストは、プロトコルのタイミングに従うためにフラッシュ消去中に送信されます。WTX リクエストごとに、値 0x60 のバイトが送信されます。

Attention: ダウンロードツールは、フラッシュダウンロードトライアルカウンター (FDTC、バイト 20) のステータスを確認する必要があります。このカウンターの値は、最初は 0x10 = 16 であり、プログラミングの試行が失敗すると、例えば、間違ったキーやファイルが使用されたために減少します。値がゼロに達するとすぐに、デバイスは使用できなくなります。

セキュリティ

無し

パラメータ

無し

Syntax

Table 10 FLASH_GET_SBSL_STATUS syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x10	0x00	0x00	-	-	0x27

応答

Table 11 FLASH_GET_SBSL_STATUS の応答

データフィールド	SW1	SW2	状態
rSbslStatus	0x90	0x00	成功 (Success)
rSbslStatus	0x65	0x81	消去手順の失敗
-	0x67	0x00	間違った L_e

戻り値

このコマンドは *rSbslStatus* を返します。

Table 12 rSbslStatus の構造

オフセット	バイト	値	説明
0	4	“SBSL”	構造を識別するマジックネーム
4	1	0xC0	SBSL バージョンタグ
5	1	0x04	次のデータの長さ
6	1	0x06	iMOTION™
7	3	vr rb bb	ソフトウェアバージョン (v)、リビジョン (r)、ビルド (b)
10	1	0xC1	SBSL パッチバージョンタグ
11	1	0x03	次のデータの長さ
12	3	vr rb bb	パッチバージョン (v)、リビジョン (r)、ビルド (b)
15	1	0xC2	SBSL 状態タグ
16	1	0x04	次のデータの長さ
17	1	ULC	**SBSL ユニファイドライフサイクル**
18	1	V	V.0 ビット: 0→ SBSL は無効です。1→ SBSL は有効です V.1 ビット: 0→ K_{IP} は無効、またはセットです。1→K_{IP} は有効です その他: 予約済み
19	1	0x00	予約済み
20	1	FDTC	**フラッシュダウンロードトライアルカウンター** 現在の残りのダウンロード試行回数を示します。 ダウンロードシーケンスを開始するたびに、最初の「FLASH_LOAD_DATA」コマンドを受信すると、FDTC の値が 1 つ減少します。

オフセット	バイト	値	説明
			ダウンロードが正常に終了した場合 (チェックサム計算で確認、セクション 2.7 FLASH_LOAD_CHECK_SIGNATURE を参照)、FDTC は初期開始値の 0x10 にリセットされます。 ダウンロードが失敗した場合、FDTC は減少した値のままになります。 FDTC が 0 に達した場合、それ以上のフラッシュダウンロードは不可逆的にブロックされ、影響を受けるチップを新しいものと交換する必要があります。
21	1	0xC3	SBSLID タグ
22	1	0x10	次のデータの長さ
23	16	SBSL ID	SBSL ID

2.6 FLASH_LOAD_DATA

このコマンドは、設定可能な長さの暗号化された SBSL ダウンロードデータをチップに送信します。このデータは、SBSL の復号化モジュールに渡されます。

復号化後、データがデコードされ、ページ全体がフラッシュメモリにフラッシュされます。チェックサムはすべてのデータブロックに対して計算され、FLASH_LOAD_CHECK_SIGNATURE コマンドで検証されます。

Note: フラッシュダウンロードシーケンスは、次の FLASH_LOAD_CHECK_SIGNATURE コマンドを用いて明示的に終了する必要があります。

セキュリティ

無し

パラメータ

D_l は、l バイトの暗号化された SBSL データです。

Syntax

Table 13 FLASH_LOAD_DATA syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x20	0x00	0x00	l	D_l	-

応答

Table 14 FLASH_LOAD_DATA の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)
-	0x64	0x00	致命的 ファブアウトではない状態
-	0x65	0x00	SBSL 状態の更新中にエラーが発生。例えば、FDTC。
-	0x65	0x01	致命的 ユーザーフラッシュ範囲外のフラッシュへの書き込みが抑制され、チップがスリープモードになります。
-	0x65	0x81	致命的 NVM プログラミングエラーが発生し、チップがスリープモードになります。
-	0x69	0x82	不十分なセキュリティ状態。例えば、ダウンロードの試用版は残っていません。
-	0x69	0x84	ダウンロードストリームでエラーが見つかりました。
-	0x69	0x85	ダウンロードレコードの解釈中にエラーが発生しました。

戻り値

このコマンドは、成功またはエラーステータス値を返します。

「ファブアウトではない」状態 (0x64 0x00) が返される場合、次の原因が当てはまる可能性があります。

- チップはもうファブアウト状態ではありません。つまり、データは SBSL を使用してフラッシュメモリにすでにフラッシュされています。SBSL を再アクティブ化した後、2.5 章で説明される FLASH_GET_SBSL_STATUS コマンドを実行して、ユーザーフラッシュを消去する必要があります。

実行中に致命的なエラーが発生した場合、SBSL はコマンド応答を送信した直後にチップを再起動します。

2.7 FLASH_LOAD_CHECK_SIGNATURE

このコマンドは、すべてのデータブロックに対して計算されたチェックサムを確認し、フラッシュダウンロード手順を終了します。

実際のダウンロード署名の検証は、ダウンロードストリームの解釈内で実行されます。その結果は、このコマンドで取得されるまでメモリに保持されます。

ステータスを報告した直後に、セキュアブートストラップローダーは、成功した場合にダウンロードされたユーザーフラッシュソフトウェアをアクティブにするか、そうでなければチップを再起動します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 15 FLASH_LOAD_CHECK_SIGNATURE syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x21	0x00	0x00	0x00	-	-

応答

Table 16 FLASH_LOAD_CHECK_SIGNATURE の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)
-	0x65	0x00	間違ったハッシュ値
-	0x65	0x81	フラッシュプログラミングエラー
-	0x69	0x82	ダウンロードが開始されていません

戻り値

このコマンドは、成功またはエラーステータス値を返します。

3 設定モード - パラメータおよびスクリプトのプログラミング (MCE 1.x)

パラメータ設定とスクリプトは暗号化されておらず、モーションコントロールエンジン (MCE) に組み込まれているローダーを使用します。したがって、最初に MCE がデバイスにプログラムされている必要があります。

ローダーは、モーター (または PFC) 制御アプリケーションが実行されない特別なモードで実行されます。

パラメータとスクリプトのプログラミングに使用される通信プロトコルは、上記のセキュアローダープロトコルの場合と同じです。

データをダウンロードするコマンド (パラメータ設定またはスクリプトのいずれか) の後には、それぞれの検証コマンドが続く必要があることに注意してください。

インフィニオンのツールがパラメータ設定およびスクリプトに使用するファイル形式は、アプリケーションノートの冒頭に記載されています。ギャングプログラマーはさまざまな形式を使用できます。

Note: 1 つのバイトがバイトストリームで最初に LSB が転送されると、すべてのデータが広くなります。例えば、0x11010005 の IMC101T-T038 のチップ ID は、.. 0x05 0x00 0x01 0x11 として受信されます。

