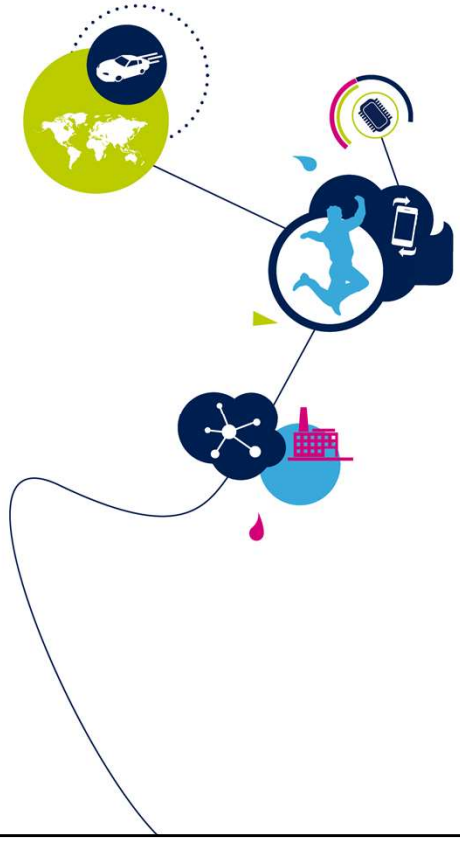
STM32H7 - MDMA

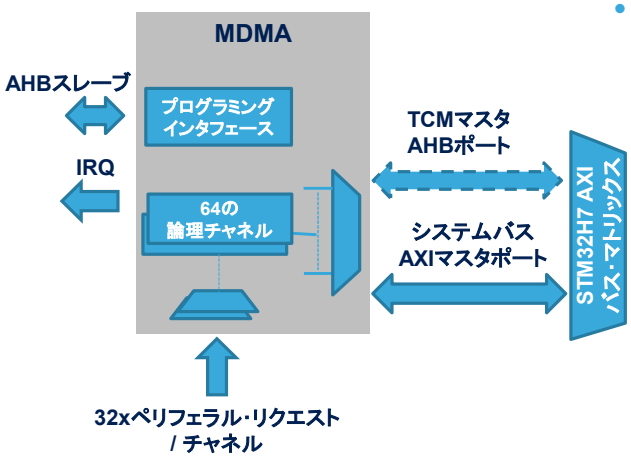
マスタ・ダイレクト・メモリ・アクセス・コントローラ（MDMA）
1.0版



life.augmented

STM32H7マスタ・ダイレクト・メモリ・アクセスコントローラ（MDMA）のプレゼンテーションへようこそ。ここでは、データ転送の処理に広く使われている本モジュールの主な特徴を説明します。

概要



- STM32H7 MDMAの特徴
  - デュアル・マスタ・バス
  - 64ビットAXI、32ビットAHBマスタ・バス
  - 柔軟性の高い個別の設定
  - ハードウェアおよびソフトウェアの優先順位管理
  - 設定可能なデータ転送モード
    - ペリフェラルからメモリ、メモリからペリフェラル、メモリからメモリのモード

アプリケーション側の利点

- ブロック転送とリンクされたリストのDMAのサポート
- CPUからデータ転送管理負荷を軽減
- 簡単な組込み

マスタ・ダイレクト・メモリアクセス（MDMA）は、CPUの介入なしに転送のチェーンリストを実行できるリンクされたリスト転送をサポートしているため、メモリ間のデータ転送用に最適化されています。これにより、CPUリソースを他の操作のために空けておくことができます。

MDMAコントローラは、メインメモリとペリフェラルレジスタのアクセス用にマスターAXIインターフェースを提供し（システムアクセスポート）、Cortex-M7 TCMメモリのアクセス用にマスターAHBインターフェースのみを提供します（TCMアクセスポート）。

