



#### 概要

このアプリケーション・ノートでは、STM32 マイクロコントローラのブートローダで使用されている USB DFU プロトコルについて説明し、サポートされている各コマンドを詳しく取り上げています。

本書は、[www.st.com](http://www.st.com) に用意されているアプリケーション・ノート AN2606 「STM32 マイクロコントローラのシステム・メモリ・ブート・モード」で規定されている V3.x、V4.x、V7.x、V9.x、V10.x、および V13.x の各バージョンのブートローダを内蔵した STM32 製品を対象としています。これらの各製品を表 1 に示します。本書では、これらの製品を総称して STM32 としています。

デバイスのブートローダに対する USB ハードウェアのリソースと要件の詳細については、上記の AN2606 を参照してください。

表 1. 対象とする製品

タイプ	部品番号またはシリーズ名
マイクロ コントローラ	STM32F0 シリーズ : – STM32F04xxx、STM32F07xxx STM32F1 シリーズ : – STM32F105xx、STM32F107xx STM32F2 シリーズ STM32F3 シリーズ : – STM32F301xx、STM32F302xx、STM32F303xB、STM32F303xC、STM32F303xD、 STM32F303xE、STM32F373xx STM32F4 シリーズ : – STM32F401xx、STM32F405xx、STM32F407xx、STM32F411xx、STM32F412xx、 STM32F415xx、STM32F417xx、STM32F427xx、STM32F429xx、STM32F437xx、 STM32F439xx、STM32F446xx、STM32F469xx、STM32F479xx STM32F7 シリーズ : – STM32F722xx、STM32F723xx、STM32F732xx、STM32F733xx、STM32F745xx、 STM32F746xx、STM32F756xx、STM32F765xx、STM32F767xx、STM32F769xx、 STM32F777xx、STM32F779xx STM32L0 シリーズ : – STM32L07xxx、STM32L08xxx STM32L1 シリーズ : – STM32L1xxxC、STM32L1xxxD、STM32L1xxxE STM32L4 シリーズ : – STM32L431xx、STM32L432xx、STM32L433xx、STM32L442xx、STM32L443xx、 STM32L471xx、STM32L475xx、STM32L476xx、STM32L486xx、STM32L496xx、 STM32L4A6

## 目次

1	ブートローダのコード・シーケンス .....	5
2	USB DFU ブートローダの各種リクエスト .....	8
3	DFU ブートローダのコマンド .....	10
4	DFU_UPLOAD リクエストのコマンド .....	11
4.1	メモリの読み出し .....	11
4.2	取得コマンド .....	11
5	DFU_DNLOAD リクエストのコマンド .....	13
5.1	メモリの書込み .....	15
5.2	アドレス・ポインタの設定コマンド .....	16
5.3	消去コマンド .....	17
5.4	読み出し保護解除コマンド .....	18
5.5	DFU モードの終了 .....	19
6	ブートローダのプロトコル・バージョンの変遷 .....	21
7	改版履歴 .....	22

## 表の一覧

表 1.	対象とする製品 .....	1
表 2.	DFU クラスのリクエスト .....	8
表 3.	DFU クラス固有リクエストの概要 .....	8
表 4.	DFU ブートローダのコマンド .....	10
表 5.	ブートローダのプロトコルバージョン .....	21
表 6.	文書改版履歴 .....	22
表 7.	日本語版文書改版履歴 .....	22

## 図の一覧

図 1.	STM32 コネクティビティ・ライン・デバイスのブートローダ	6
図 2.	他の STM32 デバイスのブートローダ	7
図 3.	DFU_UPLOAD リクエスト：デバイス側	12
図 4.	DFU_UPLOAD リクエスト：ホスト側	12
図 5.	ダウンロード・リクエスト：デバイス側	13
図 6.	ダウンロード・リクエスト：ホスト側	14
図 7.	メモリの書込み：デバイス側	16
図 8.	アドレス・ポインタの設定コマンド：デバイス側	17
図 9.	消去コマンド：デバイス側	18
図 10.	読み出し保護解除コマンド：デバイス側	19
図 11.	DFU モードの終了操作：デバイス側	20

## 1 ブートローダのコード・シーケンス

ブートローダ DFU のバージョンが異なってもプロトコル（リクエストとコマンド）の面では相違がありません。詳しい相違点の一覧については、[セクション 6：ブートローダのプロトコル・バージョンの変遷](#)を参照してください。

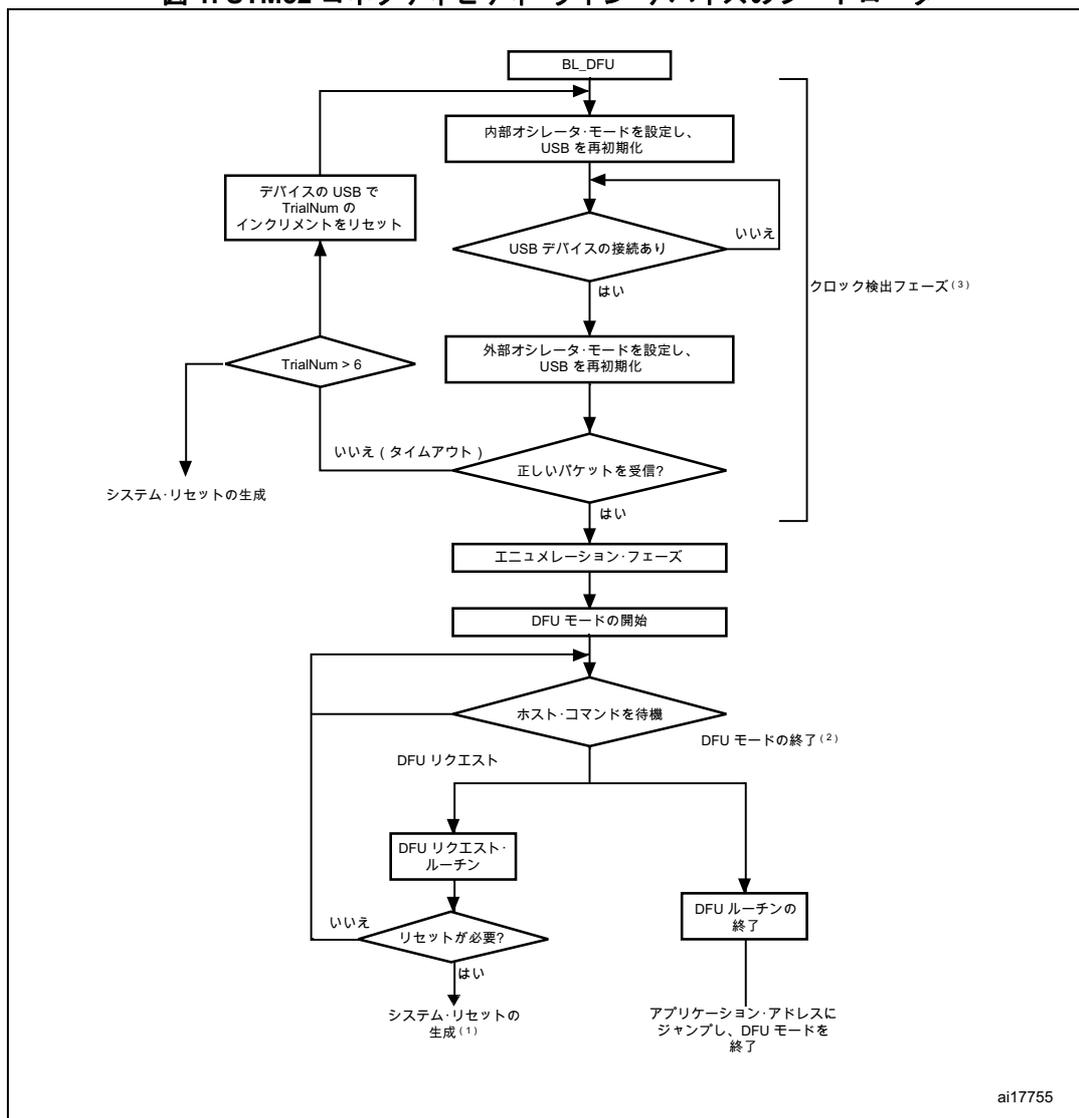
システム・メモリ・ブート・モードを開始し、STM32 デバイスを設定すると（詳細については AN2606 を参照）、ブートローダのコードでは USB とその割り込みを設定し、"エニュメレーション終了" 割り込みを待機します