3.1 サポートされているコマンド

以下の表に、ローダーがパラメータ設定とスクリプトをプログラミングするためにサポートするコマンドを示します。

Table 17 パラメータローダーコマンド

CLA	INS	名	説明
0x00	0x6C	CONNECT	自動ボーレート検出に接続します
0xA0	0x00	CHIP_RESET	チップリセットのトリガー
0xA0	0x10	GET_STATUS	デバイスの設定ステータスを提供します
0xA0	0x11	GET_PARAMETER_SET_NAME	選択したパラメータページのメタデータを提供します
0xA0	0x12	GET_PARAMETER_SET_VALUES	選択したページのすべてのパラメータのリストを提供します
0xA0	0x14	GET_SYSTEM_CONFIG	システム設定ページを提供します
0xA0	0x15	GET_CHECKSUMS	ファームウェアバージョンとチェックサムを提供します
0xA0	0x18	CHANGE_BOOT_MODE	起動モードの変更をリクエストします
0xA0	0x20	DOWNLOAD_PARAMETER	パラメータまたはスクリプトを RAM にダウンロードします
0xA0	0x21	CHECK_PARAMETER	チェックサムを確認し、パラメータとスクリプトに使用されるフラッシュにページをプログラムします
0xA0	0x22	CLEAR_PARAMETER	パラメータページをクリア

3.2 CONNECT

このコマンドは、自動ボーレート検出を使用してチップに接続します。接続コマンドは 3 つのプロトコルすべてで同じであり、応答バイトはモードごとに一意です。したがって、ホストは接続コマンドへの応答バイトによって現在のモードを識別できます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 18 CONNECT syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0x00	0x6C	-	-	-	-	-

応答

Table 19 CONNECT の応答

データフィールド	SW1	SW2	状態
-	0xCD	-	成功 (Success)、設定モード

戻り値

無し

3.3 CHIP_RESET

このコマンドは、チップリセットをトリガーします。応答が返され、100 ms 後にチップリセットが行われます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 20 CHIP_RESET syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x00	0x00	0x00	0x00	-	-

応答

Table 21 CHIP_RESET の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)

戻り値

コマンドは常に成功を報告します。

3.4 GET_STATUS

get status コマンドは、チップ ID、ソフトウェアバージョン番号およびパラメータページの使用状況を表示します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 22 GET_STATUS syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x10	0x00	0x00	-	-	0x23

応答

Table 23 GET_STATUS の応答

データフィールド	SW1	SW2	状態
rConfStatus	0x90	0x00	成功 (Success)
rConfStatus	0x65	0x81	消去手順の失敗
-	0x67	0x00	間違った L_e

戻り値

このコマンドは *rConfStatus* を返します。

Table 24 rConfStatus の構造

オフセット	バイト	値	説明
0	4	“CONF”	構造を識別するマジックネーム
4	1	0xC0	SBSL バージョンタグ
5	1	0x0C	次のデータの長さ
6	4	チップ ID	iMOTION™チップ ID
10	4	vr rb bb	ハードウェアバージョン: バージョン, ビルド
14	4	features	デバイス/MCE の機能 ID

オフセット	バイト	値	説明
18	1	0xC1	パラメータページの使用法
19	1	0x0F	次のデータの長さ
20	1	table0	0 ページのテーブルタイプ
21	1	table1	1 ページのテーブルタイプ
22	1	table2	2 ページのテーブルタイプ
23	1	table3	3 ページのテーブルタイプ
24	1	table4	4 ページのテーブルタイプ
25	1	table5	5 ページのテーブルタイプ
26	1	table6	6 ページのテーブルタイプ
27	1	table7	7 ページのテーブルタイプ
28	1	table8	8 ページのテーブルタイプ
29	1	table9	9 ページのテーブルタイプ
30	1	table10	10 ページのテーブルタイプ
31	1	table11	11 ページのテーブルタイプ
32	1	table12	12 ページのテーブルタイプ
33	1	table13	13 ページのテーブルタイプ
34	1	table14	14 ページのテーブルタイプ

3.5 GET_PARAMETER_SET_NAME

get parameter set name コマンドは、選択したパラメータページのメタデータを提供します。これには、ページ番号、テーブルタイプ、パラメータの数およびページの名前が含まれます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 25 GET_PARAMETER_SET_NAME syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x11	ページ	0x00	-	-	0x13

応答

このコマンドは *rParsStatus* を返します。

Table 26 GET_PARAMETER_SET_NAME の応答

データフィールド	SW1	SW2	状態
rParsStatus	0x90	0x00	成功 (Success)
	0x65	0x80	チェックサムエラー
-	0x67	0x00	間違った長さ
	0x6A	0x86	禁止されているページ番号

戻り値

このコマンドは *rParsStatus* を返します。

Table 27 rParsStatus 構造

オフセット	バイト	値	説明
0	4	"PARS"	構造を識別するマジックネーム
4	1	0xC2	パラメータ設定名タグ
5	1	len	データの長さ
6	1	ページ	パラメータページ
7	1	テーブル	ページのテーブルタイプ
8	1	数	パラメータ数
9	10	名前	ページ名

3.6 GET_PARAMETER_SET_VALUE

get parameter set values コマンドは、パラメータ設定の内容を提供します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 28 GET_PARAMETER_SET_VALUE syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x12	ページ	0x00	0x00	-	-

応答

このコマンドは *rParvStatus* を返します。

Table 29 GET_PARAMETER_SET_VALUE の応答

データフィールド	SW1	SW2	状態
rParvStatus	0x90	0x00	成功 (Success)
	0x65	0x83	チェックサムエラー
	0x6A	0x86	禁止されているページ番号

戻り値

このコマンドは *rParvStatus* を返します。

Table 30 rParvStatus の構造

オフセット	バイト	値	説明
0	4	"PARV"	構造を識別するマジックネーム
4	1	0xC3	パラメータ設定名タグ
5	2	len	データの長さ
すべての機能ブロックの場合:			
	1	ページ番号	
	1	テーブル	

3.7 GET_CHECKSUMS

get checksums コマンドは、MCE のバージョンといくつかのチェックサムを表示します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 31 GET_CHECKSUMS syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x15	0x00	0x00	-	-	0x16

応答

Table 32 GET _ CHECKSUMS の応答

データフィールド	SW1	SW2	状態
rConfChk	0x90	0x00	成功 (Success)
-	0x67	0x00	間違った L_e

戻り値

このコマンドは *rConfChk* を返します。

Table 33 rConfChk の構造

オフセット	バイト	値	説明
0	4	“CHKS”	構造を識別するマジックネーム
4	1	0xCF	CHKS バージョンタグ
5	1	0x10	次のデータの長さ
6	4	FWVERSION	4:6:6:3:13 ビット形式のビットフィールド (major.minor.patch.type.buildnr)
10	4	FWCRC	MCE ファームウェアの CRC
14	4	PARCRC	mceos 設定の CRC
18	2	予約済み	ClassB 関連パラメータに対する CRC を予約済み
20	2	予約済み	すべてのパラメータに対する CRC を予約済み

3.8 CHANGE_BOOT_MODE

change boot mode コマンドは、他のデバイスモードの 1 つに変更するために使用されます。最初に応答が送信され、2 番目にコマンドが成功した場合、ブートモードが変更され、3 番目にデバイスが選択されたブートモードで自動的に再起動します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 34 CHANGE_BOOT_MODE syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x18	mode	～mode	-	-	0x00

Table 35 ブートモード

mode	～mode	説明
0x5D	0xA2	ブートローダーモード (SBSL)
0xCD	0x32	設定モード (CONF)
0xAD	0x52	アプリケーションモード (MCEDesigner + ユーザーCOM)
0xAF	0x50	フェイルセーフモード (クラス B 障害)