主な機能

- 各MDMAチャネルでは、以下のことが可能
  - **単一のブロック転送**:
    1つのブロックが転送される(最大64KB)。ブロックの最後で、DMAチャネルは無効化
  - **繰り返しブロック転送**:
    チャネルを無効化する前に、いくつかのブロック(最大4096ブロック)が転送される
  - **リンクされたリストの転送**:
    現在のデータブロックの転送が完了すると、メモリから新しいブロック制御構造がロードされ、新しいブロック転送が開始される
- チャネル個別の柔軟性:
  - 転送元および転送先に対する個別のインクリメント、デクリメント、またはノンインクリメント・アドレッシング
  - 転送元および転送先に対する個別の転送サイズおよびインクリメント・サイズ
  - エンディアン形式の入れ替え:バイト、ハーフワード、ワード

life.augmented

各DMAコントローラチャネルは、転送元と転送先の間に単方向転送リンクを備えています。
各チャネルでは、以下を実行できます。
単一のブロック転送:1つのブロックが転送されます。ブロックの終了時には、DMAチャネルがディセーブルになり、チャネル転送終了の割込みが発生します。
繰り返しブロック転送:チャンネルを無効にする前にいくつかのブロックが転送されます。
リンクされたリストの転送:現在のデータブロック(または繰り返しの最後のブロック)の転送が完了すると、メモリから新しいブロック制御構造がロードされ、新しいブロック転送が開始されます。

MDMAは、送信元および転送先に対するアドレス指定をインクリメント、デクリメント、またはノンインクリメント(固定)にする機能も備えています。サイズとアドレスインクリメントは、転送元と転送先の両方について、個別に選択できます。

MDMAの使用例

- MDMAは、他のDMA（DMA1/DMA2、BDMA）からデータを収集し、D1ドメイン（DTCMまたはAXI-SRAM）でCPUにデータを提供するのに役立つ
- DMAのリンクされたリストは、CPUの介入を必要とせずに一連のDMA転送を実行するために使用される
  - 他のDMA用のデータを準備し、そのDMA設定をセットして転送を開始するために使用可能
  - 分散／集合をサポートするために使用される
    これは、転送元領域と転送先領域がメモリ内の連続した領域を占有する必要がないことを意味する
    転送元および転送先データ領域は、データ・ブロックの転送を制御する一連のリンクされたリストのディスクリプタによって定義される
  - JPEG画像の変換やDMA2Dによる描画などのグラフィック操作が可能

life.augmented

MDMAは、他のDMA（DMA1/DMA2、BDMA）からデータを収集し、D1ドメイン（DTCMまたはAXI-SRAM）でCPUにデータを提供するのに役立ちます。
DMAのリンクされたリストは、CPUの介入を必要とせずに一連のDMA転送を実行するために使用されます。
他のDMA用のデータを準備してから、DMAの設定をして転送を開始するために使用できます。
分散／集合をサポートするために使用されます。これは、転送元領域と転送先領域がメモリ内の連続した領域を占有する必要がないことを意味します。転送元および転送先データ領域は、データブロックの転送を制御する一連のリンクされたリストのディスクリプタによって定義されます。
JPEGデコーダーへのデータの受け渡しや、DMA2Dによる画像描画の高速化などに利用できます。

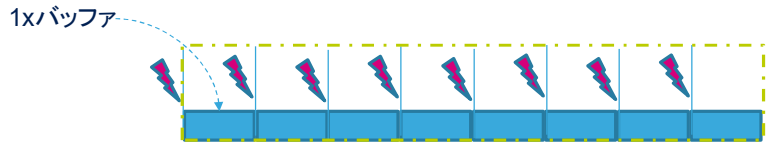
転送タイプ

- MDMAは、シングルまたはインクリメンタル・バースト転送をサポート
- 最大128バイトのソフトウェア設定可能なバーストサイズ
- 最大バースト・データ・サイズは128バイト
  データサイズがこれより大きい場合、バースト長は、転送データを一時的に格納するために使用される、128レベルのFIFOによって制限される
  - たとえば、16x64ビットまたは32x32ビット
- TCMのメモリ・アクセスでは、インクリメントとデータ・サイズが同一で32ビット以下の場合のみバースト・アクセスが可能

