AN5185

アプリケーション・ノート

STM32WB シリーズ向け ST ファームウェア・アップグレード・サービス

概要

本書では、STM32WB シリーズマイクロコントローラで使用可能なファームウェアアップグレードサービス（FUS）について説明しています。これらのサービスは、内蔵 Flash メモリのセキュア部分にある ST マイクロエレクトロニクス コードによって提供され、ユーザ Flash メモリで Cortex®-M4 上で実行されるコードによって、または内蔵ブートローダコマンド（Cortex®-M4 上でも動作する）によって使用されます。

1 一般情報

本書は、STM32WB シリーズの Arm® ベースのデバイスに適用されます。

注 Arm は、米国内およびその他の地域にある Arm Limited（またはその子会社）の登録商標です。

arm

1.1 ファームウェアアップグレードサービスの定義

FUS（ファームウェアアップグレードサービス）は、STM32WB Cortex®-M0+ で動作するファームウェアであり、次の機能があります。

1. STM32WB Cortex®-M0+ ワイヤレススタックのインストール、アップグレード、または削除：
   - ST マイクロエレクトロニクス によってのみ暗号化および署名されます。
   - オプションで、必要に応じて顧客によって追加で二重署名されます。
2. FUS セルフアップグレード：
   - ST マイクロエレクトロニクス によってのみ暗号化および署名されます。
   - オプションで、必要に応じて顧客によって追加で二重署名されます。
3. 顧客認証キーの管理：
   - イメージの二重署名に使用されます。
   - 顧客認証キーのインストール、更新、およびロック
4. ユーザキーの管理：
   - 顧客キーを格納します。
     - マスタキー
     - シンプルなクリアキー
     - 暗号化キー（マスタキーによる）
     - Cortex®-M0+ コードでのみアクセス可能なセキュア領域内にある
   - 格納されたキー（シンプルまたは暗号化）をセキュアモード（Cortex®-M4 はキーにアクセスできない）で AES1（高度暗号化標準）に書き込みます。
   - 格納されたキーをロックして、次のシステムリセットまで使用できないようにします。
   - 他のアプリケーションが使用しないよう、以前にロードしたキーを AES からアンロードします。
   - キー幅：128 または 256 ビット
   - 最大 100 個のユーザキー（マスタキーによる暗号化またはクリア）と 1 個のユーザマスタキー。
5. Cortex®-M4（ユーザコードまたはブートローダ）との通信：
   - IPCC コマンドおよびレスポンスモデル（無線スタックモデルと同じ）を経由します。
   - STM32WB ブートローダ（ROM）によってすでにサポートされているコマンド。

1.2 FUS のバージョン管理と識別

ユーザは、セクション 1.6 共有テーブルのメモリ使用量 および セクション 6.1 共有テーブルの使用方法 で説明しているように、FUS バージョンを識別するために SRAM2a の共有テーブルのメモリを読み出す必要があります。

IPCCDBA オプションバイト によって示された SRAM2a の最初のワードは、「デバイス情報テーブル」のアドレスです。このテーブル（表 7. デバイス情報テーブルで説明）には、4 バイトでエンコードされたオフセット 0xC の FUS バージョンが含まれています。通常、IPCCDBA=0x0000 で、@0x20030000 に 0x20030024 が含まれている場合、FUS のバージョンは @0x20030030 です。

FUS イメージのインストールは、イメージバイナリリリースノートに記載されている条件に従う必要があります。

注 V2.7.0 より古い STM32CubeProgrammer（STM32CubeProg）で SWD インタフェースを使用する場合、デバイス情報テーブルのアドレスは 0x20030890 に配置されます。STM32CubeProgrammer V2.7.0 以降では、デバイス情報テーブルは 0x20030024 にあります。

表 1. FUS バージョン

FUS バージョン	説明
V0.5.3	すべての STM32WB5xx デバイス用に、生産時にプログラムされたデフォルトバージョン。 STM32WB5xG デバイスでは V1.0.1 に、また STM32WB5xE/5xC デバイスでは V1.0.2 にアップグレードする必要があります。 このバージョンは www.st.com からはダウンロードできず、またユーザがインストールすることもできません。
V1.0.1	www.st.com から入手可能な初の公式リリースで、STM32WB5xG デバイス専用（1 MB の Flash メモリサイズ）。 このバージョンを STM32WB5xE/5xC デバイスにインストールしないでください。インストールすると、デバイスがロック状態になり、追加の更新ができなくなります。
V1.0.2	www.st.com から入手可能な初の公式リリースで、STM32WB5xE/5xC デバイス専用（512 KB および 256 KB の Flash メモリサイズ）。 STM32WB5xG デバイスに V0.5.3 がある場合は、デバイスの V1.0.2 を使用します。 STM32WB5xG デバイスに FUS V1.0.1 がある場合、V1.0.2 にアップグレードする必要はありません。V1.0.1 に対する新機能や変更はありません。 FUS V1.0.2 のインストールが、FUS V1.0.1 を搭載した STM32WB5xG デバイスでユーザによって開始される場合、FUS は FUS_STATE_IMG_NOT_AUTHENTIC エラーを返し、アップグレードを破棄します。
V1.1.0	FUS は、次の機能をサポートするために更新されます。 • ユーザがワイヤレススタック上でアンチロールバックを有効化できる、FUS_ACTIVATE_ANTIROLLBACK コマンドを追加します。 この機能を有効化して、古いワイヤレススタックがインストールされないようにすることができます。 • セーフブートを V1.1.0 バージョンで置き換えます（工場リセットによる完全なチップロックの交換）。 • Flash ECC、Flash 破損、または オプションバイト 破損エラーの場合に備えて工場リセットを追加します。 工場リセットとは、ワイヤレススタックが存在する場合はこれを消去し、FUS 時にリブートして、他のユーザセクタを完全に消去することを意味します。 FUS V1.1.0 は、V1.0.1 または V1.0.2 FUS を含むデバイスにのみインストールできます。 デバイスに V0.5.3 がインストールされている場合、V1.0.2 を最初にインストールし、次に V1.1.0 をインストールする必要があります。 FUS V1.1.0 を FUS V0.5.3 の後にインストールすると、FUS_STATE_IMAGE_NOT_AUTHENTIC エラーが発生し、アップグレードが破棄されます。
V1.1.1	STM32WB5xx 640KB 販売タイプをサポートするための FUS の更新。 このバージョンは www.st.com では入手できず、アップグレードに使用できません。 このバージョンは V1.1.0 との全面的な互換性があります。640 KB の新しい販売タイプで管理されること以外に違いはありません。
V1.1.2	FUS は次のために更新されます。 • Flash の使用方法の最適化：これにより、以前にインストールしたスタックの下に 1 セクタの分離を維持したまま、スタックをインストールできます（セクション 2 ワイヤレススタックイメージの操作で説明されたスタックサイズの空間の制約の代わり）。 • セキュリティの強化

FUS バージョン	説明
	FUS V1.1.0 から FUS V1.1.2 へアップグレードするには、アンチロールバック を最初に有効化する必要があります。アンチロールバックを有効化する前に、インストールされたワイヤレススタックが存在する必要があります。 V1.1.0 から V1.2.0 へのアップグレードは、制約やユーザからの追加操作なしで実行できます。
V1.2.0	FUS は次のために更新されます。 • V1.1.2 の生産時の FUS アップデートが含まれます。 • FUS V1.1.0 から FUS V1.2.0 への直接更新を、アンチロールバック を有効化せずに可能にします。 • FUS V0.5.3 から FUS V1.2.0 への直接更新を可能にします（中間の FUS バージョンのインストールなし）。 • セキュリティ更新 FUS V1.1.0 またはその他の FUS バージョンから FUS V1.2.0 へのアップグレードは、制約やユーザ操作なしで実行できます。

次の表に、FUS バージョンの互換性オプション（あるバージョンから別のバージョンにアップグレードできる場合）の詳細を示します。FUS V1.2.0 は、前のバージョンからのアップグレードが可能なバージョンです。これは、2 つのバイナリでリリースされます。

- `stm32wb5x_fus_fw_V1.2.0.bin`:任意の FUS バージョン V1.xy からのアップグレード
- `stm32wb5x_fus_fw_V1.2.0_for_V0.5.3.bin`:FUS バージョン V0.5.3 からのアップグレード

注 STM32WB10xx および STM32WB15xx には FUS V1.2.0 のみが搭載されています。これは、STM32WB5xxx FUS V1.2.0 との全面的な互換性がありますが、ユーザキーサービスは提供していません。

表 2. FUS バージョンの互換性

アップグレード		アップグレード先					
		V0.5.3	V1.0.2	V1.1.0	V1.1.1	V1.1.2	V1.2.0
アップグレード元	V0.5.3	X	√	X	X	X	√
	V1.0.2	X	X	√	X	√	√
	V1.1.0	X	X	X	X	(1)	√
	V1.1.1	X	X	X	X	√	√
	V1.1.2	X	X	X	X	√	√
	V1.2.0	X	X	X	X	X	√

凡例:
- X:アップグレード不可
- √:アップグレード可能
- X:アップグレード禁止。アップグレードすると暗号化キーが失われます。
- (1):アップグレード可能。ただし、最初に BLE スタックをインストールして、アンチロールバックを有効にする必要があります。

FUS バージョンは、2 つの異なるソースから入手できます。

- STM32WB シリーズデバイスは、ST マイクロエレクトロニクス によって生産フェーズで直接プログラムされます。
- www.st.com から入手できます。この方法は、主に FUS のバージョンアップグレードプロセスで使用されます。

次の表に、生産時および www.st.com で各バージョンが入手可能かどうかの詳細を示します。

表 3. 使用可能な FUS バージョン

FUS バージョン	生産時	www.st.com のバイナリ
V0.5.3	√	X
V1.0.2	√	√
V1.1.0	√	√
V1.1.1	√	X
V1.1.2	X	√
V1.2.0	√	√
凡例: • X:利用不可 • √:利用可能		

1.3 FUS を有効化する方法

FUS は Cortex®-M0+ および FUS およびワイヤレススタック専用の保護された Flash メモリゾーンで動作します。この場合、次の 2 つの状況が考えられます。

表 4. FUS 有効化事例

状況	FUS を有効化する方法
動作中のワイヤレススタックがない(たとえば、STM32WB シリーズデバイスを初めて実行している場合、またはワイヤレススタックが削除された場合)	PWR_CR4 レジスタの C2BOOT ビットをセットして、Cortex®-M0+ が有効化されていることを確認します。 IPCCDBA(オプションバイト)が SRAM2a の有効な共有テーブル情報構造を指していることを確認します(デバイス情報テーブルとシステムテーブルに正しいポインタを入力します)。 注 これらの条件はどちらも、システムブートローダによって自動的に実行されます。そのため、デバイスのブートがシステムメモリで設定されているときは、FUS を有効化する必要があります。追加のユーザ操作は必要ありません。 そうでない場合、これらのアクションは、Cortex®-M4 CPU で実行されているユーザコードで実行する必要があります。
ワイヤレススタックがインストールされ、動作している	上記と同じ手順を実行します。 2 つの連続した FUS_GET_STATE コマンドを送信して、ワイヤレススタックに FUS を起動するようリクエストします。1 つ目は FUS_STATE_NOT_RUNNING 状態を返す必要があり、2 つ目は FUS を開始します。

FUS が動作しているかどうかを確認するために、次のオプションを使用できます。

- 単一の FUS_GET_STATE コマンドを送信し、戻りステータスを確認します。FUS_STATE_NOT_RUNNING の場合、FUS は動作していません。
- SBRV オプションバイト値を確認します。
  - 0x3D800(FUS V0.5.3)または 0x3D000(FUS V1.xz)の場合、FUS は動作中である必要があります。
  - 0x3D800(FUS V0.5.3)および 0x3D000(FUS V1.xz)以外である場合、FUS は動作していません。
- ワイヤレススタックコマンドを送信します。
  - 確認応答された場合、FUS は動作していません。
  - 確認応答がない場合、FUS は動作中です。
- 共有テーブル情報を読み出します。
  - SRAM2a から共有テーブルの開始アドレスを取得するには、IPCCDBA(オプションバイト)を読み出します。
  - デバイス情報テーブルアドレスを取得します。
  - 「最後の FUS アクティブ状態」フィールドを読み出します。
    - 0x04 は、スタックが動作中である必要があることを意味します。
    - その他の値は、FUS が動作中である必要があることを意味します。
  - 起動時に FUS によって送信される「Async Ready」イベントを読み出します。このイベントと内容の詳細については、セクション 6.3.2 イベントパケットを参照してください。