0x5D:

- MCE をプログラミングするための SBSL モード

0xCD:

- パラメータとスクリプトの (再) プログラミングのための設定モード

0xAD:

- デバイスを実行するためのアプリケーションモード

0xAF:

- クラス B 安全障害の結果としてのフェイルセーフモード

応答

Table 36 CHANGE_BOOT_MODE の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success) したら、新しいモードに変更
	0x65	0x81	NVM プログラミングエラー
-	0x69	0x84	mode と～mode の不一致
	0x6A	0x86	間違ったブートモード

戻り値

無し

3.9 DOWNLOAD_PARAMETER (スクリプトにも使用されます)

download parameters コマンドは、データパケットをデバイスに送信し、そこでバッファに保存されます。ダウンロード後、チェックサムを計算してフラッシュ書き込みを完了するには、checkparameter コマンドを呼び出す必要があります。

セキュリティ

無し

パラメータ

D_l は l バイトの暗号化された SBSL データです。

Syntax

Table 37 DOWNLOAD_PARAMETER syntax (パラメータ設定)

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x20	ページ	0x01	l	D_l	-

Table 38 DOWNLOAD_PARAMETER syntax (スクリプト)

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x20	0x00	0x02	l	D_l	-

応答

Table 39 DOWNLOAD_PARAMETER の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success)

戻り値

無し

Note: コマンド DOWNLOAD_PARAMETER は、ステータス応答の直前にデータ ペイロードが受信された後でのみ、INS バイトを送信します。

3.10 CHECK_PARAMETER (スクリプトにも使用されます)

check parameter コマンドは、バッファ内のデータのチェックサムを計算し、有効なデータの場合、選択されたパラメータページをプログラムします。

セキュリティ

無し

パラメータ

無し

Syntax

Table 40 CHECK_PARAMETER syntax (パラメータ設定)

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x21	ページ	0x01	0x00	-	-

Table 41 CHECK_PARAMETER syntax (スクリプト)

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x21	0x00	0x02	0x00	-	-

応答

Table 42 CHECK_PARAMETER の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success)
	0x65	0x80	RAM 内のチェックサムエラー、プログラムされていません
	0x65	0x81	NVM プログラミングエラー
	0x65	0x82	ページは空ではなく、プログラムされていません

戻り値

無し

3.11 CLEAR_PARAMETER

clear parameter コマンドは、選択したパラメータページを消去します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 43 CLEAR_PARAMETER syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x22	ページ	0x00	0x00	-	-

応答

Table 44 CLEAR_PARAMETER の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success)
	0x69	0x84	NVM の消去に失敗

戻り値

無し

4 設定モード - パラメータおよびスクリプトのプログラミング (MCE 5.x)

パラメータ設定とスクリプトは暗号化されておらず、モーションコントロールエンジン (MCE) に組み込まれているローダーを使用します。したがって、最初に MCE がデバイスにプログラムされている必要があります。

ローダーは、モーター (または PFC) 制御アプリケーションが実行されない特別なモードで実行されます。

パラメータとスクリプトのプログラミングに使用される通信プロトコルは、上記のセキュアローダープロトコルの場合と同じです。

データをダウンロードするコマンド (パラメータ設定またはスクリプトのいずれか) の後には、それぞれの検証コマンドが続く必要があることに注意してください。

インフィニオンのツールがパラメータ設定およびスクリプトに使用するファイル形式は、アプリケーションノートの冒頭に記載されています。ギャングプログラマーはさまざまな形式を使用できます。

Note: 1 バイト以上のすべてのデータは、LSB ファーストのバイトストリームで転送されます。例えば、IMC101T-T038 のチップ ID 0x11010005 は、.. 0x05 0x00 0x01 0x11 として受信されます。

4.1 サポートされているコマンド

以下の表に、ローダーがパラメータ設定とスクリプトをプログラミングするためにサポートするコマンドを示します。

Table 45 パラメータローダーコマンド

CLA	INS	名	説明
0x00	0x6C	CONNECT	自動ボーレート検出に接続します
0xA0	0x00	CHIP_RESET	チップリセットのトリガー
0xA0	0x10	GET_STATUS	デバイスの設定ステータスを提供します
0xA0	0x14	GET_PAGE_DATA	ページデータをダウンロードします
0xA0	0x15	GET_CHECKSUMS	ファームウェアバージョンとチェックサムを提供します
0xA0	0x18	CHANGE_BOOT_MODE	起動モードの変更をリクエストします
0xA0	0x20	DOWNLOAD_DATA	パラメータまたはスクリプトを RAM にダウンロードします
0xA0	0x21	CHECK_DATA	チェックサムを確認し、フラッシュにデータをプログラムします
0xA0	0x22	CLEAR_DATA	データページをクリア

4.2 CONNECT

このコマンドは、自動ボーレート検出を使用してチップに接続します。接続コマンドは 3 つのプロトコルすべてで同じであり、応答バイトはモードごとに一意です。したがって、ホストは接続コマンドへの応答バイトによって現在のモードを識別できます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 46 CONNECT syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0x00	0x6C	-	-	-	-	-

応答

Table 47 CONNECT の応答

データフィールド	SW1	SW2	状態
-	0xCD	-	成功 (Success)、設定モード

戻り値

無し

4.3 CHIP_RESET

このコマンドは、チップリセットをトリガーします。応答が返され、100 ms 後にチップリセットが行われます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 48 CHIP_RESET syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x00	0x00	0x00	0x00	-	-

応答

Table 49 CHIP_RESET の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)

戻り値

コマンドは常に成功を報告します。

4.4 GET_STATUS

get status コマンドは、チップ ID、ソフトウェアバージョン番号およびパラメータページの使用状況を表示します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 50 GET _STATUS syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x10	0x00	0x00	-	-	0x23

応答

Table 51 GET _STATUS の応答

データフィールド	SW1	SW2	状態
rConfStatus	0x90	0x00	成功 (Success)
-	0x67	0x00	間違った L_e

戻り値

このコマンドは *rConfStatus* を返します。

Table 52 rConfStatus の構造

オフセット	バイト	値	説明
0	4	“CONF”	構造を識別するマジックネーム
4	1	0xC0	SBSL バージョンタグ
5	1	0x0C	次のデータの長さ
6	4	Chip ID	iMOTION™チップ ID
10	4	vr rb bb	ハードウェアバージョン: バージョン, ビルド
14	4	features	デバイス/MCE の機能 ID
18	1	0xC1	パラメータページの使用法
19	1	0x0F	次のデータの長さ
20	1	table0	0 ページのテーブルタイプ
21	1	table1	1 ページのテーブルタイプ
22	1	table2	2 ページのテーブルタイプ
23	1	table3	3 ページのテーブルタイプ
24	1	table4	4 ページのテーブルタイプ
25	1	table5	5 ページのテーブルタイプ
26	1	table6	6 ページのテーブルタイプ

オフセット	バイト	値	説明
27	1	table7	7 ページのテーブルタイプ
28	1	table8	8 ページのテーブルタイプ
29	1	table9	9 ページのテーブルタイプ
30	1	table10	10 ページのテーブルタイプ
31	1	table11	11 ページのテーブルタイプ
32	1	table12	12 ページのテーブルタイプ
33	1	table13	13 ページのテーブルタイプ
34	1	table14	14 ページのテーブルタイプ