USB ケーブルを接続すると、その直後に USB エニュメレーションが実行されます（USB ケーブルを接続済みであれば直ちに実行されます）。STM32 で USB DFU ブートローダ・アプリケーションが開始されないようにするには、USB ケーブルを取り外してからリセットします。

bcd デバイス・フィールドの MSB にあるデバイス・ディスクリプタにブートローダのバージョンが返されます（たとえば、0x2000 であればバージョン 2.0）。

コネクティビティ・ライン USB DFU ブートローダの場合、デバイスではまず 25 MHz での設定が試され、それが失敗した場合は 14.7456 MHz、次に 8 Hz での設定が試されます。それも失敗すると、長いタイムアウト値を使用してこの操作が繰り返されます（3 種類の設定が再度テストされます）。2 回目の試行が失敗するとシステム・リセットが生成されます。

図 1. STM32 コネクティビティ・ライン・デバイスのブートローダ

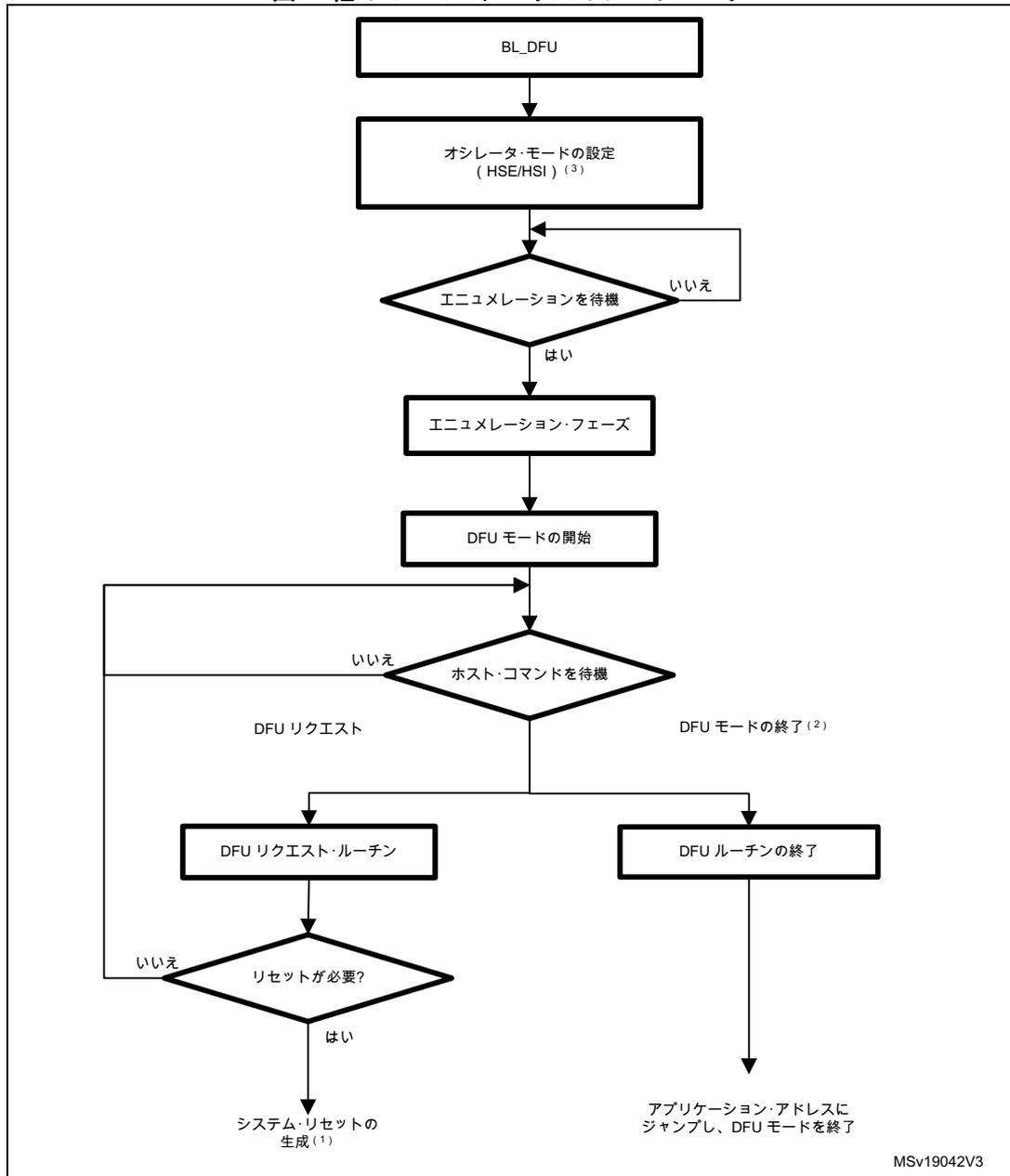


1. システムがリセットされると、接続状態とブート・ピンのステータスに応じて、デバイスでは BL\_DFU ループに戻るか、フラッシュ・メモリまたは RAM にあるコードを実行できます。
2. DFU モードの終了は、GetStatus リクエストとデバイスのリセットに続いて 0 データをダウンロードするリクエストを発行することで実行されます。
3. 6 回の試行の後 (3 回のクロック設定が 2 回テストされます)、システム・リセットが生成されます。

- USB 操作に HSE を使用している製品では以下の手順が実行されます (コネクティビティ・ラインを除く)。
  - 起動時に HSE が存在すればそのクロックが測定され、その値がサポートされていれば USB が設定されます。HSE が検出されないと、ブートローダではシステムのリセットを実行します。HSE クロックの測定値がサポート対象外の値であると、USB プロトコルが正しく機能しません。
- USB 操作に HSI を使用している製品では以下の手順が実行されます。
  - 起動時に HSI クロックを使用して USB が設定されます。

製品設定の詳細については AN2606 を参照してください。

図 2. 他の STM32 デバイスのブートローダ



1. システムがリセットされると、接続状態とブート・ピンのステータスに応じて、デバイスでは BL\_DFU ループに戻るか、フラッシュ・メモリまたは RAM にあるコードを実行できます。
2. DFU モードの終了は、GetStatus リクエストとデバイスのリセットに続いて 0 データをダウンロードするリクエストを発行することで実行されます。
3. 製品によっては、USB のブートローダの操作に外部オシレータの HSE を使用せず、内部オシレータの HSI を使用するものがあります。それぞれの製品でどのオシレータが必要であるかを把握するには、AN2606 製品のセクションを確認します。

**注：** ブートローダの起動時には、USB インタフェースのクロック・ソースとして内部オシレータ (HSI) が使用されます。USB イベントが検出されると、外部オシレータが USB のクロック・ソースとして設定されます。

## 2 USB DFU ブートローダの各種リクエスト

USB DFU ブートローダは、2004 年 8 月 5 日発効の "Universal Serial Bus Device Upgrade Specification for Device Firmware Upgrade" バージョン 1.1 に準拠した DFU プロトコルとリクエストをサポートしています。これらのリクエストの詳細については、上記の仕様を参照してください。

DFU クラス固有のリクエストとそのパラメータを表 2 と表 3 に挙げます。

表 2. DFU クラスのリクエスト

リクエスト	コード	説明
DFU_DETACH	0x00	DFU モードを終了してアプリケーションを開始することをデバイスに要求します。
DFU_DNLOAD	0x01	デバイス内部のフラッシュ・メモリにデータをロードするために、そのデータをホストからデバイスに転送することを要求します。消去コマンドも含まれています。
DFU_UPLOAD	0x02	デバイス内部のフラッシュ・メモリの内容をホストのファイルにロードするために、そのデータをデバイスからホストに転送することを要求します。
DFU_GETSTATUS	0x03	ホストにステータス・レポートを送信することをデバイスに要求します（最後のリクエスト実行で得られたステータスとそのリクエストの直後にデバイスで開始する状態も、このレポートの対象になります）。
DFU_CLRSTATUS	0x04	エラー・ステータスをクリアして次のステップに移動することをデバイスに要求します。
DFU_GETSTATE	0x05	このリクエストの直後にデバイスで開始する状態のみを送信することをデバイスに要求します。
DFU_ABORT	0x06	現在の状態と操作を終了して直ちにアイドル状態を開始することをデバイスに要求します。