1.4 メモリマッピング

FUS には、FUS サイズに応じて、Flash メモリに専用の空間があります。SRAM2a と SRAM2b 内の専用の空間、ならびに SRAM2a（共有テーブル）内の共有の空間も使用します。Flash メモリ、SRAM2a、および SRAM2b の専用の空間のサイズは、オプションバイトによって定義されます。詳細については、製品リファレンスマニュアルを参照してください。

専用の Flash メモリと SRAM 領域は、ワイヤレススタックと共有されます（インストールされている場合）。ただし、特定の時間では、FUS またはワイヤレススタックのどちらかが Cortex®-M0+ 上で動作中です。

図 1. Flash メモリマッピング

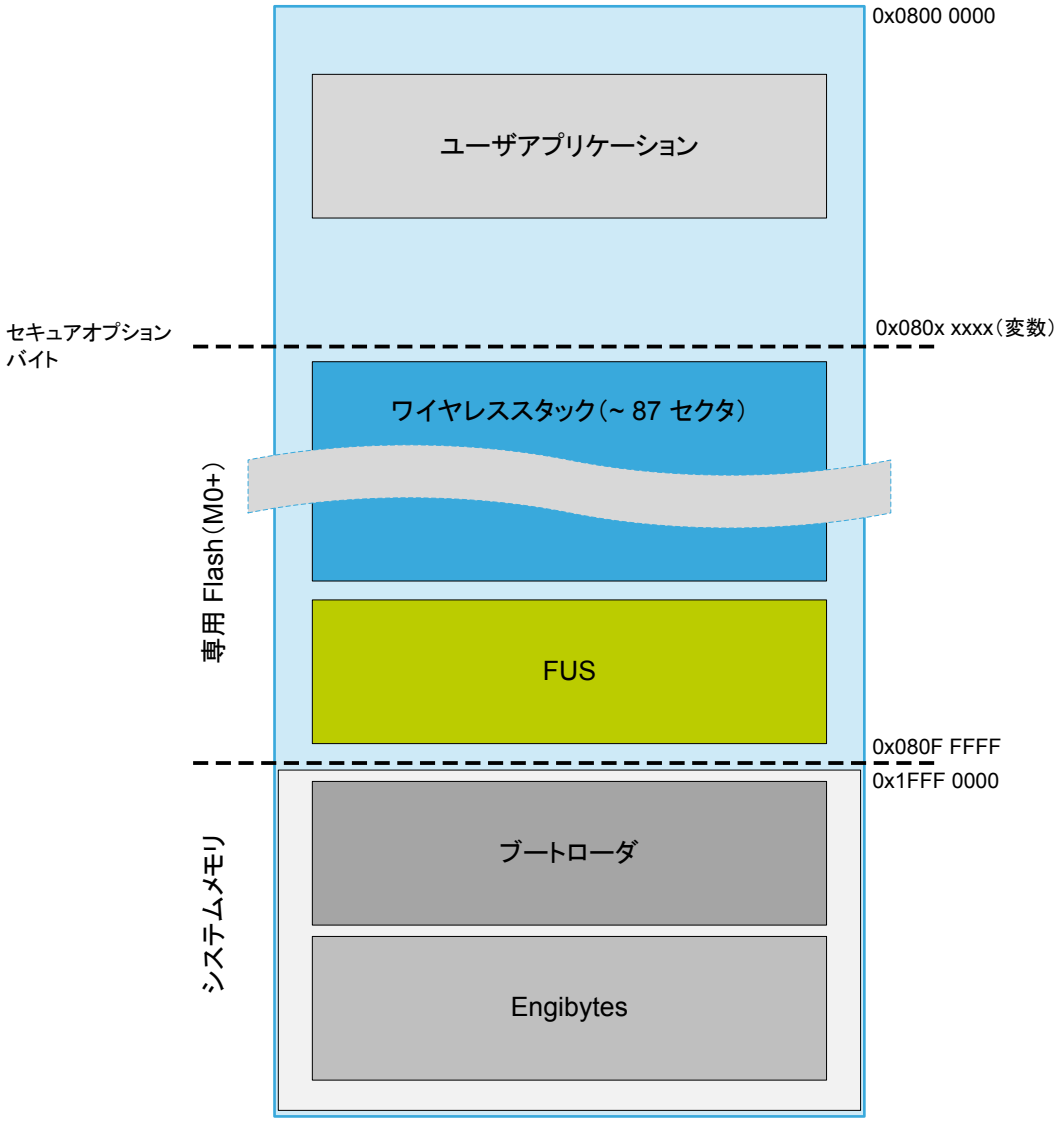
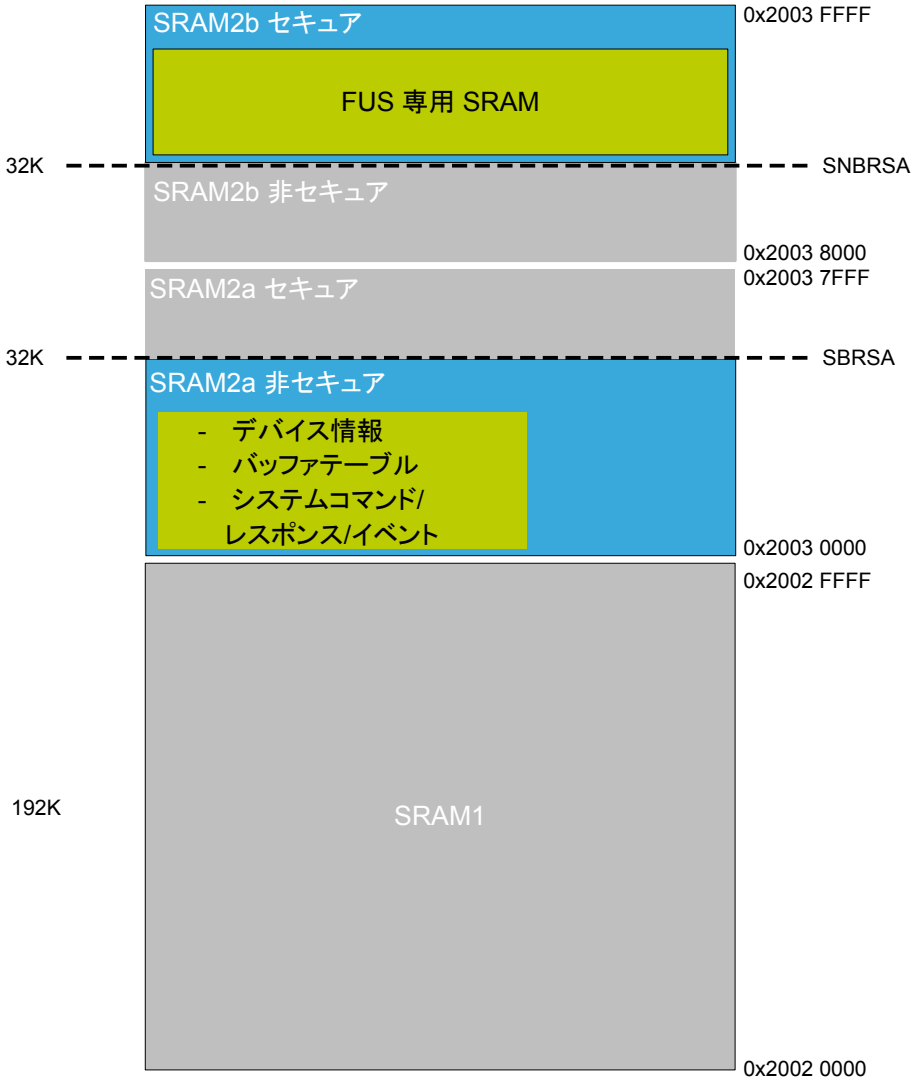


図 2. SRAM メモリマッピング



1.5 FUS リソース使用量

FUS では、表 5 に記載されたリソースのみを設定/使用します。

RCC（リセットおよびクロック制御）、Flash メモリ、PWR（電源制御）、および STM32WB シリーズマイクロコントローラの通常動作に必要なコンポーネントはすべて、Cortex®-M0+ を有効にする前に Cortex®-M4 アプリケーションで設定する必要があります（起動時にシステムブートローダによって自動的に実行されます）。

表 5. FUS リソース使用量

リソース	事例	設定
Flash	いつでも	FUS のサイズと、現在のワイヤレススタックおよびインストールが要求されているイメージのサイズに応じて、専用の Flash が FUS によって使用されます。 専用の Flash メモリの部品は、FUS 動作中に書き込まれたり、消去されたりすることがあります。 注意 FUS の動作中に Flash メモリに書込み/消去サイクルを実行する動作には注意してください。
SRAM2b	いつでも	SRAM2b のセキュア領域は、そのバージョンに応じて FUS によって使用されます。
SRAM2a	いつでも	SRAM2a のセキュア領域は、そのバージョンに応じて FUS によって使用されます。 SRAM2a のパブリック領域は、情報テーブルとコマンドテーブルの共有テーブルに書き込むために、FUS によって使用されます。
IPCC	いつでも	IPCC は、Cortex®-M0+ と Cortex®-M4 のユーザアプリケーション、ブートローダ、または JTAG の間のメールボックスのために FUS によって使用されます。 P1CH2（コマンド/レスポンスチャネル）および P1CH4（トレースチャネル）の 2 つのチャネルが使用されます。
PKA	インストールが必要な場合	PKA は有効化され、設定され、署名検証に使用されます。
AES1	キーサービスが必要な場合	AES1 はセキュアモードに設定されます（キーレジスタには Cortex®-M0+ でアクセス可能）。 AES1 キーレジスタは、ユーザが要求したキーを使用して、FUS によって書き込まれます。 AES1 は、一度セキュアモードに設定されると、次のシステムリセットまでセキュアモードを維持します。ソフトウェアでセキュアモードを無効にする方法はありません。
オプションバイト	インストール/削除が必要な場合	オプションバイト は Cortex®-M0+ レジスタを使用して FUS によってプログラムされます。SFR と SBRR レジスタのみが変更されます。
CRC	インストールが必要な場合	CRC は認証に使用され、FUS によって初期化されません。CRC が Cortex®-M4 ユーザアプリケーションによって使用される場合、FUS またはワイヤレススタックのインストール操作を開始する前に、リセットする必要があります。
システムリセット	インストール/削除が必要な場合	FUS は、オプションバイトロード時またはクリティカルエラー検出後にシステムリセットを強制します。
NVIC	いつでも	次のハンドラが使用されます。 • NMI • SysTick • IPCC_C2_RX_C2_TX_HSEM

重要

FUS またはワイヤレススタックのアップグレード/削除操作中に、Cortex®-M4 および SWD では、次を実行してはなりません。

- Flash への書込み/消去操作
- オプションバイト への書込み
- PWR および RCC の設定変更

FUS またはワイヤレススタックのアップグレード/削除中に上記のいずれかの操作が実行された場合、Flash が破損し、データが失われるリスクがあります。

重要　FUS 操作（インストール/削除）中に電源を喪失した場合、次の 3 つのケースのいずれかが発生する可能性があります。

- 影響のない電源喪失：Flash の内容が破損していない場合、FUS は障害から回復し、ユーザの介入を必要とすることなく動作を続けます。
- Flash の破損による電源喪失：Flash の内容が破損し、イメージは FUS によってインストールされません（非整数として拒否されます）。FUS はイメージを消去して、エラー（FUS_ERR_IMG_CORRUPT）を生成します。ユーザは、バイナリを再ロードすることによって、操作全体を再開し、アップグレードコマンドを FUS に送信する必要があります。
- オプションバイトの破損を伴う電源喪失：セーフブートがハードウェアによって開始され、すべての Flash はハードウェアによってロックされています。この場合、FUS V1.1.0 以上のバージョンが動作中の場合は、工場リセットがトリガされます（ユーザは、アドレス @0x5800040C に値 0x00008000 を書き込むことによって、CM0 を有効化する必要があります）。動作中の FUS のバージョンが V1.1.0 以下である場合、この時点でリカバリはできません。