4.5 GET_PAGE_DATA

get system config コマンドは、システム設定ページをダウンロードします。

セキュリティ

無し

パラメータ

無し

Syntax

Table 53 GET _PAGE_DATA syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x14	ページ	データタイプ	0x00	-	-

Table 54 データタイプおよびページ

P1 (ページ)	P2 (データ)	ダウンロードするデータ
0x0F	0x01	システムパラメータ

応答

Table 55 GET _PAGE_DATA の応答

データフィールド	SW1	SW2	状態
rPageData	0x90	0x00	成功 (Success)
	0x65	0x83	チェックサムエラー
	0x6A	0x86	無効なページ

戻り値

このコマンドは、システム設定ページのデータである *rPageData* を返します。

4.6 GET_CHECKSUMS

get checksums コマンドは、MCE のバージョンといくつかのチェックサムを表示します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 56 GET_CHECKSUMS syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x15	0x00	0x00	-	-	0x16

応答

Table 57 GET _ CHECKSUMS の応答

データフィールド	SW1	SW2	状態
rConfChk	0x90	0x00	成功 (Success)
-	0x67	0x00	間違った L_e

戻り値

このコマンドは *rConfChk* を返します。

Table 58 rConfChk の構造

オフセット	バイト	値	説明
0	4	“CHKS”	構造を識別するマジックネーム
4	1	0xCF	CHKS バージョンタグ
5	1	0x10	次のデータの長さ
6	4	FWVERSION	4:6:6:3:13 ビット形式のビットフィールド (major.minor.patch.type.buildnr)
10	4	FWCRC	MCE ファームウェアの CRC
14	4	PARCRC	mceos 設定の CRC
18	2	予約済み	ClassB 関連パラメータに対する CRC を予約済み
20	2	予約済み	すべてのパラメータに対する CRC を予約済み

4.7 CHANGE_BOOT_MODE

change boot mode コマンドは、他のデバイスモードの 1 つに変更するために使用されます。最初に応答が送信され、2 番目にコマンドが成功した場合、ブートモードが変更され、3 番目にデバイスが選択されたブートモードで自動的に再起動します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 59 CHANGE_BOOT_MODE syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x18	mode	～mode	-	-	0x00

Table 60 ブートモード

mode	～mode	説明
0x5D	0xA2	ブートローダーモード (SBSL)
0xCD	0x32	設定モード (CONF)
0xAD	0x52	アプリケーションモード
0xAF	0x50	フェイルセーフモード (クラス B 障害)

0x5D:

- MCE をプログラミングするための SBSL モード

0xCD:

- パラメータとスクリプトの (再) プログラミングのための設定モード

0xAD:

- デバイスを実行するためのアプリケーションモード

0xAF:

- クラス B 安全障害の結果としてのフェイルセーフモード

応答

Table 61 CHANGE_BOOT_MODE の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success) したら、新しいモードに変更
	0x65	0x81	NVM プログラミングエラー
-	0x69	0x84	mode と～mode の不一致
	0x6A	0x86	間違ったブートモード

戻り値

無し

4.8 DOWNLOAD_DATA

ダウンロード コマンドは、データ パケット (パラメータ セット、スクリプトなど) をデバイスに送信し、バッファに保存します。ダウンロード後、チェックサムを計算してフラッシュ書き込みを完了するために、チェック データ コマンドを呼び出す必要があります。

セキュリティ

無し

パラメータ

ダウンロードデータタイプとページ (必要な場合) は、Table 63 に従って P1 と P2 に示されます。
D_l は l バイトの暗号化された SBSL データです。

Syntax

Table 62 DOWNLOAD_ DATA syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x20	ページ	データタイプ	l	D_l	-

Table 63 データタイプおよびページ

P1 (ページ)	P2 (データ)	ダウンロードするデータ
0x00 ~ 0x0F	0x04	パラメータ設定
0x00	0x05	スクリプト
0x0F	0x01	システムパラメータ

応答

Table 64 DOWNLOAD_PARAMETER の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success)
	0x6A	0x86	禁止されているページ番号

戻り値

無し

Note: コマンド DOWNLOAD_PARAMETER は、ステータス応答の直前にデータ ペイロードが受信された後でのみ INS バイトを送信します。

4.9 CHECK_DATA

データチェックコマンドは、バッファ内のデータのチェックサムを計算し、有効なデータの場合は、選択したパラメータデータをプログラムします。

セキュリティ

無し

パラメータ

ダウンロードデータタイプとページ (必要な場合) は、Table 66 に従って P1 と P2 に示されます。

Syntax

Table 65 CHECK_DATA syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x21	ページ	データタイプ	0x00	-	-

Table 66 データタイプおよびページ

P1 (ページ)	P2 (データ)	ダウンロードするデータ
0x00 .. 0x0F	0x04	パラメータ設定
0x00	0x05	スクリプト
0x0F	0x01	システムパラメータ

応答

Table 67 CHECK _PARAMETER の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success)
	0x65	0x80	RAM 内のチェックサムエラー、プログラムされていません
	0x65	0x81	NVM プログラミングエラー
	0x65	0x82	ページは空ではなく、プログラムされていません
	0x6A	0x86	禁止されているページ番号

戻り値

無し

4.10 CLEAR_DATA

コマンドは、選択したデータ (例えば、パラメータページまたはスクリプト) を消去します。

セキュリティ

無し

パラメータ

ダウンロード データ タイプとページ (必要な場合) は、Table 69 に従って P1 と P2 に示されます。

Syntax

Table 68 CLEAR_DATA syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x22	ページ	データタイプ	0x00	-	-

Table 69 データタイプおよびページ

P1 (ページ)	P2 (データ)	消去するデータ
0x00 .. 0x0F	0x04	パラメータ設定
0x00	0x05	スクリプト
0x0F	0x01	システムパラメータ

応答

Table 70 CLEAR _DATA の応答

データフィールド	SW1	SW2	状態
	0x90	0x00	成功 (Success)
	0x69	0x84	NVM の消去に失敗
	0x6A	0x86	禁止されたページ番号

戻り値

無し

5 安全故障モード

5.1 サポートされているコマンド

以下の Table 71 に、ローダーがパラメータ設定とスクリプトをプログラミングするためにローダーでサポートするコマンドを示します。

Table 71 サポートされているコマンドのリスト

CLA	INS	名	説明
0xA0	0x00	CHIP_RESET	チップリセットのトリガー
0xA0	0x10	GET_STATUS	デバイスの安全障害ステータスを提供
0xA0	0x18	CHANGE_BOOT_MODE	起動モードの変更をリクエスト

5.2 CHIP_RESET

このコマンドは、チップリセットをトリガーします。応答が返され、100 ms 後にチップリセットが行われます。

セキュリティ

無し

パラメータ

無し

Syntax

Table 72 CHIP_RESET syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x00	0x00	0x00	0x00	-	-

応答

Table 73 CHIP_RESET の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)

戻り値

コマンドは常に成功を報告します。

5.3 GET_STATUS

get status コマンドは、チップ ID、ソフトウェアバージョン番号、および安全障害リセット状態を表示します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 74 GET_STATUS syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x10	0x00	0x00	-	-	0x18

応答

Table 75 GET_STATUS の応答

データフィールド	SW1	SW2	状態
rConfStatus	0x90	0x00	成功 (Success)
rConfStatus	0x67	0x00	間違った L_e