注： ブートローダでは、Detach リクエストが意味を成しません。ブートローダは、ブート・モードの設定に応じ、システム・リセットによって起動するので、その時点でほかに実行されているアプリケーションは存在しません。

表 3. DFU クラス固有リクエストの概要

bmRequest	bRequest	wValue	wIndex	wLength	データ
00100001b	DFU_DETACH	wTimeout	インタフェース	ゼロ	なし
00100001b	DFU_DNLOAD	wBlockNum	インタフェース	長さ	ファームウェア
10100001b	DFU_UPLOAD	ゼロ	インタフェース	長さ	ファームウェア
00100001b	DFU_GETSTATUS	ゼロ	インタフェース	6	ステータス
00100001b	DFU_CLRSTATUS	ゼロ	インタフェース	ゼロ	なし
00100001b	DFU_GETSTATE	ゼロ	インタフェース	1	状態
00100001b	DFU_ABORT	ゼロ	インタフェース	ゼロ	なし

## 通信の安全性

ホストとデバイス間の通信は、CRC チェックや確認応答など、内蔵の USB 保護メカニズムによって確実に保護されています。転送されるデータやブートローダ固有のコマンドとデータに対しては、これ以上の保護は実行されません。

## 3 DFU ブートローダのコマンド

簡潔なメモリ書き込み操作とメモリ読み出し操作の実行では、主に DFU\_DNLOAD リクエストと DFU\_UPLOAD リクエストを使用します。これらのリクエストは、統合されたブートローダ・コマンド（書き込み、読み出し保護解除、消去、アドレスの設定など）の開始でも使用します。これらのリクエストに続き、DFU\_GETSTATUS コマンドによって、これらのコマンド実行がトリガされます。

DFU のダウンロード・リクエストでは、USB リクエストの構造にある **wValue** パラメータによって、実行するコマンドが選択されます。**wValue** が 0 であれば、このリクエストの後でホストから送信されるデータは、ブートローダのコマンド・コードです。先頭のバイトはコマンド・コードで、それ以降にバイトがあれば、そのコマンドに関連するデータです。

DFU のアップロード・リクエストでは、USB リクエストの構造にある **wValue** パラメータによって、実行するコマンドが選択されます。**wValue** が 0 であれば、取得コマンドが選択されて実行されます。

表 4. DFU ブートローダのコマンド

DFU リクエスト	ブートローダのコマンド	書き込み保護の無効化 読み出し保護の無効化	書き込み保護の有効化 読み出し保護の無効化	読み出し保護の有効化
DFU_UPLOAD	メモリの読み出し	可能	可能	不可能
	取得	可能	可能	可能
DFU_DNLOAD	メモリの書き込み	可能	可能 <sup>(1)</sup>	不可能
	消去	可能	可能 <sup>(1)</sup>	不可能
	読み出し保護解除	N/A <sup>(2)</sup>	N/A <sup>(2)</sup>	可能 <sup>(3)</sup>
	アドレス・ポインタの設定	可能	可能	可能
	DFU モードの終了	可能	可能	可能

- この操作は可能ですが、効果はありません。ブートローダからエラーは返されませんが、セクタが書き込み保護されているのでこの操作が実行されないからです。この扱いはフラッシュ・メモリにのみ該当します。RAM メモリやオプション・バイト領域は該当しません。
- 該当せず（この操作は可能ですが、メモリが保護されていないので意味がありません）。
- この場合は、フラッシュ・メモリ（0x0800 0000 から）と RAM の両方が消去されます。オプション・バイト領域はデフォルト値にリセットされます。

メモリ保護をしていない状態で、「読み出し保護解除」操作を実行すると、RAM メモリはその全体がブートローダのファームウェアによってクリアされますが、フラッシュ・メモリは消去されません（これは読み出し保護されていないからです）。

「書き込み保護」、「書き込み保護解除」、および「読み出し保護」の各操作に相当するコマンドはありません。これらの操作は、オプション・バイト領域に対するメモリの書き込みコマンドとメモリの読み出しコマンドで実行します。

## 4 DFU\_UPLOAD リクエストのコマンド

アップロードのリクエストを使用すると、さまざまなコマンドを実行できます。実行するコマンドは、USB リクエスト構造にある **wValue** パラメータの値で選択します。[セクション 4.1](#) から [セクション 5.5](#) で説明しているコマンドがサポートされています。

### 4.1 メモリの読み出し

**wValue** が 1 より大きい場合は、メモリの読み出し操作が選択されます。

ホストからデバイスに対して、内部フラッシュ・メモリ、内蔵 RAM、またはシステム・メモリの有効なメモリ・アドレス（注を参照）あるいはオプション・バイト領域から、指定のバイト数（**wLength**）のデータを送信するようにリクエストが発行されます。

**注：** 使用しているデバイスで有効なメモリ・アドレスの詳細については[セクション 4](#)を参照してください。

読み出すことができるバイト数は、メモリのターゲットに応じて以下のようになります。

- 内部フラッシュ・メモリ、内蔵 RAM、およびシステム・メモリ：2～2048 バイトのサイズを読み出すことができます。
- オプション・バイト領域：オプション・バイトのブロック・サイズに等しいサイズが読み出されます。
- 上記以外のメモリ場所：AN2606 の「重要な考慮事項」を参照してください。

ホストが要求するデータの読み出し位置となるアドレスは、wBlockNumber (**wValue**) の値とアドレス・ポインタを使用して次の式で計算します。

$$\text{アドレス} = ((\text{wBlockNum} - 2) \times \text{wTransferSize}) + \text{アドレス・ポインタ}$$

– wTransferSize は、要求されたデータ・バッファのサイズです。

このアドレス・ポインタは、アドレス・ポインタの設定コマンドで事前に指定しておく必要があります (DFU\_DNLOAD リクエストを使用します)。このアドレス・ポインタを事前に指定しないと、デバイス側では内部フラッシュの開始アドレス (0x08000000) から読み出すものと見なします。

フラッシュの読み出し保護が有効だと、この読み出し操作は実行されず、ターゲットとしたメモリ (内部フラッシュ・メモリ、内蔵 RAM、システム・メモリ、またはオプション・バイト領域) に関係なくステータスは dfuERROR、状態は errVENDOR として返されます。

### 4.2 取得コマンド

**wValue** 値が 0 である場合は、このコマンドが選択されます。

ブートローダでサポートされているコマンドの読み出しがホストから要求されます。このコマンドを受け取ったデバイスからは、そのコマンドのコードを収めた N バイトが返されます。