注　FUS によって保存されているユーザキーがある場合、FUS のアップグレード時にこれらのキーは消去されます。

1.6 共有テーブルのメモリ使用量

通信データバッファは、ルックアップテーブルによって指定され、アドレスはオプションバイトによって決定されます。IPCCDBA（IPCC メールボックスデータのバッファベースアドレス）。このアドレスでは、ワイヤレス・アプリケーションの構築（AN5289）に詳細を示す、バッファテーブルポインタのベースアドレスを提供しています。

IPCCDBA が (SRAM2a_END_ADDRESS - SharedTable_BaseAddress) < SizeOf(SharedTable) などのようなすべてのテーブルポインタに適合しないアドレスを指す場合、FUS は共有テーブルの使用を完全に破棄する必要があり、そのため FUS では通信やコマンドを実行できません。

ユーザアプリケーションは、共有テーブルベースアドレスを正しくセットアップする必要があります。そうでない場合は、FUS サービスの初期化を停止する必要があります。

図 3. 共有テーブルのアーキテクチャ

FUS は 2 つのテーブルのみを使用します。

- デバイス情報テーブル：このテーブルは、起動時に FUS から Cortex®-M4 ユーザアプリケーション（または JTAG）に有用な情報を提供します（起動時に FUS によって書き込まれる内容）。
- システムテーブル：このテーブルでは、FUS と Cortex®-M4 ユーザアプリケーションの間でコマンドとレスポンスの交換が可能です。

2 ワイヤレススタックイメージの操作

ユーザは FUS を使用して、ワイヤレススタックのインストール、アップグレード、および削除を行うことができます。

FUS でインストールするには、ワイヤレススタックは ST マイクロエレクトロニクス（暗号化および署名を行う）によって提供される必要があります。ユーザは、ST ツールを使用して、セクション 4 ユーザ認証 で説明するように、ワイヤレススタックイメージバイナリにカスタム署名を追加することができます（ユーザ認証キーが FUS によってすでにロードされている場合）。

ワイヤレススタックのインストール、アップグレード、および削除の操作は、ブートローダ、JTAG、またはユーザアプリケーションから実行されます。STM32CubeProgrammer には、ブートローダインタフェースからこの操作を実行するツールが用意されています。USART および USB-DFU、および直接 SWD インタフェース経由で実行することもできます。

2.1 ワイヤレススタックのインストールとアップグレード

覚えておくと便利な 2 つの定義を示します。

- ワイヤレススタックのインストール：ワイヤレススタックがまだインストールされていないチップへの最初のインストールを意味します。
- ワイヤレススタックのアップグレード：ワイヤレススタックがすでにインストールされているチップへのインストールを意味します（実行中かどうかは問わない）。

操作説明

ワイヤレススタックのインストールまたはアップグレードを実行するには、次の手順に従います。

1. ワイヤレススタックイメージを www.st.com または STM32CubeMX リポジトリからダウンロードします。
2. ワイヤレススタックイメージをユーザ Flash メモリの次のアドレスに書き込みます。
   - 新規インストールの場合（ワイヤレススタックが現在インストールされていない場合）：0x08000000 + (SFSA x 4KB) - イメージサイズ
   - ワイヤレススタックがすでにインストールされている場合：
     - 新しいイメージサイズ <= 現在のイメージサイズの場合：ワイヤレススタックアドレス - イメージサイズ
     - 新しいイメージサイズ > 現在のイメージサイズの場合：ワイヤレススタックアドレス -（2 x イメージサイズ – ワイヤレススタックサイズ）
       ワイヤレススタックのインストールを開始する前に、FUS_FW_DELETE 動作を実行することをお勧めします（これは、オプション-最初のインストール= 0 が選択されている場合に、STM32CubeProgrammer によって実行されます）。
3. FUS が動作中であることを確認します（セクション 1.3 FUS を有効化する方法 の手順に従います）。
4. IPCC メカニズムを介して、FUS_FW_UPGRADE コマンドを送信します（以降のセクションで説明します）。
5. FUS_STATE_NOT_RUNNING と同じ状態になるまで、FUS_GET_STATE を送信します（これは、ワイヤレススタックがインストールされ、動作中であることを意味します）。

インストールプロセス中は、システムリセットが複数回発生する場合があります。これらのシステムリセットは FUS によって実行され、専用のメモリパラメータを変更したり、インストールされたワイヤレススタックを Cortex®-M0+ で動作させたりするために必要です。システムリセットの数は、設定と新旧のイメージの場所によって異なります。

次の表に、インストール/アップグレード操作がリクエストされたときに発生する可能性のあるエラーと、それぞれの結果を示します。

表 6. FUS アップグレードから返されたエラー

誤差	理由	結果
十分なスペースがない	現在インストールされているワイヤレススタックと、ロードされているイメージのアドレスとの間隔が小さすぎます。	インストールリクエストは拒否されます。FUS はエラー状態を返して、アイドル状態に戻ります。
イメージ署名が見つからない	署名のヘッダまたはボディが不正または破損しています。	FUS は認証エラーを返した後、アイドル状態に戻ります。イメージはインストールされず、Flash メモリ/SRAM は変更されません。
イメージ顧客の署名が見つからない	署名のヘッダまたはボディが不正または破損しています。	FUS は認証エラーを返した後、アイドル状態に戻ります。イメージはインストールされず、Flash メモリ/SRAM は変更されません。
イメージが破損した	イメージヘッダが不正、またはイメージが破損しています。	FUS はイメージ破損エラーを返した後、アイドル状態に戻ります。イメージはインストールされず、FUS によって消去されます。

誤差	理由	結果
FUS によって返される状態がない	コマンドレスポンスを受信する前に、FUS によるリセットが発生しました。	コマンドを再送信した場合、FUS レスポンスを受信する必要があります。
その他の障害	FUS 動作中の外部電源の割込みまたは外部リセット	FUS は、破損したイメージをリカバリして削除し、デフォルトの状態に戻ることができなければなりません。リカバリ操作を完了するために、システムリセットを数回実行することがあります。

メモリの考慮事項

初回インストール時、またはワイヤレススタックがインストールされていない場合、FUS はワイヤレススタックがインストールされているアドレスで最適化を行いません。ワイヤレススタックイメージは、ユーザによってロードされたのと同じアドレスにインストールする必要があります。

ワイヤレススタックのアップグレード時（ワイヤレススタックがすでにインストールされている場合）、FUS はアップグレードしたスタックを移動する場合があります（アップグレード後、実行前）。

この場合、残り空間は Cortex®-M4 ユーザアプリケーションで使用するために空けたままになります。

インストール/アップグレード操作が正常に完了すると、SRAM2a、SRAM2b、Flash のメモリのセキュア境界と SBRV の値は、インストールされたワイヤレススタックの要件に従って変更されます。

2.2 ワイヤレススタックの削除

ワイヤレススタックの削除とは、チップにすでにインストールされているワイヤレススタックを削除することです（動作中かどうかにかかわらず）。

操作説明

ワイヤレススタックの削除を実行するには、次の手順に従います。

1. FUS が動作中であることを確認します（セクション 1.3 FUS を有効化する方法 の手順に従います）。
2. IPCC メカニズムを介して、FUS_FW_DELETE コマンドを送信します（以降のセクションで説明します）。
3. FUS_STATE_NOT_RUNNING と同じ状態になるまで、FUS_GET_STATE を送信します（これは、ワイヤレススタックがインストールされ、動作中であることを意味します）。
   削除プロセス中は、システムリセットが複数回発生する場合があります。これらのシステムリセットは FUS によって実行され、専用のメモリパラメータを変更するために必要です。システムリセットの数は、設定とワイヤレススタックの場所によって異なります。

ワイヤレススタックがインストールされておらず、削除リクエストが送信されている場合、FUS は、ワイヤレススタックが見つからなかったことを通知するエラー状態（FUS_STATE_IMG_NOT_FOUND）を返します。

メモリの考慮事項

削除操作が正常に実行されると、ワイヤレススタックによって使用されていたすべての空間を Cortex®-M4 ユーザアプリケーションまたは追加のワイヤレススタックのインストール操作に自由に使用できるようになります。

イメージの開始アドレスはセクタの開始（4 KB の倍数）に合わせる必要があり、イメージのサイズは 4 バイトの倍数である必要があります。そうでない場合、FUS はインストール手順を拒否します。

2.3 ワイヤレススタック開始

ワイヤレススタックがインストールされていても、現在動作していない可能性があります。結果の状況：インストールが完了し、ワイヤレススタックが動作中のとき、ユーザアプリケーションが FUS_GET_STATE を 2 回連続で送信すると、FUS が再び開始します。

その場合、FUS_START_WS コマンドを送信するで、ワイヤレススタックの実行を起動できます。このコマンドにより、Cortex®-M0+ 実行がワイヤレススタックに切り替わり、1 つ以上のシステムをリセットします。

コマンドは、FUS_GET_STATE が FUS_STATE_NOT_RUNNING 値を返したときに完了します。この値を受信すると、他の FUS_GET_STATE を発行する必要はありません。そうでない場合は、FUS が再実行されます。

2.4 アンチロールバックの有効化

FUS がアンチロールバックをサポートしている場合、FUS にコマンドを送信することで、この機能を有効にすることができます。

このコマンドが FUS によって実行された場合は、決して無効にすることはできません。

この機能は、FUS_ACTIVATE_ANTIROLLBACK コマンドで実行されます。

このコマンドの送信後、FUS_GET_STATE コマンドを送信することでステータスを確認できます。
その後、FUS は状態 FUS_STATE_IDLE を返します。
このコマンドは、元に戻すことはできません。
いずれにしても、FUS ではロールバックできないため、このコマンドは FUS には適用されません。

重要　アンチロールバックを有効にする前に、ワイヤレススタックが正しくインストールされていること、および削除されていないことを確認します。ワイヤレススタックがインストールされていない状態で有効にすると、FUS は 0xFFFFFFFF を新しいバージョンとして登録し、ワイヤレススタックをインストールできなくなります。

アンチロールバック が有効化されると、インストール可能なワイヤレススタックのバージョンがロックされます。
現在のバージョンより前のバージョンのワイヤレススタックをインストールすることはできません。
たとえば、ワイヤレススタック V1.9.0 がインストールされていて、アンチロールバックが有効になっている場合、ワイヤレススタックのバージョン V1.9.0 以降のみをインストールできます（たとえば、V1.8.0 はインストールできなくなります）。

3 FUS のアップグレード

FUS は、ワイヤレススタックのアップグレードと同じ方法でセルフアップグレードできます。FUS を削除することはできません。

3.1 操作説明

FUS のアップグレードを実行するには、次の手順に従います。

1. www.st.com または STM32CubeMx リポジトリから FUS イメージをダウンロードします。
2. FUS イメージを、FUS のイメージディレクトリ Release_Notes.html ファイルに指定されているアドレスのユーザ Flash メモリに書き込みます。
3. FUS が動作中であることを確認します（セクション 1.3 FUS を有効化する方法 の手順に従います）。
4. IPCC メカニズムを介して、FUS_FW_UPGRADE コマンドを送信します（以降のセクションで説明します）。
5. FUS_STATE_NOT_RUNNING と同じ状態を得られるまで、FUS_GET_STATE を送信します（これは、ワイヤレススタックがインストールされ、動作中であることを意味します）。

インストールプロセス中は、システムリセットが複数回発生する場合があります。これらのシステムリセットは FUS によって実行され、専用のメモリパラメータを変更したり、インストールされたワイヤレススタックを Cortex®-M0+ で動作させたりするために必要です。システムリセットの数は、設定と新旧のイメージの場所によって異なります。