戻り値

このコマンドは *rConfStatus* を返します。

Table 76 rConfStatus 構造

オフセット	バイト	値	説明
0	4	“FSMD”	構造を識別するマジックネーム
4	1	0xF0	SBSL バージョンタグ
5	1	0x0C	次のデータの長さ
6	4	チップ ID	iMOTION™チップ ID
10	4	vr rb bb	ハードウェアバージョン: バージョン、ビルド
14	4	features	デバイス/MCE の機能 ID
18	1	0xF1	安全バージョンタグ
19	1	0x 04	次のデータの長さ
20	4		故障リセット状態

5.4 CHANGE_BOOT_MODE

change boot mode コマンドは、ブートモードを変更するための要求です。最初に応答が送信され、2 番目にコマンドが成功した場合、ブートモードが変更され、3 番目にデバイスが選択されたブートモードで自動的に再起動します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 77 CHANGE_BOOT_MODE syntax

CLA	INS	P1	P2	L_c	データフィールド	L_e
0xA0	0x18	mode	～mode	-	-	0x00

Table 78 ブートモード

mode	～mode	説明
0x5D	0xA2	ブートローダーモード (SBSL)
0xCD	0x32	設定モード (CONF)
0xAD	0x52	アプリケーションモード (MCE 実行中)
0xAF	0x50	フェイルセーフモード (クラス B 障害)

応答

Table 79 CHANGE_BOOT_MODE の応答

データフィールド	SW1	SW2	状態
-	0x90	0x00	成功 (Success)、新しいモードに変更
-	0x69	0x84	mode と～mode の不一致
-	0x6A	0x86	間違ったブートモード

戻り値

無し

6 アプリケーションモード- MCE 1.x コマンド設定

iMOTION™デバイスが MCE のアプリケーションモードで、少なくとも 1 つの有効なパラメータセットがプログラムされている場合、電源投入後、パラメータセットで設定されたとおりに自動的に起動します。

MCE、パラメータセットまたはスクリプトのアップデートを実行するには、デバイスを以下のいずれかのモードにする必要があります。設定モードまたは SBSL モードのいずれかにする必要があります。これは、CHANGE_BOOT_MODE コマンドを使用するか、電源投入直後にパルスを送信することによって行えます。どちらの方法も以下に説明します。

iMOTION™デバイスで利用可能な UART は、異なる通信プロトコルに設定できます。詳細は、アプリケーションノート「iMOTION™製品とのインターフェース」[2]に記載されています。通信プロトコルに応じて UART の通信プロトコル (MCE またはユーザーモード) により、CHANGE_BOOT_MODE コマンドの syntax が異なります。

設定モードでは、新しいパラメータセットやスクリプトをプログラム、またはデバイスを SBSL モードに切り替えて MCE 自体を再プログラムできます。(3 設定モード - パラメータおよびスクリプトのプログラミング (MCE 1.x))

MCE コマンドについて

- 接続ボーレートは 115200 bps に固定されています
- 接続チェックデータフレームを使用して通信を確立する必要があります

6.1 サポートされているコマンド

以下の Table 80 に、MCE のアプリケーションモードでサポートされるコマンドを示します。ここには、プログラミング目的に関連するコマンドのみがリストされます。

Table 80 MCE 1.x コマンド

名前	説明
CHECK_COMMUNICATION	通信とデバイスの状態の確認
CHANGE_BOOT_MODE	ブートモードを SBSL または設定モードに変更

6.2 CHECK_COMMUNICATION

このコマンドは、デバイスのステータスを確認するために使用され、UART が MCE モードであるときのみ有効です。

セキュリティ

無し

パラメータ

無し

Syntax

Table 81 CHECK_COMMUNICATION syntax

バイト 1	バイト 2	バイト 3	バイト 4			
0x7E	0x13	0x7E	0x13	-	-	-

応答

Table 82 CHECK_COMMUNICATION の応答

バイト 1	バイト 2	バイト 3	バイト 4	
0x7E	0x17	0x7E	0x17	
0x7E	0x1B	0x7E	0x1B	

戻り値

無し

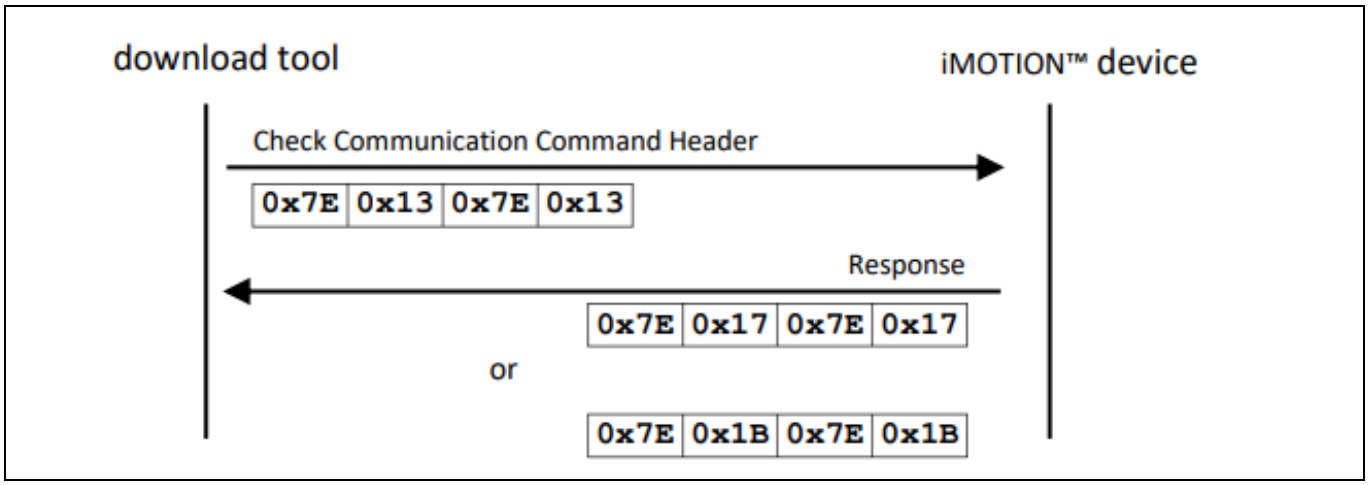


Figure 7 MCE 通信プロトコルチェック

6.3 CHANGE_BOOT_MODE

change boot mode コマンドは、別のデバイス状態に変更するための要求です。最初に応答が送信され、2 番目にコマンドが成功した場合、ブートモードが変更され、3 番目にデバイスが選択されたブートモードで自動的に再起動します。

セキュリティ

無し

パラメータ

無し

Syntax

Table 83 CHANGE_BOOT_MODE syntax

バイトシーケンス	説明
0x7E 0x02 0x80 0x31 0x51 0x81 0x10 0xFA 0xF8 0x7E 0x87	SBSL モードに変更します
0x7E 0x02 0x80 0x38 0x51 0x82 0x10 0x32 0xCD 0x7E 0x9C	設定モードに変更します

応答

Table 84 CHANGE_BOOT_MODE の応答 (MCE)

対応	状態
0xFE	成功 (Success) 後、SBSL モードに変更
0x7E 0x01 0x7E 0x01	成功 (Success) 後、設定モードに変更