STM32 デバイスでは、以下のバイトが送信されます (N = 4)。

バイト 1:	0x00	取得コマンド
バイト 2:	0x21	アドレス・ポインタの設定
バイト 3:	0x41	消去
バイト 4:	0x92	読み出し保護解除

DFU\_UPLOAD コマンドの処理を [図 3](#) と [図 4](#) に示します。

図 3. DFU\_UPLOAD リクエスト：デバイス側

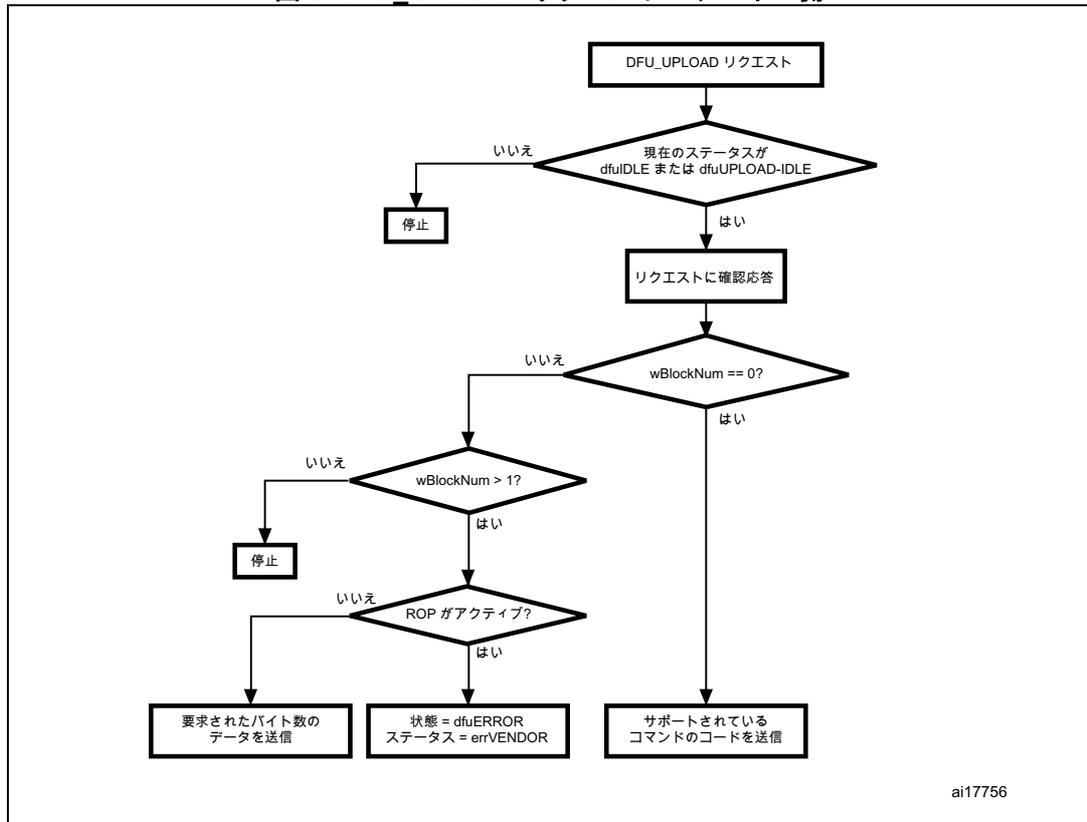
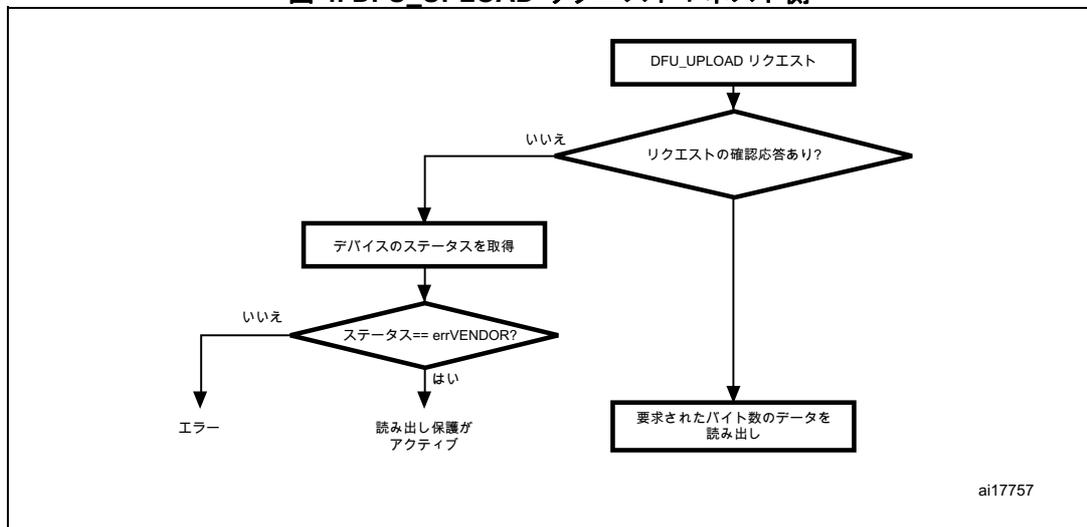


図 4. DFU\_UPLOAD リクエスト：ホスト側



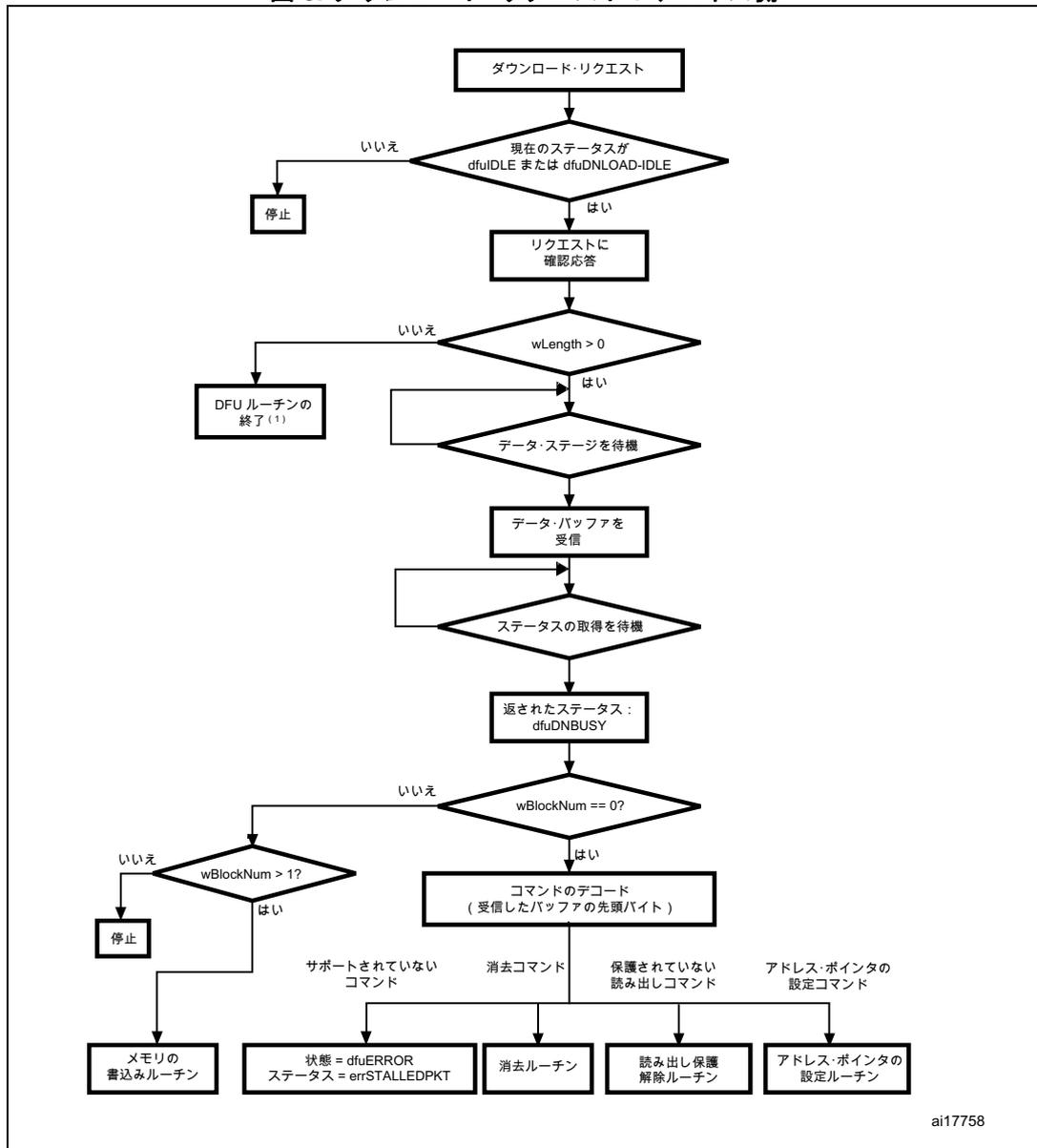
注： ホストでは、デバイスが正しい状態にあること（dfuIDLE または dfuUPLOAD-IDLE の状態）およびそのステータスにエラーが報告されていないことを確認したうえで、アップロードのリクエストを発行する必要があります。デバイスがその状態にない場合、ホストではすべてのエラーをクリアし（DFU\_CLRSTATUS リクエスト）、デバイスが dfuIDLE 状態に戻るまで新しいステータスを取得し続ける必要があります。