FUS は、イメージを FUS アップグレードイメージとして識別し、それに応じて FUS のアップグレードを起動します。ファームウェアスタックがすでにインストールされていて、新しい FUS のサイズが現在の FUS のサイズより大きい場合、この操作によってファームウェアスタックが再配置されることがあります。この情報および関連するすべての制約の詳細については、FUS イメージのリリースノートを参照してください。

3.2 メモリの考慮事項

FUS のアップグレードに、特定のメモリ条件は必要ありません。ただし、新しい FUS のイメージサイズが既存の FUS のサイズより大きい場合は、FUS のアップグレードに十分な空間を確保するために、ワイヤレススタックが Flash メモリの低い方に移動することがあります。

これは以下のことを意味します。

- Cortex®-M4 ユーザアプリケーションで使用可能な Flash メモリは少なくなっています。
- ワイヤレススタックは、現在のアドレスから FUS で定義された別のアドレスに移動します。
- ユーザコードがワイヤレススタック開始セクタに隣接するセクタに書き込まれている場合は、この操作中に消去されるリスクがあります。

FUS のサイズとアップグレードの結果の詳細については、Release_Notes.html ファイルに記載されています。

イメージの開始アドレスはセクタの開始（4 KB の倍数）に合わせる必要があり、イメージのサイズは 4 バイトの倍数である必要があります。そうでない場合、FUS はインストール手順を拒否します。

4 ユーザ認証

FUS サービスを使用すると、ユーザは、ST マイクロエレクトロニクス によって提供される任意のイメージ（ワイヤレススタックまたは FUS イメージ）にカスタマイズした署名を追加することができます（ST マイクロエレクトロニクス によって暗号化および署名されます）。

ユーザ認証キーを使用してバイナリに署名する手順を STM32CubeProgrammer ユーザマニュアルに示します。

ユーザ認証キーがすでにインストールされている場合のみ、FUS はユーザの署名を確認します。

署名は、RSA ECC Prime256v1（NIST P-256）および HASH-256 に基づく 64 バイトのデータバッファです。これは、STM32CubeProgrammer ツールによって生成されます。

4.1 ユーザ認証キーのインストール

FUS では、次の手順で、ユーザ認証キーを格納できます。

1. FUS が動作中であることを確認します（セクション 1.3 FUS を有効化する方法 の手順に従います）。
2. IPCC メカニズムを介して、FUS_UPDATE_AUTH_KEY コマンドを送信します（以降のセクションで説明します）。
3. FUS_STATE_IDLE と同じ状態を得られるまで、FUS_GET_STATE を送信します。
   この操作によってシステムリセットが生成されることはありません。

ユーザ認証キーがインストールされたら、上記と同じフローで変更できます（ユーザ認証キーのロック操作が行われていない場合）。ただし、削除することはできません。

インストールされると、FUS は、インストールまたはアップグレードを実行する前に、バイナリユーザの署名を体系的にチェックする必要があります。署名 が存在しない場合、または認証されていない場合、インストールまたはアップグレードは FUS_STATE_IMG_NOT_AUTHENTIC と同等のエラーで拒否されます。

4.2 ユーザ認証キーのロック

FUS では、ユーザ認証キーをロックできます。これは、製品ライフ･サイクル全体を通じて、このキーを変更できなくなることを意味します。この操作は、一度実行すると元に戻せません。

ユーザ認証キーをロックするには：

1. FUS が動作中であることを確認します（セクション 1.3 FUS を有効化する方法 の手順に従います）。
2. IPCC メカニズムを介して、FUS_LOCK_AUTH_KEY コマンドを送信します（以降のセクションで説明します）。
3. FUS_STATE_IDLE と同じ状態を得られるまで、FUS_GET_STATE を送信します。
   この操作によってシステムリセットが生成されることはありません。

この操作が完了すると、ユーザー認証キーはロックされます。

5 顧客キーストレージ

FUS では、顧客キーを専用の FUS Flash メモリ領域に格納でき、その後格納されたキーを AES1 にセキュアモードでロードします（AES1 キーレジスタは、Cortex®-M0+ と Cortex®-M4 ユーザアプリケーションでアクセス可能なデータレジスタでのみアクセス可能）。

5.1 キータイプと構造

FUS は、101 個のキー（1 個のマスタキーと 100 個のクリア/暗号化キー）の格納をサポートします。

キーサイズは 128 ビットまたは 256 ビットです。キーサイズと構造は、すべてのキータイプで同じです。格納されているキーを変更したり削除したりすることはできません。

FUS では、3 つのキータイプをサポートします。

- クリアキー：暗号化されずに FUS に送信されるキー。
- マスタキー：暗号化されず、後で FUS に送信される他のキーの復号化に使用される、FUS に送信されるキー。このキーの格納は、信頼できる環境（通信パス上にキーを抽出できない環境）で行う必要があります。これにより、ユーザは、信頼できない環境でも内容を公開することなく暗号化キーを共有できます。マスタキーを AES1 キーレジスタに書き込むことはできません。復号化にのみ使用され、変更したり削除したりすることはできません。マスタキーは 1 回だけ書き込まれ、その後更新されることはありません。マスタキーが書き込まれると、マスタキーを再度書き込むリクエストは拒否され、エラーメッセージが表示されます。100 を超えるキーを書き込むと、コマンドが拒否されます。
- 暗号化キー：暗号化フォーマットで FUS に送信されるキー。その後、使用する前に FUS によってマスタキーを使用して復号化されます。このキーには、FUS による復号化を可能にする IV（初期化ベクタ）が必要です。16 ビット IV は、キー自体と同じコマンドパケットで送信されます。

ユーザキーの暗号化は、AES-128 GCM モードに基づいて行う必要があります。FUS は、AES ハードウェアを使用せずにキーを復号化します。

キータイプは、キーが送信されるコマンドパケットで FUS に知らせる必要があります（詳細はコマンドの説明を参照）。

キーは、インデックスによって管理されます。

キーが FUS に送信されると、FUS は受信を確認し、キーに割り当てられたインデックスを返します。このインデックスは FUS によって割り当てられ、ユーザアプリケーションでは変更できません。

キーを格納するには、ユーザアプリケーションは FUS_STORE_USR_KEY を FUS に送信し（キータイプと、関連する IV がある場合はそれも含む）、その後キーインデックスを受信する必要があります。

格納されたキーを使用するには、ユーザアプリケーションで次を行う必要があります。

- AES1 初期化レジスタおよび IV レジスタを設定します。
- FUS_LOAD_USR_KEY を FUS に送信し、キーが AES1 キーレジスタに書き込まれたことを示すレスポンスの受信を待ちます。
- 格納されたキーを使用してデータを復号/暗号化するために、AES1 データレジスタに書き込みます（キーレジスタは保護されたままであり、Cortex®-M4 ユーザアプリケーションはアクセスできません）。100 を超えるキーが書き込まれた場合、そのコマンドは拒否されます。

ユーザキーの管理のために、FUS が提供する追加サービスが 2 つあります。これらの 2 つのサービスは、セキュアアプリケーションのコンテキストでの Cortex®-M4 ユーザアプリケーションによる使用を対象としています。これらがブートローダまたは STM32CubeProgrammer によって公開されることはありません。

ユーザキーのロック

このサービスでは、デバイスが次にリセットされるまで、どのアプリケーションでもキーを使用できない（AES にロードできない）ようにします。このサービスは、ロックするキーのインデックスを含む FUS_LOCK_USR_KEY コマンドを送信することで使用できます（マスタキーインデックスは常に 0 であり、ロックすることもロードすることもできません）。

FUS_LOCK_USR_KEY コマンドが送信されると、FUS はリクエストされたキーをロック状態として格納し、そのキーインデックスに対する FUS_LOAD_USR_KEY の発行がすべて操作失敗に終わるようにします（コマンドレスポンスによって 0x01 が返されます）。

ユーザキーのアンロード

このサービスは、AES に現在ロードされているキーをアンロードし（FUS_LOAD_USR_KEY が使用された場合）、ユーザアプリケーションによるロードされたキーを使用した追加の操作を防ぐために使用されます。

このサービスは、アンロードするキーのインデックスを含む FUS_UNLOAD_USR_KEY コマンドを送信することで使用できます（マスタキーインデックスは常に 0 で、ロードすることもアンロードすることもできません）。

FUS_UNLOAD_USR_KEY が送信されると、FUS は AES のキーレジスタにゼロを書き込みます。これにより、ロードされたキーは使用できなくなります。

6 FUS との通信

FUS との通信は、IPCC チャネルと Cortex®-M4 ユーザアプリケーション、ブートローダ、または JTAG によって行われます。いずれの場合でも、通信の原理はまったく同じです。

STM32 システムブートローダを使用して FUS と通信し、ブートローダインタフェース（USART または USB-DFU）を直接使用することで、すべての低階層を抽象化できます。

FUS との通信には、次の 2 つの要素を使用します。

- 共有テーブル：FUS 情報の格納と、コマンド/レスポンスパケットの取得に使用されます。
- IPCC：メッセージ通知の交換に使用されます（メッセージ内容は共有テーブルに配置されます）。