戻り値

無し

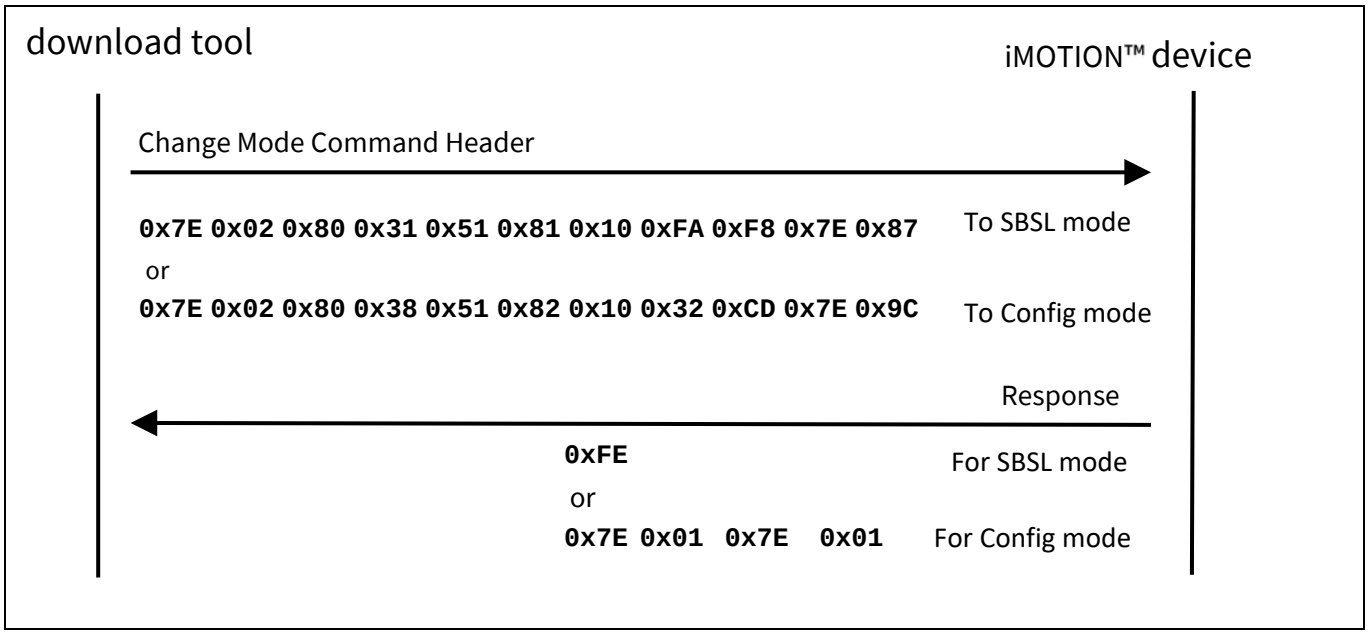


Figure 8 ブートモードを変更する MCE コマンド

7 アプリケーションモード- MCE 5.x コマンド設定

iMOTION™デバイスが MCE を使用したアプリケーションモードにあり、少なくとも 1 つの有効なパラメータセットがプログラムされている場合、パラメータセットを介して設定されたとおりに、電源投入後に自動的に起動します。

iMOTION™デバイスで使用可能な UART は、さまざまな通信プロトコルに設定できます。詳細は、アプリケーションノート iMOTION™製品とのインターフェース [2] に記載されています。UART がユーザーモード用に設定されている場合、CHANGE_BOOT_MODE の syntax は次のとおりです。

設定モード内で、新しいパラメータセットまたはスクリプトをプログラムしたり、デバイスを SBSL モードに切り替えて MCE 自体を再プログラムできます。(3 設定モード - パラメータおよびスクリプトのプログラミング (MCE 1.x))。

ユーザーモードコマンドについて

- ボーレートは 2400 bps から 230400 bps の間で設定可能です。
- ノードアドレス指定のあるデータフレームが使用されます。

7.1 サポートされるコマンド

以下の Table 85 に、MCE のアプリケーションモードでサポートされるコマンドを示します。プログラミング目的に関連するコマンドのみがここにリストされています。ユーザーモードには、通信をチェックするためのコマンドがいくつか用意されています。MCE リファレンスマニュアル[3] を参照してください。

Table 85 MCE コマンド

名	説明
CHANGE_BOOT_MODE	起動モードを SBSL または設定モードに変更します

7.2 CHANGE_BOOT_MODE

このコマンドは、UART がユーザーモードに設定されている場合に、アプリケーションモードから SBSL または設定モードに変更するための要求です。[2]

フォーマットは、定義された UART データ フレーム フォーマット[7]に従っており、最初のバイトがノード アドレス、最後の 2 バイトがチェックサムです。したがって、別のノードがアドレス指定された場合は、チェックサムを再計算する必要があります。

セキュリティ

無し

パラメータ

無し

Syntax

Table 86 CHANGE_BOOT_MODE syntax (ユーザーモード)

バイトシーケンス	説明
0x01 0x20 0x18 0x00 0x5D 0xA2 0x8A 0x3D	SBSL モードに変更します
0x01 0x20 0x18 0x00 0xCD 0x32 0x1A 0xAD	設定モードに変更します

応答

無し

戻り値

無し

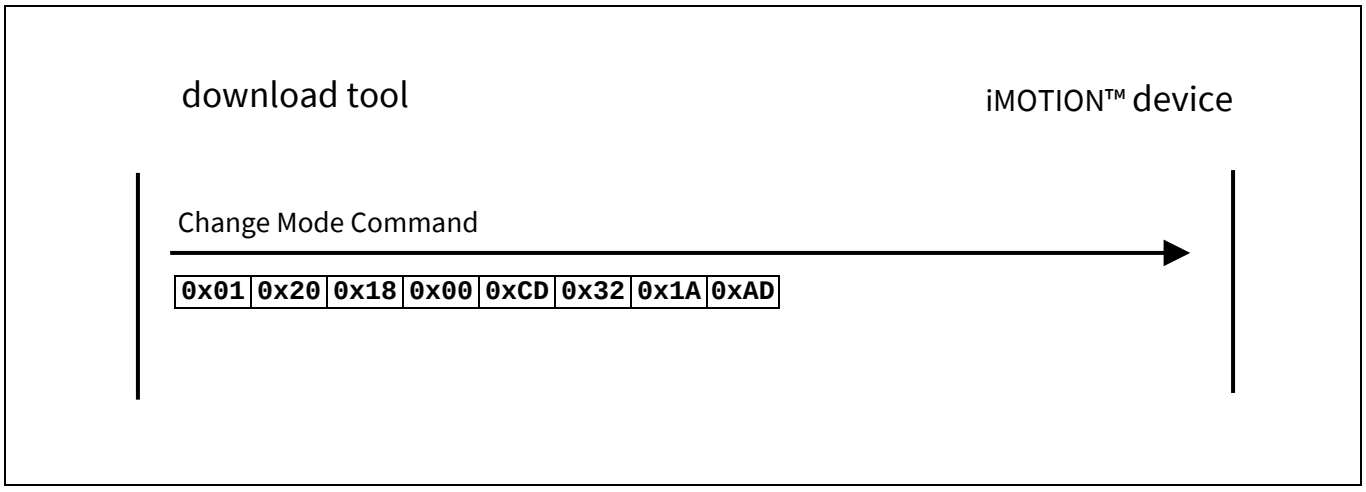


Figure 9 設定モードに変更するためのユーザーモード UART コマンド

8 キャッチ・アット・スタートアップ方式によるブートモードの変更

MCE ファームウェアと有効なパラメータ セットがプログラムされている場合、デバイスは電源投入後に直接アプリケーションを起動します。パラメータ セット、スクリプトまたは MCE 自体のいずれかの更新を開始するには、最初のステップとしてブート モードを設定モードに変更する必要があります。

iMOTION™デバイスは、起動直後に RXD0 ピンを監視して、定義されたパルス幅 155 µs のロー パルスを受信します。これは、57,600 Baud でのデータ値 0x00 に相当します。パルスは起動後 100 ms 以内に印加する必要があります。パルスを受信すると、デバイスはピン TXD0 でデータ値 0x06 で応答し、設定モードに入ります。