## 5 DFU\_DNLOAD リクエストのコマンド

ダウンロードのリクエストを使用してさまざまなコマンドを実行します。実行するコマンドは、USB リクエスト構造にある **wValue** パラメータの値で選択します。以下の操作がサポートされています。

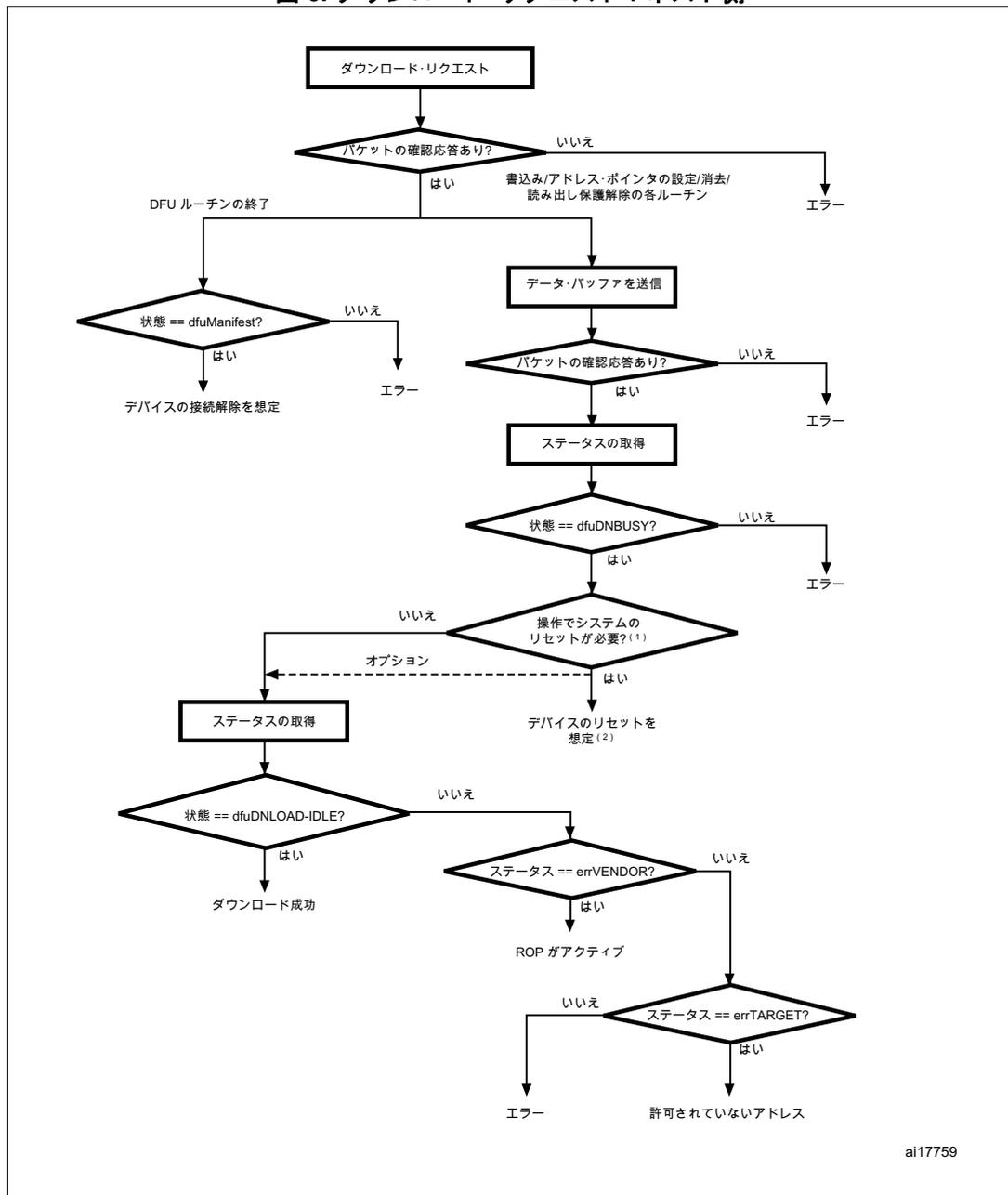
- メモリの書込み (**wValue** が 1 より大きい場合)
- アドレス・ポイントの設定 (**wValue** が 0 で、先頭バイトが 0x21 の場合)
- 消去 (**wValue** が 0 で、先頭バイトが 0x41 の場合)
- 読み出し保護解除 (**wValue** が 0 で、先頭バイトが 0x92 の場合)
- DFU の終了 (DFU モードを終了してアプリケーションにジャンプ)

図 5. ダウンロード・リクエスト：デバイス側



1. このルーチンを使用して、デバイスのリセットまたはアプリケーションへのジャンプが可能です。

図 6. ダウンロード・リクエスト：ホスト側



1. システムリセットを必要とする操作として、読み出し保護解除のコマンドおよびオプション・バイト領域への書き込み操作があります。
2. dfuDNBUSY 状態に戻ったデバイスでは、要求した操作が実行され、システムがリセットされます。ホストでは次回のエニュメレーションを待機するにとどめるほか、再度ステータスを取得することもできますが、要求した操作の実行にデバイスが失敗するまで、デバイスからは応答が得られません。

**注：** ホストでは、デバイスが正しい状態にあり (dfuIDLE または dfuDNLOAD-IDLE の状態) およびそのステータスにエラーが報告されていないことを確認したうえで、ダウンロードのリクエストを発行する必要があります。デバイスがその状態にない場合、ホストではすべてのエラーをクリアし (DFU\_CLRSTATUS リクエスト)、デバイスが dfuIDLE 状態に戻るまでステータスを再度取得し続ける必要があります。

## 5.1 メモリの書込み

wValue が 1 より大きい場合は、メモリの書込み操作が選択されます。

ホストからデバイスに対して、指定のバイト数 (wLength) のデータを受信し、それを内部フラッシュ・メモリ、内蔵 RAM、またはオプション・バイト領域の有効なメモリ・アドレス (注を参照) にロードするようにリクエストが発行されます。

**注：** 使用しているデバイスで有効なメモリ・アドレスの詳細については [セクション 4 : DFU\\_UPLOAD リクエストのコマンド](#) を参照してください。

書き込むことができるバイト数は、メモリのターゲットに応じて以下のようになります。

- 内部フラッシュ・メモリおよび内蔵 RAM : 2 ~ 2048 バイトのサイズを書き込むことができます。
- オプション・バイト領域 : オプション・バイトのブロック・サイズに等しいサイズが書き込まれます。
- 上記以外のメモリ場所 : AN2606 に記述された製品に応じて「重要な考慮事項」を参照してください。

**注：** オプション・バイト領域には、そのブロック・サイズと異なるサイズの書き込みも可能ですが、データの整合性を確保するために、一度にブロック全体に書き込むことをお勧めします。ターゲットがオプション・バイト領域である場合は、アドレス・ポインタを必ずそのオプション・バイトの開始アドレスとする必要があります。そのようにしないと、リクエストが実行されません。

メモリの書込み操作は、ホストから DFU\_GETSTATUS リクエストが発行された場合にのみ、有効に実行されます。デバイスから dfuDNBUSY 以外のステータスが返された場合は、エラーが発生しています。

ターゲットがオプション・バイト領域である場合を除き、もう一度 DFU\_GETSTATUS リクエストを発行して、コマンドが正しく実行されたことを確認する必要があります。オプション・バイト領域をターゲットとした場合は、書き込み操作の完了後、デバイスが直ちにリセットされます。デバイスが受け取ったアドレスが正しくない場合またはサポートされていない場合、そのデバイスではステータスが dfuERROR、状態が errTARGET になります。

ホストが要求するデータの書込み位置となるアドレスは、wBlockNumber (wValue) の値とアドレス・ポインタを使用して、アップロード・リクエスト同様に次の式で計算します。

$$\text{アドレス} = ((\text{wBlockNum} - 2) \times \text{wTransferSize}) + \text{アドレス・ポインタ}$$