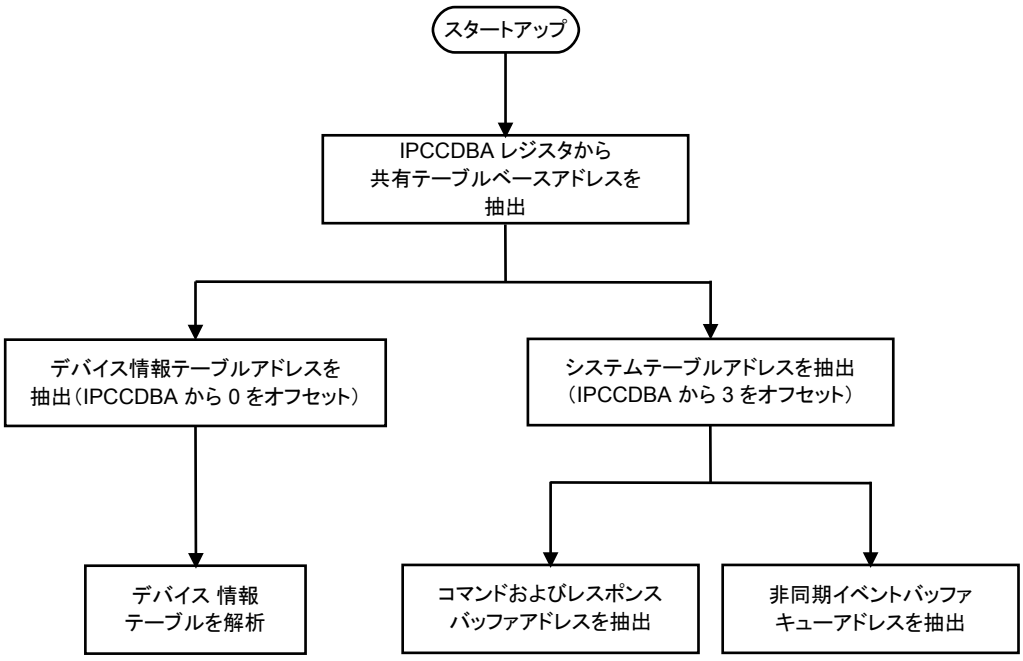
6.1 共有テーブルの使用方法

共有テーブルは、SRAM2a のパブリック領域にある情報構造で、その構造について説明しています。FUS では、2 つの共有テーブルを使用します。

- デバイス情報テーブル
- システムテーブル

両方とも、FUS と正しく通信するために、Cortex®-M4 ユーザアプリケーション（または JTAG アプリケーション）で解析する必要があります。

図 4. 共有テーブルの使用プロセス



6.1.1 デバイス情報テーブル

デバイス情報テーブルは、デバイスの現在のステータスを更新するために使用される 42 バイトのバッファです。

このテーブルは、起動時、またはプログラムされたシステムリセットの前に、FUS コードまたはワイヤレススタックコードによって更新されます。

表 7. デバイス情報テーブル

フィールド	サイズ（バイト）	値
デバイス情報テーブルの状態	4	0xA94656B9：デバイス情報テーブル有効

フィールド	サイズ(バイト)	値
		その他の値:デバイス情報テーブル無効
予約済みです。	1	予約済みです。
最後の FUS アクティブ状態	1	• 0x00:FUS はアイドル状態です。 • 0x01:ワイヤレススタックファームウェアアップグレード • 0x02:FUS ファームウェアアップグレード • 0x03:FUS サービス • 0x04:動作中のワイヤレススタック • 0x05～0xFE:未使用 • 0xFF:誤差
最後のワイヤレススタック状態	1	0x00:未開始 0x01:実行中 0x08～0xFE:未使用 0xFF:誤差
現在のワイヤレススタックタイプ	1	0x00:なし 0x01:BLE 0x02:スレッドタイプ 1 0x03:スレッドタイプ 2 詳細については、ワイヤレススタックのドキュメントを参照してください。
セーフブートバージョン	4	ファームウェアバージョン: [31:24]:メジャー(下位互換性が失われたときに更新) [23:16]:マイナー(主要機能が追加されたときに更新) [15:8]:サブバージョン(小さな変更のために更新) [7:4]:ブランチ(特定のビルド) [3:0]:ビルド(ビルドバージョン)
FUS バージョン	4	ファームウェアバージョン: [31:24]:メジャー(下位互換性が失われたときに更新) [23:16]:マイナー(主要機能が追加されたときに更新) [15:8]:サブバージョン(小さな変更のために更新) [7:4]:ブランチ(特定のビルド) [3:0]:ビルド(ビルドバージョン)
FUS メモリサイズ	4	現在の FUS スタックメモリの使用量: [31:24]:SRAM2b 使用された 1 K セクタの数 [31:24]: SRAM2a 使用された 1 K セクタの数 [15:8]:予約済みです。 [14:0]:使用された 4 K セクタの Flash メモリ数
ワイヤレススタックバージョン	4	ファームウェアバージョン: [31:24]:メジャー(下位互換性が失われたときに更新) [23:16]:マイナー(主要機能が追加されたときに更新) [15:8]:サブバージョン(小さな変更のために更新) [7:4]:ブランチ(特定のビルド) [3:0]:ビルド(ビルドバージョン) スタックが存在しない場合、データはすべて 0xFFFF FFFF
ワイヤレススタックのメモリサイズ	4	現在のワイヤレススタックのメモリ使用量: [31:24]:SRAM2b 使用された 1 K セクタの数 [31:24]:SRAM2a 使用された 1 K セクタの数 [15:8]:予約済みです。

フィールド	サイズ（バイト）	値
		[14:0]：使用された 4 K セクタの Flash メモリ数 スタックが存在しない場合、データはすべて 0xFFFF FFFF
ワイヤレス FW-BLE 情報	4	[31:0]：ワイヤレススタックで使用するために予約済み スタックが存在しない場合、データはすべて 0xFFFF FFFF
ワイヤレス FW スレッド情報	4	[31:0]：ワイヤレススタックで使用するために予約済み スタックが存在しない場合、データはすべて 0xFFFF FFFF
予約済みです。	4	0x00000000
UID64	8	STM32 デバイスのユニーク ID 64 ビット
デバイス ID	2	STM32 汎用デバイス ID

6.1.2 システムテーブル

システムテーブルは、次の表で説明する 2 つのバッファポインタを含む 8 バイトのテーブルです。

表 8. システムテーブルの内容

アドレス	サイズ（バイト）	内容	説明
0x00	4	システムコマンド/レスポンスバッファのアドレス	シングルバッファは、特定の時間に使用されます。コマンドまたはそのレスポンスのみを書き込む必要があります。レスポンスがコマンドを上書きします。新しいコマンドが以前のコマンドレスポンスを上書きします。
0x04	4	システムイベントのキューバッファのアドレス（最初のイベントのアドレス）	FUS コードは、必要に応じて解析し、キューを満たす必要があります。イベントメッセージは連鎖リストとして管理され、Cortex®-M4 がそれらを読み出すと開放されます（IPCC を介して通知されます）。 イベントの解析は、サイズのみで行われます（連鎖リスト構造ではありません）。

FUS との通信に役立つ情報を得るために、Cortex®-M4 コード（アプリケーションまたはブートローダ）は、図 4 で説明されている解析を実行します。

6.2 IPCC の使用

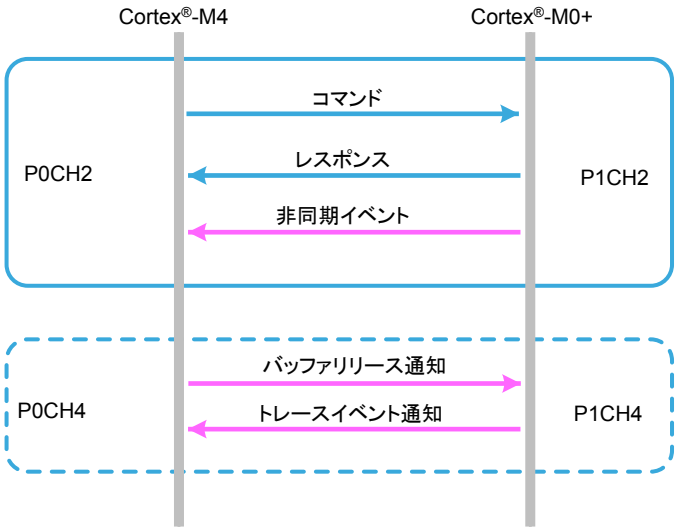
FUS はシステム IPCC によって割り当てられたチャネル、P0CH2（Cortex®-M4 側）および P1CH2（Cortex®-M0+ 側）を使用します。これらのチャネルには、3 つの通信方法があります。

- Cmd：Cortex®-M4 から Cortex®-M0+ へのコマンドリクエスト。このルートは、Cortex®-M0+ にコマンドを送信するために使用されます。
- Rsp：Cortex®-M0+ から Cortex®-M4 へのコマンドに対するレスポンス。このルートは、Cortex®-M4 によってリクエストされたコマンドに応答するためにのみ使用されます。
- Asynch Evt：Cortex®-M0+ から Cortex®-M4 への非同期イベント。このルートは、非同期イベントに関する Cortex®-M4 からの応答を必要とせずに、このイベントについて Cortex®-M4 に通知するために使用されます。

FUS によって使用されるオプションのチャネルがあります。

- P1CH4 は、トレースイベントを出力するために FUS（Cortex®-M0+）に使用される場合があります。
- P0CH4 は、Cortex®-M0+ にバッファ解放イベントについて通知するために Cortex®-M4 に使用される場合があります。

図 5. FUS によって使用される IPCC チャネル

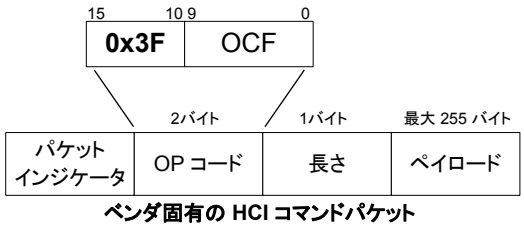


6.3 FUS コマンド

FUS は、ワイヤレススタックと同じ HCI モデルに基づくコマンド/レスポンス構造を使用します。FUS では、次の HCI コマンドのサブセットを使用します。

- ベンダ固有の HCI コマンドパケット：Cortex®-M4 から Cortex®-M0+ にコマンドを送信するために使用されます。
- HCI コマンド完了イベントパケット：Cortex®-M0+ から Cortex®-M4 にレスポンスを送信するために使用されます。
- ベンダ固有の HCI イベントパケット：Cortex®-M0+ から Cortex®-M4 に非同期イベントを送信するために使用されます。

図 6. FUS HCI サブセット



ベンダ固有の HCI コマンドパケット

HCI コマンド完了イベントパケット

ベンダ固有の HCI イベントパケット

6.3.1 パケットインジケータ

パケットインジケータは 1 バイトで、その値はパケットタイプによって異なります。

表 9. パケットインジケータの値

パケットタイプ	パケットインジケータの値
コマンドパケット	0x10
レスポンスパケット	0x11
イベントパケット	0x12

6.3.2 イベントパケット

FUS によって送信される非同期イベントは 1 つだけです。FUS 起動時にのみ送信されます。
長さフィールドは、SubEvtCode+Payload の長さを表します。

表 10. FUS 非同期イベント(ベンダ固有の HCI イベント)

長さ	SubEvtCode	ペイロード	意味
3	0x9200	エラーコード: • 0x00:動作中のワイヤレススタック • 0x01:FUS 実行中 • 0x02:SW エラー • 0x03 から 0xFF:未使用	FUS 初期化フェーズが終了し、エラーコードがペイロードバイトで表示されます。

6.3.3 コマンドパケット

次の表に、FUS でサポートされるすべてのコマンドとその HCI フォーマット値の詳細を示します。

表 11. FUS コマンド(ベンダ固有の HCI コマンドパケット)

コマンド	OP コード	長さ(バイト)	ペイロード
予約済みです。	0xFC00	N/A	N/A
FUS_GET_STATE	0xFC52	0	なし
予約済みです。	0xFC53	N/A	N/A
FUS_FW_UPGRADE	0xFC54	0 / 4[(1)] / 8 [(2)]	なし (オプションの 4 バイト)ファームウェアイメージ位置のアドレス (オプションの 8 バイト)ファームウェア転送先のアドレス
FUS_FW_DELETE	0xFC55	0	なし
FUS_UPDATE_AUTH_KEY	0xFC56	最大 65	Byte0:認証キーのサイズ N(バイト単位) バイト 1 からバイト N-1:認証キーのデータ
FUS_LOCK_AUTH_KEY	0xFC57	0	なし
FUS_STORE_USR_KEY	0xFC58	N+2	Byte0:キータイプ: • 0x00:なし • 0x01:シンプルなキー • 0x02:マスタキー • 0x03:暗号化されたキー Byte1:キーサイズ N(バイト単位) Byte2-ByteN-1:キーデータ(キー値 + ある場合は IV)
FUS_LOAD_USR_KEY	0xFC59	1	Byte0:キーインデックス(0 から 124)
FUS_START_WS	0xFC5A	0	なし

コマンド	OP コード	長さ（バイト）	ペイロード
FUS_LOCK_USR_KEY	0xFC5D	1	1 バイト、ロックするキーのインデックス
FUS_UNLOAD_USR_KEY	0xFC5E	1	1 バイト、アンロードするキーのインデックス
FUS_ACTIVATE_ANTIROLLBACK	0xFC5F	0	なし
予約済みです。	0xFC60-0xFCFF	N/A	N/A

1. 4 バイトで、現在のバージョンでは使用されていません。
2. 8 バイトで、現在のバージョンでは使用されていません。

6.3.4 レスポンスパケット

各コマンドパケットに対して、次の表に詳しく説明されている情報を含むレスポンスパケットが FUS によって送信されます。NumHCI フィールドの値は常に 0xFF に設定されます。

長さフィールドは、NumHCI+CmdOpcode+Status+Payload の長さを示します。したがって、ペイロードがない場合、長さの値は 4 です。

表 12. FUS レスポンス（HCI コマンドの完全なパケット）

ステータス	長さ	コマンド OP コード値	ステータス値	ペイロード
FUS_STATE	5	0xFC52	FUS 状態値テーブルの値	FUS 状態エラー値テーブルの値。
FW_UPGRADE_STATE	4	0xFC54	• 0x00：動作開始 • 0x01：失敗 • 0x02～0xFF：未使用	なし
FW_DELETE_STATE	4	0xFC55		なし
UPDATE_AUTH_KEY_STATE	4	0xFC56	• 0x00：動作完了 • 0x01：失敗 • 0x02～0xFF：未使用	なし
LOCK_AUTH_KEY_STATE	4	0xFC57		なし
STORE_USR_KEY_STATE	5	0xFC58		1 バイト：格納されたキーインデックス（0 ~ 100）
LOAD_USR_KEY_STATE	4	0xFC59		なし
FUS_START_WS_START	4	0xFC5A	• 0x00：動作開始 • 0x01：失敗 • 0x02～0xFF：未使用	なし
FUS_LOCK_USR_KEY	4	0xFC5D	• 0x00：動作完了 • 0x01：失敗 • 0x02～0xFF：未使用	なし
FUS_UNLOAD_USR_KEY	4	0xFC5E	• 0x00：動作完了 • 0x01：失敗 • 0x02～0xFF：未使用	なし
FUS_ACTIVATE_ANTIROLLBACK	4	0xFC5F	• 0x00：動作完了 • 0x01：失敗 • 0x02～0xFF：未使用	なし