この後、設定モード章の 3.2 CONNECT で説明されているように、デバイスに接続できます。

設定モードへ変更のキャッチ手順について

- RXD0 ピンに 155 µs 幅のローパルスを送信
  これは以下と同等です
- 57600 bps の接続ボーレートで 0x00 を送信
- デバイスは 0x06 で応答し、設定モードに変更します

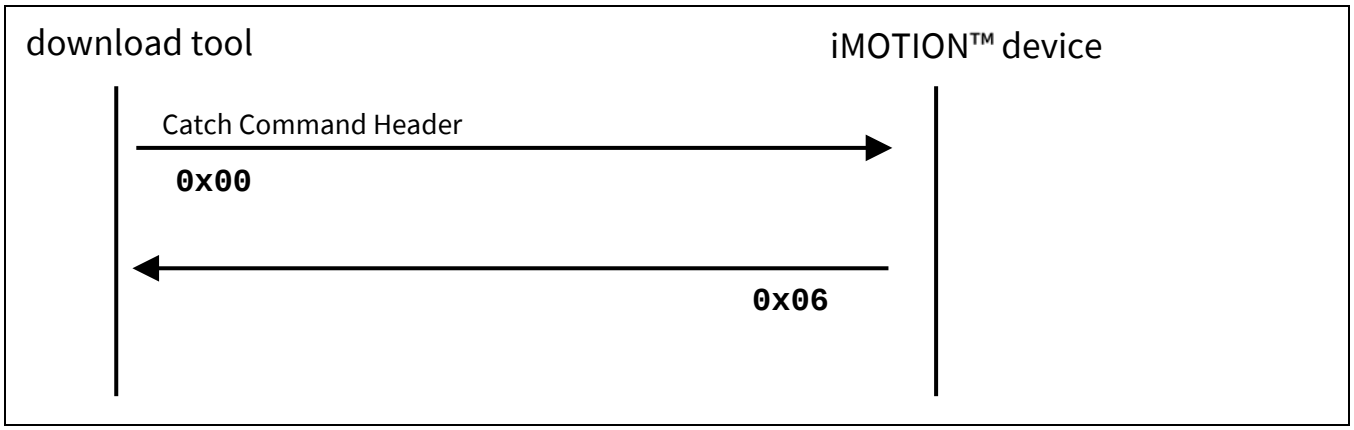


Figure 10 起動時のキャッチデータフローチャート

Note: MCE 5.x では、デバイスからの応答バイト“0x06”を正常に受信してから、設定モードでデバイスに”Connect”コマンドを送信するまでに、ある程度の待ち時間を入れる必要があります。推奨待ち時間は 40 ms です。

9 事例紹介

9.1 デバイスの SBSL ステータスの読み出し

SBSL ID と詳細情報を含む SBSL ステータスは、次のようにデバイスから読み出せます。

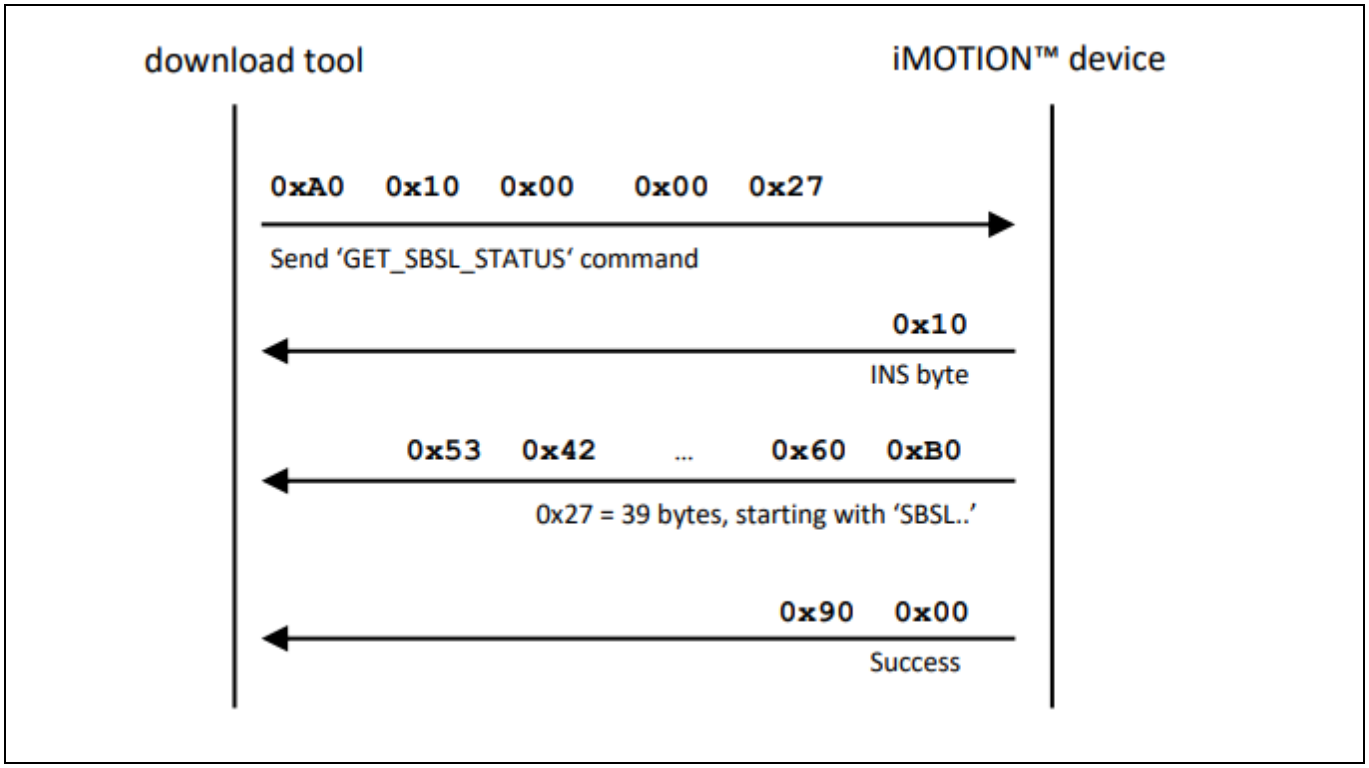


Figure 11 SBSL Id 評価のプロトコルフロー

完全な応答のレイアウトについては、Table 12 を参照してください。

応答には、バイト 20 として Flash Download Trial カウンター (FDTC) が含まれます。

SBSL ステータスの最後に返される SBSL-ID は、それぞれのデバイスデータシートに記載されています。

Note: *生産でのプログラミング環境では、実際のプログラミング手順の前にこのコマンドを使用して正しいデバイスタイプを確認し、誤ったプログラミングを回避するために FDTC を確認することを推奨します。*
以前にプログラムされたデバイスの場合、このコマンドの使用はプログラミングを開始する前に必須です。