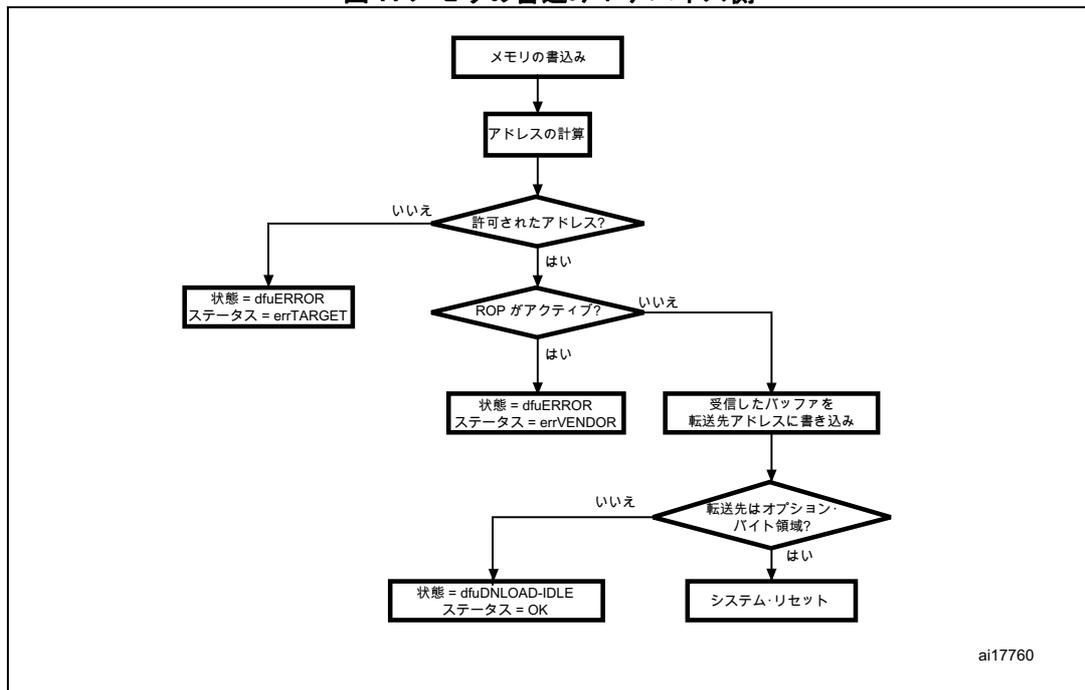
- wTransferSize : ホストから送信されたデータ・バッファの長さ
- wBlockNumber : wValue パラメータの値

フラッシュの読み出し保護が有効であると、このメモリの書込み操作は実行されず、ターゲットとしたメモリ (内部フラッシュ・メモリ、内蔵 RAM、またはオプション・バイト領域) にかかわらずステータスは dfuERROR、状態は errVENDOR として返されます。

オプション・バイト領域に対してメモリの書込みコマンドを発行すると、すべてのオプションが消去されてから新しい値が書き込まれます。オプション・バイトの新しい設定を反映するために、このコマンドの最後でブートローダによってシステム・リセットが発行されます。

- 注：**
- 1 RAM に書き込む場合は、ブートローダのファームウェアで使用されている RAM メモリの先頭部分に、データの書込み先が重ならないように注意する必要があります。
  - 2 書込み保護されているセクタに対して書込み操作を実行しても、エラーは返されません。

図 7. メモリの書込み：デバイス側



ai17760

## 5.2 アドレス・ポインタの設定コマンド

wValue が 0 で、ホストから送信されたバッファの先頭バイトが 0x21 である場合、アドレス・ポインタの設定コマンドが選択されます。このバッファの長さは 5 バイトで、先頭バイトを除く各バイトは、2 番目のバイトを LSB とするアドレス・バイトです (32 ビット・アドレス・フォーマット)。

このパラメータを指定して、ホストから DFU\_DNLOAD リクエストが送信されます。このリクエストによって、メモリの読み出し操作とメモリの書込み操作の開始アドレスを計算する際に使用するアドレス・ポインタ値が設定されます。

STM32 デバイスでは、以下のバイトを受信します。

- バイト 1: 0x21 アドレス・ポインタの設定コマンド
- バイト 2: A[7:0] アドレス・ポインタの LSB
- バイト 3: A[15:8] アドレス・ポインタの 2 番目のバイト
- バイト 4: A[22:16] アドレス・ポインタの 3 番目のバイト
- バイト 4: A[31:23] アドレス・ポインタの MSB

アドレス・ポインタの設定コマンドを送信したホストでは、つづいて DFU\_GETSTATUS リクエストを送信する必要があります。

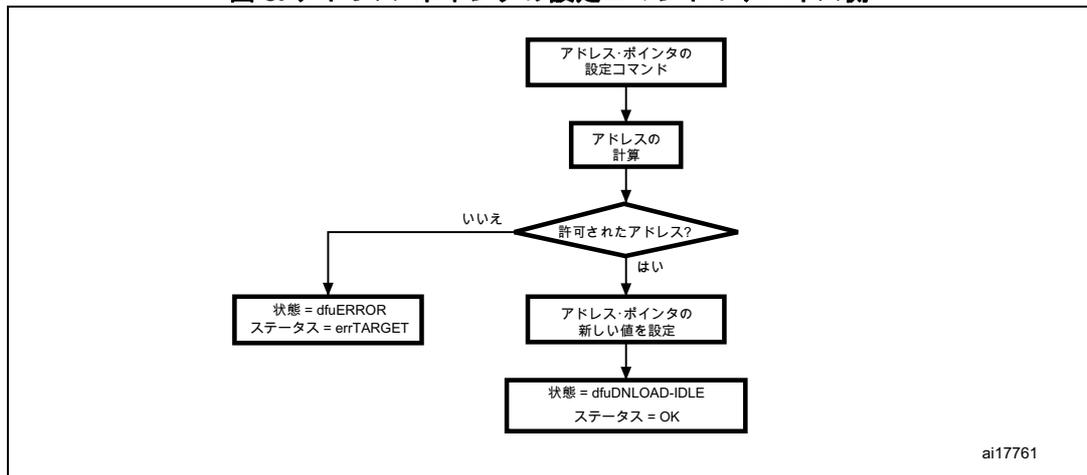
アドレス・ポインタの設定コマンドは、ホストから DFU\_GETSTATUS リクエストが発行された場合にのみ、有効に実行されます。デバイスから dfuDNBUSY 以外のステータスが返された場合は、エラーが発生しています。

コマンドが正しく実行されたかどうかを確認するために、2 番目の DFU\_GETSTATUS リクエストが必要です。デバイスが受け取ったアドレスが正しくない場合またはサポートされていない場合、そのデバイスではステータスが dfuERROR、状態が errTARGET になります。

アドレス・ポインタの値として指定できる場所は、内部フラッシュ・メモリ、内蔵 RAM、システム・メモリ、オプション・バイト領域の有効なメモリ・アドレス（注を参照）です。

- 注：
- 1 使用しているデバイスで有効なメモリ・アドレスの詳細については[セクション 4 : DFU\\_UPLOAD リクエストのコマンド](#)を参照してください。
  - 2 フラッシュの読み出し保護が有効または無効の場合に、アドレス・ポインタの設定コマンドを指定し、実行できます。

図 8. アドレス・ポインタの設定コマンド：デバイス側



## 5.3 消去コマンド

wValue が 0 で、ホストから送信されたバッファの先頭バイトが 0x41 である場合は消去コマンドが選択されます。ページを消去する操作では、バッファの長さを 5 バイトにすることができます（先頭バイトを除くバイトは、2 番目のバイトを LSB とするアドレス・バイトです）。また、全体を消去する操作では、長さが 1 バイトのバッファ（コマンド・バイトのみのバッファ）とすることができます。

このパラメータを指定して、ホストから DFU\_DNLOAD リクエストが送信されます。このリクエストによって、内部フラッシュ・メモリの 1 ページまたは全体が消去されます。

デバイスでは、以下のバイトを受信します（ページの消去の場合）。

- バイト 1: 0x41 消去コマンド
- バイト 2: A[7:0] ページ・アドレスの LSB
- バイト 3: A[15:8] ページ・アドレスの 2 番目のバイト
- バイト 4: A[22:16] ページ・アドレスの 3 番目のバイト
- バイト 4: A[31:23] ページ・アドレスの MSB