FUS レスポンス状態値の詳細を、次の表に示します。一部の値は範囲（0x10 から 0x1F など）で表され、これはその範囲のすべての値が同じ状態の意味を提供することを意味します（0x12 と 0x1E は両方とも FUS_STATE_FW_UPGRD_ONGOING を意味する、など）。この値の範囲は、プロトコルの将来の拡張のために予約されています。

表 13. FUS 状態値

値	名前	意味
0x00	FUS_STATE_IDLE	FUS はアイドル状態です。最後の動作が正常に完了し、その状態に戻りました。進行中の動作はありません。
0x01..0x0F	未使用	これらの値は、将来の使用のために予約済みです。

値	名前	意味
0x10..0x1F	FUS_STATE_FW_UPGRD_ONGOING	ファームウェアのアップグレード操作が進行中です。
0x20..0x2F	FUS_STATE_FUS_UPGRD_ONGOING	FUS のアップグレード操作が進行中です。
0x30..0x3F	FUS_STATE_SERVICE_ONGOING	サービスが進行中です。認証キーサービス（更新/ロック）またはユーザキーサービス（格納/ロード）。
0x40..0xFE	未使用	これらの値は、将来の使用のために予約済みです。
0xFF	FUS_STATE_ERROR	エラーが発生しました。エラーの原因の詳細については、レスポンスのペイロードを参照してください。

表 14. FUS 状態エラー値

値	名前	意味
0x00	FUS_STATE_NO_ERROR	発生したエラーはありません。
0x01	FUS_STATE_IMG_NOT_FOUND	ファームウェア/FUS のアップグレードがリクエストされましたが、イメージが見つかりません（イメージヘッダが破損している、Flash メモリが破損している、など）。
0x02	FUS_SATE_IMC_CORRUPT	ファームウェア/FUS のアップグレードがリクエストされ、イメージが検出されましたが、認証されておらず、整数ではありません（データの破損）。
0x03	FUS_STATE_IMG_NOT_AUTHENTIC	ファームウェア/FUS のアップグレードがリクエストされ、イメージが検出されましたが、署名が無効です（誤った署名、誤った署名ヘッダ）。
0x04	FUS_SATE_NO_ENOUGH_SPACE	ファームウェア/FUS のアップグレードがリクエストされ、イメージが検出され認証されましたが、すでにインストールされたイメージがあり、インストールするための十分な空間がありません。スタックを低い位置にインストールして、再試行してください。
0x05	FUS_IMAGE_USRABORT	操作はユーザによって中止されたか、またはパワーオフが発生しました。
0x06	FUS_IMAGE_ERSERROR	Flash 消去エラー
0x07	FUS_IMAGE_WRTERROR	Flash 書込みエラー
0x08	FUS_AUTH_TAG_ST_NOTFOUND	イメージに ST マイクロエレクトロニクス の認証タグが見つからないエラー
0x09	FUS_AUTH_TAG_CUST_NOTFOUND	イメージに顧客認証タグが見つかりません。
0x0A	FUS_AUTH_KEY_LOCKED	ユーザがロードしようとしているキーが現在ロックされています。
0x11	FUS_FW_ROLLBACK_ERROR	古いバージョンの FW へのロールバックが検出されましたが、許可されませんでした。
0x12..0xFD	N/A	将来の使用のために予約済みです。
0xFE	FUS_STATE_NOT_RUNNING	FUS は現在動作していません。ワイヤレススタックが動作中であり、この状態に戻りました。
0xFF	FUS_STATE_ERR_UNKOWN	不明なエラー

6.4 イメージフッタ

イメージアップグレードの各要素には、独自のフッタがあります。

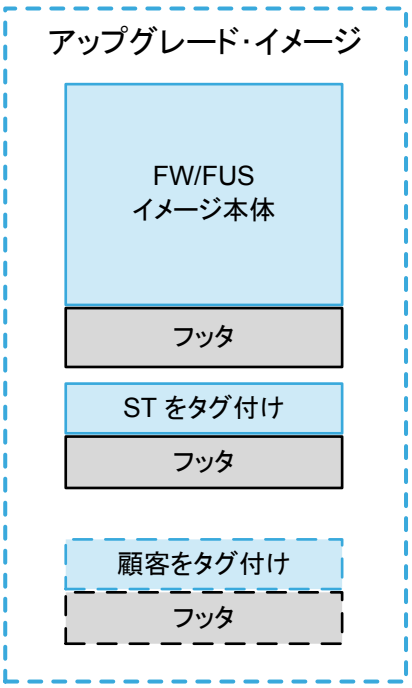
- イメージ本体
- ST マイクロエレクトロニクス の署名（必須要素）
- 顧客の署名（オプション要素）

フッタは、相対要素の最後からフッタとして直接続ける必要があります（たとえば、イメージ本体のヘッダアドレスは、イメージ本体のアドレスに隣接している必要があります）。

認証タグにはこの連続性の義務はなく、イメージの隣に配置する必要はありません。ユーザ Flash メモリの任意の場所に配置されます。FUS は、イメージの位置に関係なく、それらを探します。

すべてのイメージ、フッタアドレス、およびサイズは、4 バイトの倍数で、4 バイトに整列させる必要があります。そうしないと、FUS に認識されません。

図 7. イメージフッタの配置



各フッタには、FUS が認識できるようにするための識別値が含まれています。

図 8. FW/FUS のアップグレードイメージのフッタ構造

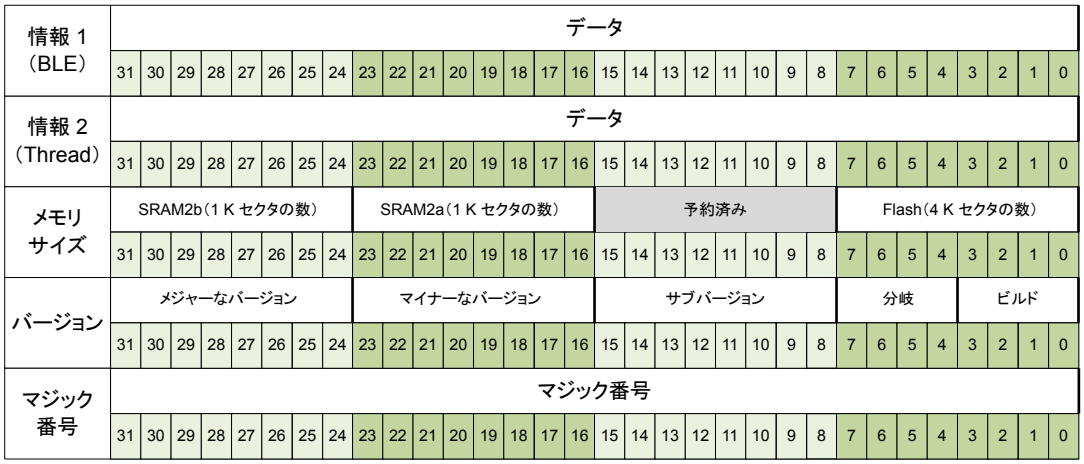


表 15. イメージフッタ構造の解析

フィールド	意味
Info1	ワイヤレススタック / FUS イメージ固有
Info2	ワイヤレススタック / FUS イメージ固有
Flash メモリ	4 KB の倍数で表されるイメージの合計サイズ
SRAM2a	SRAM2a セキュア領域でのイメージの総必要容量
SRAM2b	SRAM2b セキュア領域でのイメージの総必要容量
ビルド	バージョンのビルド番号
分岐	バージョンの枝番号
サブバージョン	バージョンのサブバージョン番号
マイナーなバージョン	マイナーなバージョン番号
メジャーなバージョン	メジャーなバージョン番号
マジック番号	イメージの性質を識別可能にする特定の値。

注　FUS V1.2.0 のバージョンは、FUS のすべてのバージョンからアップグレードできるように、バイナリで 0xFFFFFFFF と書き込まれます。

図 9. 署名（タグ）フッタの構造

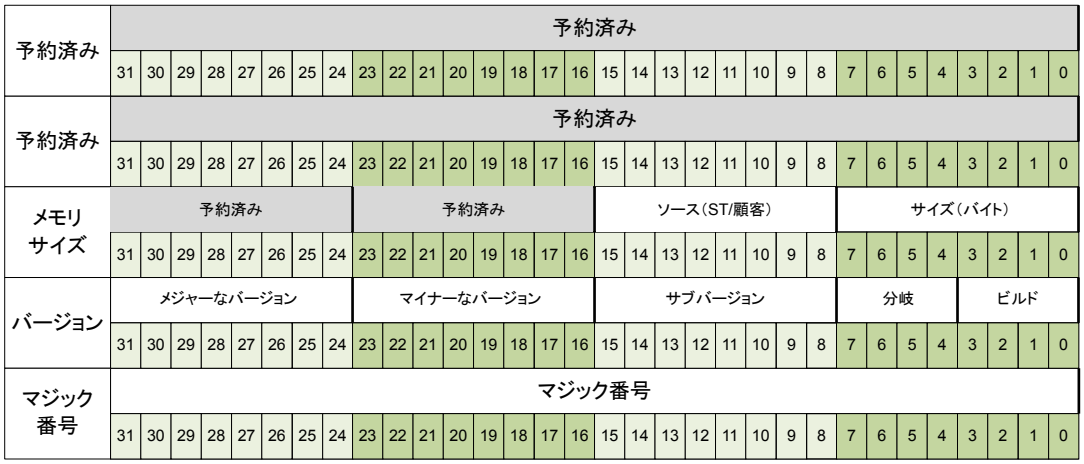


表 16. 署名フッタの解析

フィールド	意味
予約済みです。	このバージョンでは使用されない
サイズ	署名の合計サイズ（バイト）（フッタなし）
転送元	署名の性質： 0x00：ST の署名 0x01：顧客の署名
ビルド	バージョンのビルド番号
分岐	バージョンの枝番号
サブバージョン	バージョンのサブバージョン番号
マイナーなバージョン	マイナーなバージョン番号
メジャーなバージョン	メジャーなバージョン番号
マジック番号	イメージの性質を識別可能にする特定の値。

次の表に、イメージの性質を識別可能にするマジック番号の値について詳しく示します。

表 17. マジック番号の値

値	性質
0x23372991	ワイヤレススタックイメージ
0x32279221	FUS イメージ
0xD3A12C5E	ST マイクロエレクトロニクス の署名
0xE2B51D4A	顧客の署名
0x42769811	その他のファームウェアイメージ

7 FUS 用の STM32 システムブートローダ拡張

FUS 動作をサポートするために、STM32WB システムブートローダにコマンドセット拡張が追加されました。これらのコマンドは USART および USB-DFU インタフェースに実装され、既存の標準ブートローダコマンドと同じルールに従います。

このセクションの理解を助けるために、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（AN2606）および STM32 ブートローダで使用される USART プロトコル（AN3155）および STM32 ブートローダで使用される USB DFU プロトコル（AN3156）ドキュメントを、事前に読む必要があります。

7.1 USART 拡張

FUS 拡張をサポートするために、ブートローダ USART 標準プロトコルに 2 つのコマンドが追加されました。すべての FUS コマンドは、これら 2 つの特殊コマンドを経由します。1 つは書込み用（ホストから FUS へのすべての FUS コマンドで使用）、1 つは読出し用（FUS からホストへのすべての FUS コマンドで使用）です。

表 18. ブートローダ USART コマンドの拡張

コマンド	OP コード	用途
特殊読出しコマンド	0x50（補数 0xAF）	FUS からデータを取得
特殊書込みコマンド	0x51（補数 0xAE）	FUS にデータを送信