9.2 製造ラインの最後でのプログラミング

以下に、製造時に iMOTION™デバイスをプログラムするために従うべき推奨フローを示します。

ここでの推奨事項は、プログラミング速度と堅牢性の観点からバランスの取れたアプローチです。これを実装し、場合によってはさらに最適化するのはメーカーの責任です。

高いボーレートを確実に達成できるプロのプログラミングツールを使用することを強く推奨します。この目的で開発ツール iMOTION™ Link を使用することは推奨しません。

ほとんどのコマンドは、それぞれの章に記載されるように、応答または「待機時間延長要求」(0x60) を提供します。これらのチェックは、以下の各ステップでは提供されません。

以下の手順を推奨

1. デバイスの電源を入れ、1.2 UART 設定 (0x00 0x93) で説明されている手順を使用して拡張ボーレートをネゴシエートしてください。達成可能なボーレートは、プログラマーとプログラミング環境によって異なり、慎重に検証する必要があります。
2. GET_SBSL_STATUS コマンドを使用して、デバイスが SBSL 状態にあること、デバイスタイプが正しいこと (SBSL ID)、および FDTC が初期値にあることを確認してください。(9.1 デバイスの SBSL ステータスの読み出しを参照)
3. オンチップに保存されている SBSL ID は、MCE イメージの復号化のベースとして使用され、イメージの暗号化に使用されるキーと一致する必要があります。不一致の場合、ダウンロード操作は中止され、デバイスはフラッシュコンテンツが消去された状態で SBSL モードのままになります。
ダウンロードが中止されると、FDTC が減少します。
4. 次のステップでは、MCE ファームウェアを一度に 1 行ずつデバイスにダウンロードしてください。load コマンドは上記で指定され (2.6 FLASH_LOAD_DATA)、使用可能になるローダーファイルにも含まれます (1.3.1 ローダーファイル形式の説明)。
5. プログラミングは、FLASH_LOAD_CHECK_SIGNATURE コマンドを送信することで終了します。これは、ローダーファイル (*.ldf) にも記載されています。このコマンドは、プログラムされたフラッシュ全体のチェックサムを計算するため、応答に時間がかかる場合があり、「待機時間延長リクエスト」を送信します (Figure 6 を参照)。シグニチャチェックが成功した後、デバイスは「成功」(0x90 0x00) を送信し、設定モードに移行するリセットを実行します。
6. ダウンロードツールは、標準のボーレートモード (0x00 06C) を使用してデバイスに再接続し、デバイスが設定モードになっていることを確認する必要があります (0xCD にて応答)。
7. これで、第 3 章「設定モード- パラメータおよびスクリプトのプログラミング」に記載されているコマンドを使用して、パラメータ設定をダウンロードできます。(結合された) ローダーファイル (*.ldf) が使用される場合、これらのコマンドはファイルに含まれます。パラメータとスクリプトの場合、最初に RAM にロードされ、フラッシュに永続的に保存するには CHECK_PARAMETER コマンドが必要であることに注意してください。
8. CHECK_PARAMETER コマンドを送信して、チェックサムを計算し、フラッシュにプログラムしてください。
9. スクリプトのプログラミングは最後のステップであり、パラメータ設定に使用されるのと同じコマンドを使用します。
10. CHECK_PARAMETER コマンドを送信して、チェックサムを計算し、フラッシュにプログラムしてください。

これで、デバイスを使用する準備が整いました。

9.3 MCE の更新の実行

SBSL モードへの移行後、モーションコントロールエンジンの更新は最初のプログラミングと同じ方法で行われます。

唯一の違いは、更新の場合、**GET_SBSL_STATUS** コマンドの使用 (上記ステップ 2.) は**必須**です。以前にプログラムされたデバイスの場合、このコマンドはフラッシュの消去を開始し、実際のプログラミングを準備します。

9.4 ユーザーモードに設定された UART からの MCE の更新

デバイスがユーザーモードに設定された UART で実行されている場合、次のシーケンスが必要です。

- CHANGE_BOOT_MODE (ユーザーモード、7) を送信して、デバイスを設定モードにします。
- SBSL モードへの変更を要求する CHANGE_BOOT_MODE (設定モード、3.8) の送信。成功すると、デバイスは SBSL モードで自動的に再起動します。
- FLASH_GET_SBSL_STATUS を介してデバイスの状態を確認します。
- FLASH_LOAD_DATA コマンドを介して新しい MCE イメージをプログラムします。
- FLASH_LOAD_CHECK_SIGNATURE コマンドを使用してプログラミングと検証を終了します。

次回の起動時に、デバイスは設定モードになり、パラメータセットおよびスクリプトをプログラムできます。

9.5 パラメータまたはスクリプトの更新の実行

パラメータ設定またはスクリプトの更新が必要な場合、これは次のように実行されます。

- デバイスを設定モードにする必要があります。実行中の MCE アプリケーションから、これは 6 アプリケーションモード- MCE 1.x コマンド設定で指定されているそれぞれの CHANGE_BOOT_MODE コマンドを使用して実行されます。
- 設定モードでは、パラメータ設定は、CLEAR_PARAMETER、DOWNLOAD_PARAMETER などのコマンド (3 設定モード - パラメータおよびスクリプトのプログラミング (MCE 1.x)) または CLEAR_DATA、DOWNLOAD_DATA コマンド (4 設定モード - パラメータおよびスクリプトのプログラミング (MCE 5.x)) をそれぞれ使用して、選択的に消去および書き込みができます。
- ダウンロードしたパラメータ設定の各ブロックの後で、プログラミングサイクルを完了するために CHECK_PARAMETER/CHECK_DATA コマンドを送信する必要があります。
- スクリプトの更新は、DOWNLOAD_PARAMETER/DOWNLOAD_DATA コマンドのそれぞれのバリアントを使用して行われます。
- スクリプトをダウンロードした後、CHECK_PARAMETER (スクリプト)/CHECK_DATA を要求する必要があります。

Note: *コマンド DOWNLOAD_PARAMETER と CHECK_PARAMETER は、パラメータ P1 と P2 の点から異なる syntax になります。 (3.9 および 3.10 を参照)*

ローダーファイル (*.ldf) が iMOTION™ツールで生成される場合、パラメータ設定とスクリプトの両方のそれぞれのコマンド (消去 – プログラム – チェック) の正しいシーケンスが既に含まれています。

参考資料

[1] iMOTION™ web サイト、http://www.infineon.com/iMOTION

[2] AN2020-07 iMOTION™製品とのインターフェース、アプリケーションノート

[3] iMOTION™モーションコントロールエンジン、ソフトウェアリファレンスマニュアル

[4] iMOTION™ソリューションデザイナー、PC ツール、
https://softwaretools.infineon.com/tools/com.ifx.tb.tool.imotionsolutiondesigner

[5] 追加の MCU を備えた iMOTION™ IMC300 モーターコントローラー、www.infineon.com/IMC300

[6] SWD に関する Arm®ドキュメント、https://developer.arm.com

[7] iMOTION™ Motion Control Engine, Functional Reference Manual

改訂履歴

版数	発行日	変更内容
1.3	2021-11-09	本版は英語版 AN2018-33 について、CYPRESS 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。
1.5	2023-03-29	本版は英語版 AN2018-33 Revision 1.5 を翻訳した日本語版 AN2018-33_JA Revision 1.5 です。
1.6	2025-04-25	本版は英語版 AN2018-33 Revision 1.6 を翻訳した日本語版 AN2018-33_JA Revision 1.6 です。

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

Edition 2025-04-25
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2025 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?
Go to **erratum@infineon.com**

Document reference
Revision 1.6

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。