または、次のように 1 バイトのコマンドを受信します。

STM32 デバイスは、以下のバイトを受信します（ページ全体の消去の場合）。

- バイト 1: 0x41 消去コマンド

消去コマンドを送信したホストでは、つづいて DFU\_GETSTATUS リクエストを送信する必要があります。

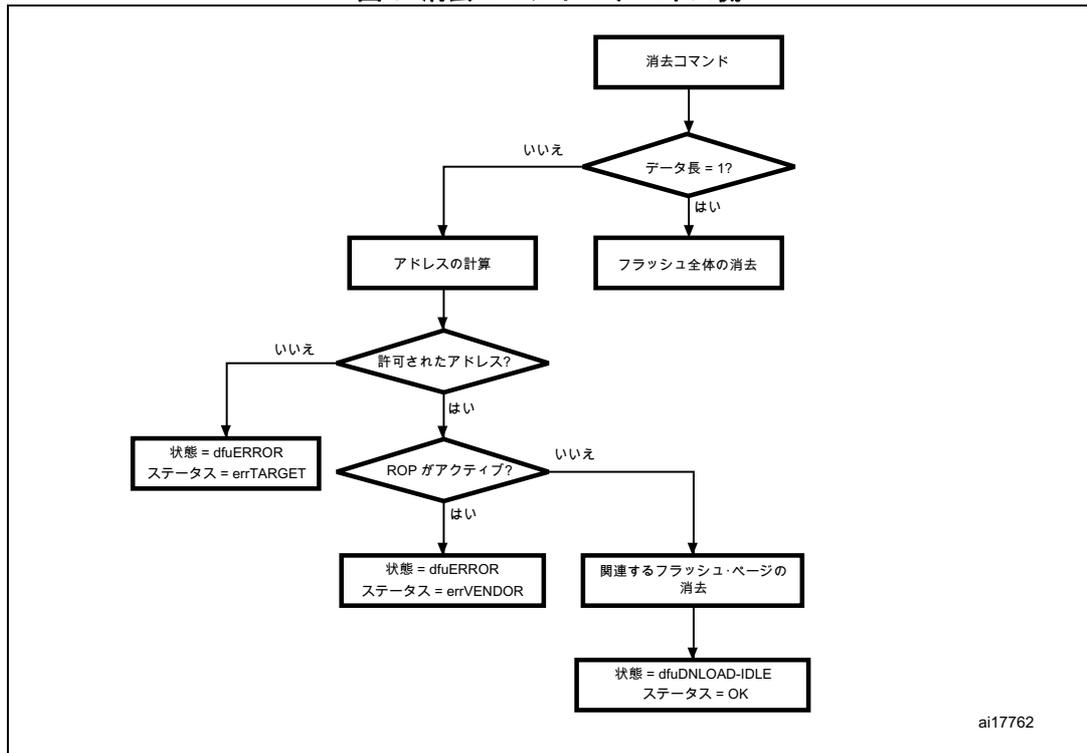
消去コマンドは、ホストから DFU\_GETSTATUS リクエストが発行された場合にのみ、有効に実行されます。デバイスから dfuDNBUSY 以外のステータスが返された場合は、エラーが発生しています。

コマンドが正しく実行されたかどうかを確認するために、2 番目の DFU\_GETSTATUS リクエストが必要です。デバイスが受け取ったページ・アドレスが正しくない場合またはサポートされていない場合、そのデバイスではステータスが dfuERROR、状態が errTARGET になります。フラッシュの読み出し保護がアクティブな場合、そのデバイスではステータスが dfuERROR、状態が errVENDOR となり、消去操作は無視されます。

消去するページのアドレスとして指定できるのは、フラッシュ・メモリのアドレスです。

注： 書込み保護されているセクタに対して消去操作を実行しても、エラーは返されません。

図 9. 消去コマンド：デバイス側



## 5.4 読み出し保護解除コマンド

wValue が 0 で、ホストから送信されたバッファの先頭バイトが 0x92 である場合は、読み出し保護解除コマンドが選択されます。バッファの長さは 1 バイトのみとします（コマンド・バイトのみ）。

このパラメータを指定して、ホストから DFU\_DNLOAD リクエストが送信されます。このリクエストによって、内部フラッシュ・メモリの読み出し保護が解除されます。

デバイスでは、以下のバイトを受信します。

バイト 1: 0x92 読み出し保護解除コマンド

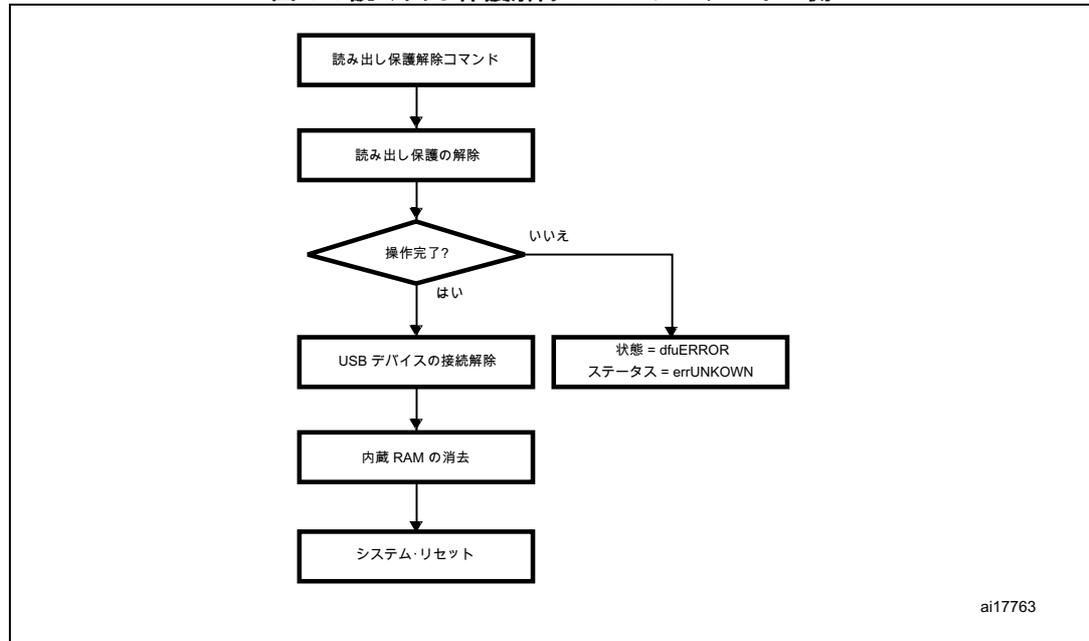
読み出し保護解除コマンドを送信したホストでは、つづいて DFU\_GETSTATUS リクエストを送信する必要があります。

読み出し保護解除コマンドは、ホストから DFU\_GETSTATUS リクエストが発行された場合にのみ、有効に実行されます。デバイスから dfuDNBUSY 以外のステータスが返された場合は、エラーが発生しています。この操作の後、デバイスでは読み出し保護が解除され、その結果、内部フラッシュ・メモリと内蔵 RAM の全体が消去されます。

これにより、このコマンドを実行すると、デバイスではその接続が切断され、システムのリセットが実行されます。この状態では、2 番目の状態取得リクエストに対してデバイスは応答できません。したがって、ホストでは、デバイスが再度エニューメレーションされるまで待機する必要があります。

コマンドが正しく実行されたかどうかを確認するために、(デバイスがまだ接続されているときに) 2 番目の DFU\_GETSTATUS リクエストを発行することもできます。デバイスでこのコマンドの実行に失敗すると、そのエラーのタイプに応じたエラー・ステータスがデバイスから返されます。

図 10. 読み出し保護解除コマンド：デバイス側



ai17763

## 5.5 DFU モードの終了

DFU のダウンロード・リクエストを使用して、DFU モード（およびブートローダ）を終了し、内部フラッシュまたは内蔵 RAM にロード済みのアプリケーションにジャンプできます。

ホストからは、DFU モードを終了する必要があることをデバイスに通知するために、データ長が 0 (リクエストの後に続くデータ・ステージがないリクエスト) の DFU\_DNLOAD リクエストが送信されます。デバイスでは、その現在の状態が dfuDNLOAD-IDLE または dfuIDLE であれば、このリクエストに対して確認応答します。