注 ブートローダの場合、FUS で追加された次のコマンドはサポートされません（UART および USB DFU でもサポートされません）。

- FUS_LOCK_USR_KEY
- FUS_UNLOAD_USR_KEY
- FUS_ACTIVATE_ANTIROLLBACK

ロックおよびアンロードのユーザキーは、Cortex®-M4 ユーザアプリケーションで使用することのみを目的とした 2 つのコマンドです。

アンチロールバックの有効化は、Cortex®-M4 ユーザアプリケーションコードに実装するか、STM32CubeProgrammer 機能を使用するか、STM32 オープンブートローダのサンプルコードを使用することによって、使用できます。

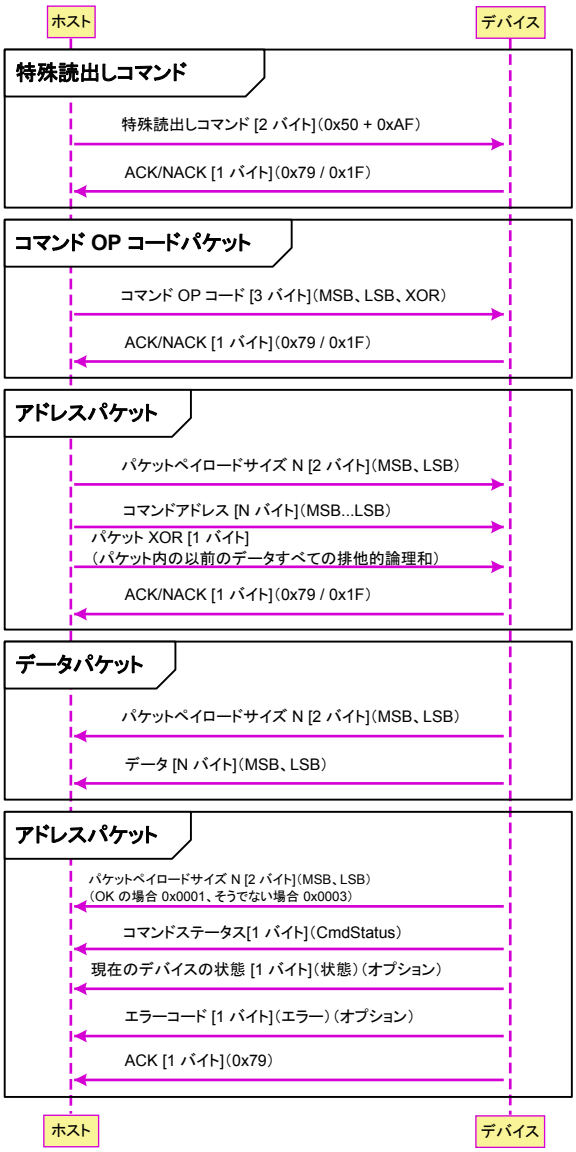
7.1.1 USART 特殊読出し

特殊読出しコマンドを使用して、デバイスからリクエストデータを送信する FUS コマンドを実行します。これは 5 つの個別のパケットに分割されます。

- 特殊読出しコマンドパケット：
  - ホストは特殊読出しコマンドコードと補数（0x50、0xAF）を送信し、ACK/NACK バイトを待ちます。NACK の場合、コマンドがサポートされていないことを意味します。
- コマンド OP コードパケット
  - ホストは、以下を含むコマンドパケットを送信します。
    - FUS コマンド OP コード（2 バイト）
    - FUS コマンド OP コード（2 バイト）の XOR
  - OP コードがサポートされている場合、デバイスは ACK を送信します。そうでない場合は NACK を送信します。
- アドレスパケット：
  - ホストは、アドレスパケットペイロードサイズを 2 バイトで送信します（MSB から順に）。
  - ホストはアドレスペイロードバイトを送信します（MSB から順に）。
  - ホストはパケットの XOR 値を送信します（現在のパケットの以前のすべてのバイトのチェックサム、1 バイト）。
  - データが正しく、サポートされている場合、デバイスは ACK を送信します。そうでない場合は NACK を送信します。
- レスポンスデータパケット：（オプション）
  - デバイスは、パケットデータペイロードサイズ（バイト単位）を 2 バイトで送信します（MSB から順に）。
  - デバイスはデータペイロードバイトを送信します（MSB から順に）。データペイロードを必要としないコマンドもあります。

- レスポンスステータスパケット:
  - デバイスは、パケットペイロードサイズ（バイト単位）を 2 バイトで送信します（MSB から順に）。
  - デバイスは、コマンドステータス（ホストがリクエストしている現在のコマンドのステータス）を 1 バイトで送信します。
  - デバイスは、現在のデバイス状態を 1 バイトで送信します（オプション、ペイロードサイズ > 3 の場合）。
  - デバイスは、現在のコマンドエラーコード（またはキーインデックス）を 1 バイトで送信します。
  - デバイスは、レスポンスパケットの信号終了に対して ACK を送信します。

図 10. USART 特殊読出しコマンド

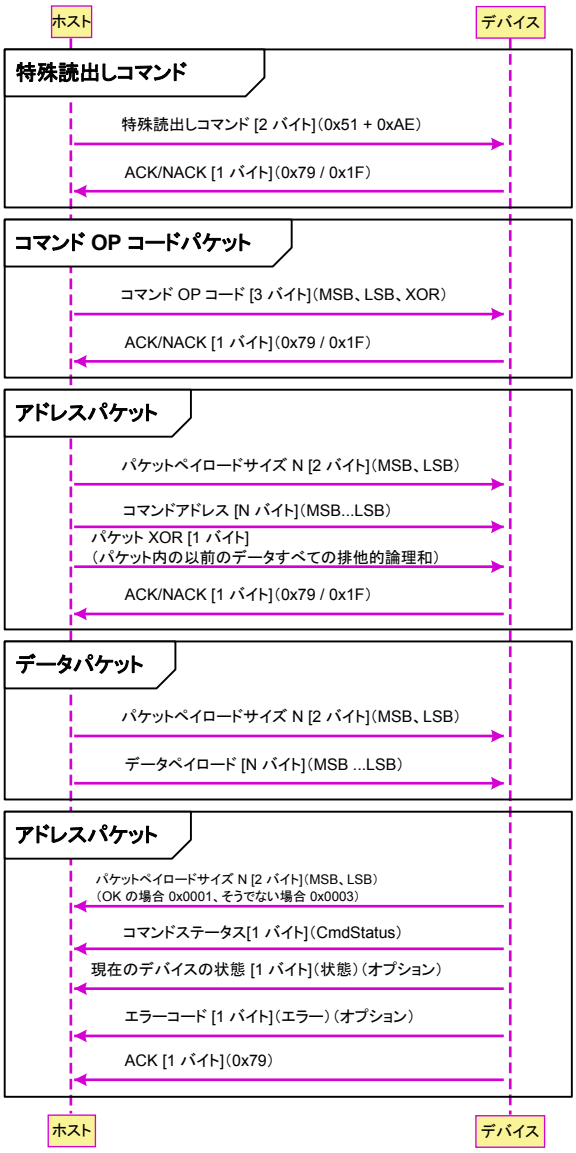


7.1.2 USART 特殊書込み

特殊書込みコマンドを使用して、デバイスからリクエストデータを送信する FUS コマンドを実行します。これは 4 つの個別のパケットに分割されます。

- 特殊書込みコマンドパケット：ホストは特殊書込みコマンドコードと補数（0x51、0xAE）を送信し、ACK/NACK バイトを待ちます。NACK の場合、コマンドがサポートされていないことを意味します。
- コマンド OP コードパケット：
  - ホストは、以下を含むコマンドパケットを送信します。
    - FUS コマンド OP コード（2 バイト）
    - FUS コマンド OP コード（2 バイト）の XOR
  - OP コードがサポートされている場合、デバイスは ACK を送信します。そうでない場合は NACK を送信します。
- アドレスパケット：
  - ホストは、アドレスパケットペイロードサイズを 2 バイトで送信します（MSB から順に）。
  - ホストはアドレスペイロードバイトを送信します（MSB から順に）。
  - ホストはパケットの XOR 値を送信します（現在のパケットの以前のすべてのバイトのチェックサム、1 バイト）。
  - データが正しく、サポートされている場合、デバイスは ACK を送信します。そうでない場合は NACK を送信します。
- データパケット：
  - ホストは、パケットデータペイロードサイズ（バイト単位）を 2 バイトで送信します（MSB から順に）。コマンドにデータが必要ない場合、この数字はゼロになることがあります。
  - ホストはデータペイロードバイトを送信します（MSB から順に）。ペイロードサイズがゼロの場合、データは送信されません。
  - ホストはパケットの XOR 値を送信します（現在のパケットの以前のすべてのバイトのチェックサム、1 バイト）。
  - データが正しく、サポートされている場合、デバイスは ACK を送信します。そうでない場合は NACK を送信します。
- レスポンスパケット：
  - デバイスは、パケットペイロードサイズ（バイト単位）を 2 バイトで送信します（MSB から順に）。
  - デバイスは、コマンドステータス（ホストがリクエストしている現在のコマンドのステータス）を 1 バイトで送信します。
  - デバイスは、現在のデバイス状態を 1 バイトで送信する場合があります（オプション、ペイロードサイズ > 1 の場合）。
  - デバイスは、現在のコマンドエラーコードを 1 バイトで送信します（オプション、ペイロードサイズ > 1 の場合）。
  - デバイスは、レスポンスパケットの信号終了に対して ACK を送信します。

図 11. USART 特殊書込みコマンド



7.1.3 USART FUS コマンドの配置

特殊読出しコマンドには 1 つの FUS コマンドのみが配置されます。

表 19. 読出しコマンドでの USART FUS コマンドの配置

コマンド	OP コード	アドレスパケット	データパケット	Cmd ステータスパケット
FUS_GET_STATE	0x54	サイズ = 0x0000 データ = なし	サイズ = 0x0003 データ = [0x00, FUS_STATE, ErrorCode]	サイズ = 0x0001 または 0x0003 データ = OK の場合 [0x00]、または KO の場合 [0x01、状態、エラー]

特殊な書込みコマンドに 7 つの FUS コマンドが配置されています。

表 20. 書込みコマンドでの USART FUS コマンドの配置

コマンド	OP コード	アドレスパケット	データパケット	Cmd ステータスパケット
FUS_FW_DELETE	0x52	サイズ = 0x0000 データ = なし	サイズ = 0x0000 データ = なし	サイズ = 0x0001 または 0x0003 データ = OK の場合 [0x00]、または KO の場合 [0x01、状態、エラー]
FUS_FW_UPGRADE	0x53	サイズ = 0x0000 データ = なし	サイズ = 0x0000 データ = なし	
FUS_UPDATE_AUTH_KEY	0x56	サイズ = 0x0000 データ = なし	サイズ = 最大 65 データ = キー(1 バイトのキーサイズ + 64 バイトのキーデータ)	
FUS_LOCK_AUTH_KEY	0x57	サイズ = 0x0000 データ = なし	サイズ = 0x0000 データ = なし	
FUS_STORE_USR_KEY	0x58	サイズ = 0x0000 データ = なし	サイズ = 最大 34 データ = [キータイプ(1 バイト)、キーサイズ(1 バイト)、キーデータ(16/32 バイト)]	サイズ = 0x0003 データ = [0x00、状態、キーインデックス]
FUS_LOAD_USR_KEY	0x59	サイズ = 0x0000 データ = なし	サイズ = 0x0001 データ = [キーインデックス]	サイズ = 0x0001 または 0x0003 データ = OK の場合 [0x00]、または KO の場合 [0x01、状態、エラー]
FUS_START_WS	0x5A	サイズ = 0x0000 データ = なし	サイズ = 0x0000 データ = なし	

7.2 USB-DFU 拡張

FUS コマンドは、ブートローダ USB-DFU 標準のダウンロードおよびアップロードコマンドで処理されます。

7.2.1 USB-DFU ダウンロード FUS 拡張

ブートローダ USB-DFU ダウンロード FUS 拡張は、SET_ADDRESS_POINTER および ERASE 標準コマンドと同じ方法で管理されます。値 = 0 と以降のバイトはコマンドデータです(MSB から順に)。