DFU モードの終了操作は、ホストから DFU\_GETSTATUS リクエストが発行された場合にのみ、有効に実行されます。デバイスから dfuMANIFEST 以外のステータスが返された場合は、エラーが発生しています。この操作の後、デバイスでは以下のアクションが実行されます。

- デバイスが自身を接続から切り離します。
- ブートローダで使用されているペリフェラルのレジスタを、そのデフォルトのリセット値に初期化します。
- ユーザ・アプリケーションのメイン・スタック・ポインタを初期化します。
- 受け取ったアドレス・ポインタに 4 を加算したアドレスにプログラムされているメモリの場所にジャンプします。この場所は、アプリケーションのリセット・ハンドラのアドレスに相当します。たとえば、受け取ったアドレスが 0x0800 0000 であれば、アドレス 0x0800 0004 にプログラムされているメモリの場所にブートローダがジャンプします。

一般的には、アプリケーションのジャンプ先をプログラムしたベース・アドレスをホストから送信する必要があります。

DFU の終了ルーチンを開始する前に、アドレス・ポインタの設定コマンドを使用して、このアドレス・ポインタを設定しておく必要があります。そのようにしないと、ブートローダは内部フラッシュ・メモリのデフォルト・アドレスである 0x08000000 にジャンプします。

最後のメモリの書き込み操作でアドレス・ポインタを設定することもできます。ダウンロード操作を実行すると、そのダウンロードで使用したアドレス・ポインタが格納され、以降のジャンプで使用されます。

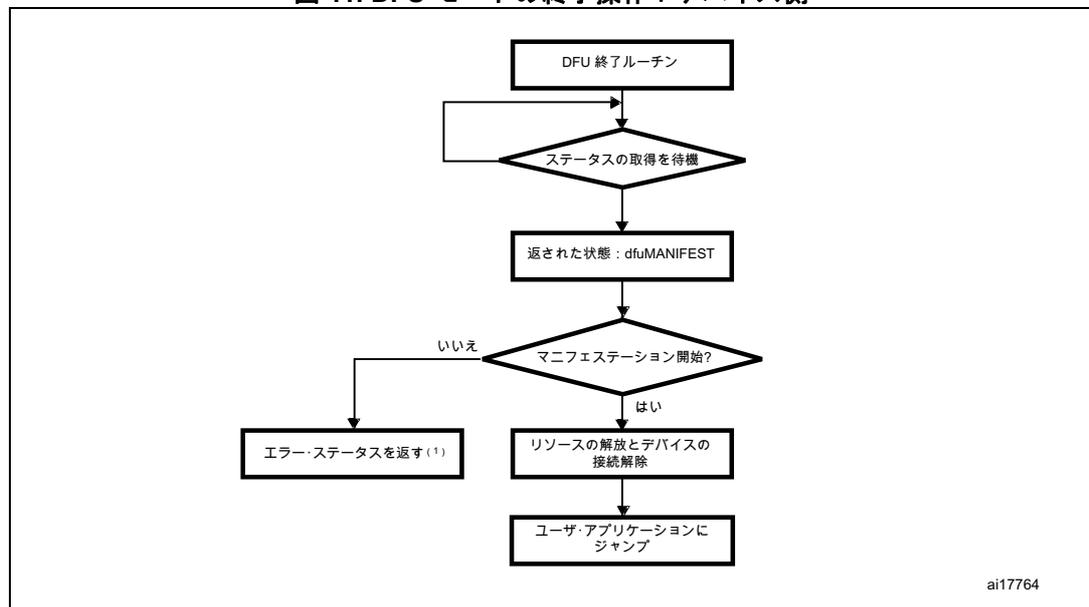
**注：** アドレス・ポインタが指しているアドレスに実行可能コードが存在しない場合、デバイスはリセットされ、ブート・ピンの状態に応じて、再度ブートローダ・モードが開始されます。

ブートローダの DFU アプリケーションはマニフェステーショントレラントではないので、マニフェステーション・フェーズの完了後、デバイスはホストからのリクエストに応答できなくなります。

コマンドが正しく実行されたかどうかを確認するために、(デバイスがまだ接続されているときに) 2 番目の DFU\_GETSTATUS リクエストを発行することもできます。デバイスでこのコマンドの実行に失敗すると、そのエラーのタイプに応じたエラー・ステータスがデバイスから返されます。

- 注：**
- 1 ユーザ・アプリケーションのアドレスを正しく指すように、そのアプリケーションでベクタ・テーブルを正しく設定している場合にのみ、アプリケーションへのジャンプが機能します。
  - 2 USB IP を使用して、ロード済みアプリケーション・コードにブートローダからジャンプする場合、保留中のすべての USB 割り込みをアプリケーションから無効にする必要があります。割り込みを有効にするにはアプリケーションからコアをリセットする必要があります。そのようにしないと、ブートローダのコードから発行されて保留中の割り込みがユーザ・コードと干渉し、機能上の障害の原因となることがあります。システム・メモリ・ブート・モードを終了した後であれば、この手順は不要です。

図 11. DFU モードの終了操作：デバイス側



1. このステータスは、エラーの発生元と現在のステータスに応じて異なります。

## 6 ブートローダのプロトコル・バージョンの変遷

表 5 にブートローダの各バージョンを示します。

表 5. ブートローダのプロトコル・バージョン

バージョン	説明
V2.0	最初のブートローダ・バージョン。
V2.1	DFU ブートローダ・バージョン V2.1。バージョン V2.0 と異なり、このバージョンでは、OTP メモリ・インタフェースやデバイス機能インタフェースなどの拡張インタフェース・ディスクリプタを採用しています。 V2.0 と V2.1 はさまざまなデバイスに実装されています。使用しているデバイスにどのバージョンが実装されているかを把握するには AN2606 を参照してください。 妥当なタイムアウトを使用してデータ・メモリのタイミングで書き込む際に見られるバグを修正しました。
V2.2	サポート対象の操作を、読み出し、書込み、消去の各操作ではなく、読み出しと書込みのみとするように、オプション・バイト、OTP、およびデバイス機能を更新しました。

## 7 改版履歴

表 6. 文書改版履歴

日付	版	変更内容
2010年3月9日	1	初版発行
2011年4月15日	2	ブートローダ・バージョン V2.0 と V2.1 を導入し、 <b>セクション 1</b> のブートローダ・シーケンスの説明を更新。 <b>図 2: 他の STM32 デバイスのブートローダ</b> を追加。 <b>セクション 4.1: メモリの読み出し</b> で、オプション・バイト領域から読み出すことができるバイト数を更新し、他のメモリの場所を追加。 <b>セクション 5.1: メモリの書込み</b> で、オプション・バイト領域に書き込むことができるバイト数を更新し、他のメモリの場所を追加。 <b>セクション 6</b> で、ブートローダ V2.1 を追加。
2013年2月12日	3	<b>図 1: STM32 コネクティビティ・ライン・デバイスのブートローダ</b> のタイトルを変更。 <b>図 2: 他の STM32 デバイスのブートローダ</b> で、タイトルなどを更新。 <b>図 2</b> の下に <b>注:</b> を追加。 <b>表 1: 対象とする製品</b> を追加。
2014年4月30日	4	<b>表 1: 対象とする製品</b> および <b>表 5: ブートローダのプロトコル・バージョン</b> を更新。 <b>セクション 1: ブートローダのコード・シーケンス</b> を更新。 デバイス依存のブートローダ・パラメータ専用の <b>セクション</b> を削除。 <b>図 2: 他の STM32 デバイスのブートローダ</b> を更新して <b>3.</b> を追加
2016年10月21日	5	<b>概要</b> および <b>表 1: 対象とする製品</b> を更新。
2017年3月16日	6	<b>表 1: 対象とする製品</b> を更新。

表 7. 日本語版文書改版履歴

日付	版	変更内容
2018年6月	1	日本語版 初版発行

**重要なお知らせ（よくお読み下さい）**

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST製品は、注文請書発行時点で有効なSTの販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してSTは一切の責任を負いません。

明示又は黙示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

STおよびSTロゴはSTMicroelectronicsの商標です。その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

この資料は、STMicroelectronics NV並びにその子会社(以下ST)が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くためにSTマイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST及びSTマイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

© 2018 STMicroelectronics - All rights reserved