FUS_STORE_USR_KEY には 2 つのステップに分割された例外があります。

1. ダウンロードコマンド、キーデータ(最大 34 バイト)の送信のみが可能
2. アップロードコマンド、ダウンロードステップの後に実行し、キーインデックス(1 バイト)を取得できるようにする必要がある

表 21. USB-DFU ダウンロード拡張

コマンド	OP コード	データ
FUS_FW_DELETE	0x52	なし
FUS_FW_UPGRADE	0x53	なし
FUS_UPDATE_AUTH_KEY	0x56	キーバッファ = [キーサイズ（1 バイト）、キーデータ（64 バイト、MSB から順に）]
FUS_LOCK_AUTH_KEY	0x57	なし
FUS_STORE_USR_KEY	0x58	キーバッファ = [キータイプ（1 バイト）、キーサイズ（1 バイト）、キーデータ（16/32 バイト）]
FUS_LOAD_USR_KEY	0x59	キーインデックス（1 バイト）
FUS_START_WS	0x5A	なし

7.2.2 USB-DFU アップロード FUS 拡張

ブートローダ USB-DFU アップロード FUS 拡張は、物理アドレスを読み出すための通常のアップロードコマンドと同じ方法で管理されます（wBlockNum > 1）。ただし、この場合、仮想メモリのアドレスマスクが使用されます（0xFFFF0000）。そのため、FUS 読出しコマンドは、仮想アドレス 0xFFFF00YY への読出しによって管理されます。ここで、YY は FUS コマンドの OP コードです。

アップロードコマンドにより、キーインデックスを取得する FUS_STORE_USR_KEY の 2 番目のステップを実行できます。

表 22. USB-DFU アップロード拡張

コマンド	アドレス	返されるデータ
FUS_GET_STATE	0xFFFF0054	ステートバッファ = [FUS 状態（1 バイト）、FUS エラーコード（1 バイト）]
FUS_STORE_USR_KEY	0xFFFF0058	キーインデックス（1 バイト）

8 FAQ とトラブルシューティング

表 23. よくある質問と回答

質問/トラブルシューティング	回答
ST から未使用の STM32WB デバイスを受け取った場合、具体的には何が含まれていますか?	ST マイクロエレクトロニクス が提供するすべての STM32WB デバイスには、デフォルトで FUS とブートローダが含まれています。 プリインストールされたワイヤレススタックは含まれていません。
FUS のバージョンを読み取れません。	次の条件を満たしている場合は、デバイス情報テーブルにアクセスできます。 1. デバイス情報テーブルのアドレスが IPCCDBA オプションバイトが示す位置に書き込まれていること 2. Cortex®-M0+ が有効であること 3. FUS が Cortex®-M0+(ワイヤレススタックではない)で動作していること(ワイヤレススタックが動作中の場合は、FUS_GET_STATE コマンドを 2 回送信することで、FUS を強制的に実行可能)。したがって、SWD を介してデバイスにアクセスする場合、デバイス情報テーブルは通常、まだ書き込まれていないか、Cortex®-M0+ がまだ有効化されていないため、無効です。そのため、ブートローダの実行中にデバイス情報テーブルを読み出す方法がより便利です。これは、ブートローダが上記の(1)と(2)のアクションを実行するためです。 注 SWD を介して接続してハードウェアリセットオプション(ホットプラグ)を無効にし、ブートローダでブートを保持することで、ユーザがデバイス情報テーブルを読み出せるようにすることもできます。
FUS イメージをアップグレードしたいです。ワイヤレススタックはすでにインストール済みです。FUS をアップグレードする前に、ワイヤレススタックを削除する必要がありますか?	通常、特に FUS V0.5.3 からアップグレードする場合は、FUS のアップグレードを実行する前にワイヤレススタックを削除することをお勧めします。 既存の FUS のバージョンが V0.5.3 以降である場合は、ワイヤレススタックの削除は必須ではありません。
デバイスで FUS またはワイヤレススタックが動作中かどうかをすばやく確認するには、どうすればよいですか?	複数の確認方法があります。 • オプションバイトを読み、SBRV の値を確認します。FUS が動作している場合は 0x3D000(FUS V0.5.3 が動作している場合は 0x3D800)です。 • デバイス情報テーブル @0x20030030 を読み出します。FUS バージョンが異なる場合は、ワイヤレススタックが動作中か、Cortex®-CM0+ が無効です。 • FUS_GET_STATE コマンドを送信します。FUS_STATE_NOT_RUNNING を受信した場合は、ワイヤレススタックが動作中か、Cortex®-CM0+ が無効です。
IPCCDBA オプションバイトの用途は?	IPCCDBA は、デバイス情報テーブルの読出し/書込みを行う場所のオフセットを変更するために使用します。
アップグレード操作後、Flash メモリにアクセスできなくなり、FUS と通信できません。	最初に、SFSA=0x00 かどうかを確認します。この場合、セーフブートがトリガされたことを意味します。 セーフブートは、オプションバイト 破損が発生した場合にトリガされます。 これは、FUS のアップグレード操作中、または オプションバイト を扱うユーザアプリケーション操作中に発生することがあります。 セーフブートがトリガされると、SFSA=0x00(すべての Flash メモリセキュア)が設定され、デバイスがロックされます。これにより、ユーザアプリケーション/デバッガはユーザ Flash メモリにアクセスできなくなります。 この操作は、元に戻すことはできません。 FUS V1.1.0 以降、セーフブートはデバイスをロックする代わりに工場リセットを行うよう変更されました。
FUS のバージョンをダウングレードできますか(たとえば、現在動作中の FUS のバージョンが V1.0.2 のとき、FUS V1.0.1 はインストールできますか?)	FUS のダウングレードは、どのような組み合わせでもできません。インストールはアップグレードのみ可能です。 一時的なダウングレードする場合、FUS はアップグレードを拒否し、エラーメッセージを返します。

質問/トラブルシューティング	回答
FUS を使用してアップグレードを行う場合の典型的な STM32CubeProgrammer コマンドは?	1. 最初に、FUS_STATE_IDLE 状態レスポンスを受信するまで FUS_GET_STATE コマンドを送信し、FUS が動作していることを確認します。 – STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate – STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate – STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate ほとんどの場合、FUS_GET_STATE コマンドを 3 回送信すると、FUS が動作中であり、アイドル状態であることを確認できます。 2. 既存のワイヤレススタックを削除して、新しいワイヤレススタックをインストールします(ワイヤレススタックをアップグレードする場合)。 – STM32_Programmer_CLI.exe -c port=usb1 -fwupgrade stm32wb5x_BLE_Stack_fw.bin 0x080CB000 firstinstall=0 – STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate – STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate – .....(受信した状態が FUS_STATE_NOT_RUNNING になるまで -fusgetstate を送信し続けます。) 「firstinstall=0」を設定することで、新しいスタックをインストールする前に、前のスタックを削除できます。 以前にインストールされたスタックがなくても、"firstinstall=0" の設定によって問題が生じることはありません。 または、FUS イメージのインストールに進みます(FUS をアップグレードする場合)。 • STM32_Programmer_CLI.exe -c port=usb1 -fwupgrade stm32wb5x_FUS_fw.bin 0x080EC000 firstinstall=0 • STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate • STM32_Programmer_CLI.exe -c port=usb1 -fusgetstate • .....(受信した状態が FUS_STATE_IDLE になるまで -fusgetstate を送信し続けます。) 「firstinstall=0」は、FUS をアップグレードする前に既存のワイヤレススタックが削除されたことを意味します。 FUS V0.5.3 以外の FUS バージョンからアップグレードする場合は、「firstinstall=1」を使用できます。
セーフブートとは? どのように使用できますか?	セーフブートは FUS から独立した部分であり、オプションバイトの破損という特定の 1 つのケースを管理しています。 オプションバイトが破損していると、STM32WB ハードウェアは、実行中のファームウェアが何であれ、セーフブートするために強制的にブートします。 その後、セーフブートは次のいずれかを行います。 • デバイスを完全セキュアモードでロックします(V1.1.0 より前の FUS バージョンで)。これは、すべてのデバイスの Flash メモリにアクセスできないことを意味します。この操作を元に戻すことはできません(この操作をキャンセルする方法はなく、デバイスは使用できなくなります)。 • または、工場リセットを行いします(FUS バージョン V1.1.0 以上の場合)。これは、ワイヤレススタックがあれば削除され、Cortex®-M4 コードが消去され、ブートが FUS(未使用部品状態)にリセットされることを意味します。この操作を元に戻すことはできません。セーフブートを有効化するには、ユーザは SWD インタフェースを使用してアドレス 0x5800040C に値 0x00008000 を書き込んで、Cortex®-M0+ を有効化する必要があります。
FUS はインストール後に暗号化済ファームウェアのシャドウを消去しますか?	はい。FUS は、暗号化済ファームウェアがインストールされ、上位のアドレスに移動された後で、そのシャドウが残っているセクタを消去します。
インストールできるファームウェアイメージのサイズに制限はありますか? 新しいイメージをインストールする前に、インストールされているファームウェアイメージを削除する必要がありますか?	V1.2.0 より前の FUS のバージョンを使用する場合: • 別のワイヤレスファームウェアイメージ A がすでにインストールされている/動作しているときに、ワイヤレスファームウェアイメージ B がすでにインストールされている場合。B のサイズが A のサイズより大きい場合で、B が A に近すぎるアドレスにロードされている場合(セクション 2 ワイヤレススタックイメージの操作 で説明されているように、A の開始アドレスと B の終了アドレスの間には十分な空き空間がありません)、デバイスはファームウェア B ではなくファームウェアイメージ A(破損する)を指す SBRV 値でブロックされることがあり、その場合はリカバリを実行できない可能性があります。

質問/トラブルシューティング	回答
	• このため、ファームウェアイメージ B をインストールする前にファームウェアイメージ A を削除するか（FOTA の場合は実行できない場合があります）、インストールを実行する前に十分な空間があることを確認します。この既知の制限は、FUS V1.2.0 で修正されています。

改版履歴

表 24. 文書改版履歴

日付	版	変更内容
2019 年 3 月 21 日	1	初版発行
2019 年 6 月 17 日	2	セクション 1.2 FUS のバージョン管理と識別を追加。 更新: • セクション 2 ワイヤレススタックイメージの操作、セクション 5.1 キータイプと構造、セクション 5.1 キータイプと構造 • 表 20.書込みコマンドでの USART FUS コマンドの配置 表 22.USB-DFU アップロード拡張
2019 年 7 月 10 日	3	表 7 を更新。デバイス情報テーブル
2020 年 3 月 31 日	4	更新: • セクション 1.1 ファームウェアアップグレードサービス定義、セクション 2 ワイヤレススタックイメージの操作、セクション 2.1 ワイヤレススタックのインストールとアップグレード、セクション 2.2 ワイヤレススタックの削除、セクション 5.1 キータイプと構造、セクション 7.1 USART 拡張 • 表 1.FUS バージョン、表 11. FUS コマンド(ベンダ固有の HCI コマンドパケット)、表 12. FUS レスポンス(HCI コマンドの完全なパケット)、表 14. FUS 状態エラー値 追加:セクション 2.4 アンチロールバックの有効化、およびセクション 8 FAQ とトラブルシューティング
2021 年 5 月 6 日	5	更新: • セクション 1.2 FUS のバージョン管理と識別 • 表 1.FUS バージョン • セクション 1.3 FUS を有効化する方法 • 図 1.Flash メモリマッピング • セクション 1.5 FUS リソース使用量 • 表 5.FUS リソース使用量 • セクション 2.4 アンチロールバックの有効化 • セクション 8 FAQ とトラブルシューティング 追加: • 表 1.FUS バージョン • 表 2.FUS バージョンの互換性
2021 年 10 月 22 日	6	更新: • セクション 1.5 FUS リソース使用量 • 図 8. FW/FUS のアップグレードイメージのフッタ構造 • 図 9. 署名(タグ)フッタの構造 • セクション 8 FAQ とトラブルシューティング(新しい制限付き)

目次

表一覧

図一覧