MDMAでは、インクリメンタル・バースト転送をサポートしています。各バーストのサイズはソフトウェアで設定可能で、最大値は128バイトです。これよりも大きなデータサイズの場合、最大128バイトのデータバーストサイズを考慮して、バースト長は制限されます（16x64ビットまたは32x32ビットなど）。
TCMのメモリアクセスでは、インクリメントとデータサイズが同一で32ビット以下の場合のみバーストアクセスが可能です。

MDMAチャネル・トリガ・モード

- シングル・リクエストで転送されるデータ配列のサイズは、以下のいずれか
  1. バッファ転送サイズ
     - TRGM="00"

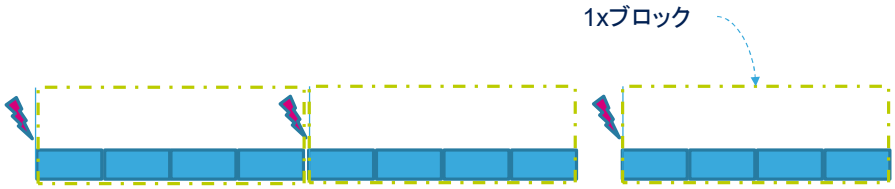


リクエストごとに1バッファが転送

サイズは、TRGM[1:0](トリガモード)選択フィールドを使用して選択されます。シングルリクエストで転送されるデータ配列のサイズは、TRGM(トリガモード)="00"の場合のバッファ転送サイズになります。

MDMAチャネル･トリガ･モード

- シングル･リクエストで転送されるデータ配列のサイズは、以下のいずれか
  1. バッファ転送サイズ
     - TRGM=“00”
  2. ブロックサイズ
     - TRGM=“01”

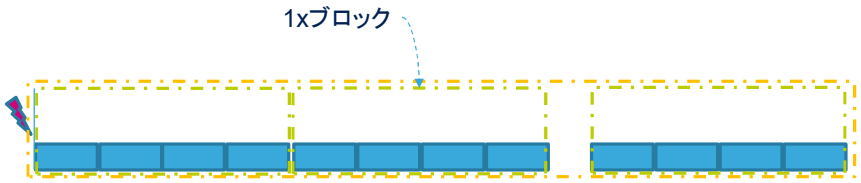


リクエストごとに1ブロックが転送

TRGM（トリガモード）=“01”の場合、シングルリクエストで転送されるデータ配列のサイズはブロックサイズとなります。

MDMAチャネル･トリガ･モード

- シングルリクエストで転送されるデータ配列のサイズは、以下のいずれか
  1. バッファ転送サイズ
     - TRGM="00"
  2. ブロックサイズ
     - TRGM="01"
  3. 繰り返しブロック
     - TRGM="10"

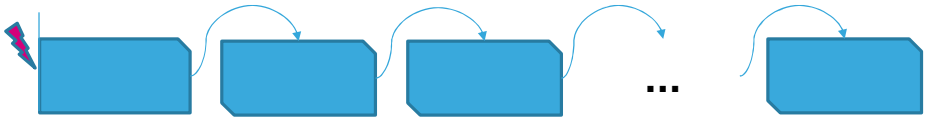


リクエストごとに複数ブロックが転送

TRGM="10"の場合、シングルリクエストで転送されるデータ配列のサイズは繰り返されるブロックとなります。

MDMAチャネル･トリガ･モード

- シングル･リクエストで転送されるデータ配列のサイズは、以下のいずれか
  1. バッファ転送サイズ
     - TRGM="00"
  2. ブロックサイズ
     - TRGM="01"
  3. 繰り返しブロック
     - TRGM="10"
  4. チャネルデータ全体
     - TRGM="11"



1回のリクエストで転送が開始され、チャネルの
リンクされたリスト･ポインタがnullになるまで継続

TRGM="11"の場合、シングルリクエストで転送されるデータ配列のサイズは、チャネルデータ全体（チャネルのリンクリスト･ポインタがnullになるまで）になります。

リクエスト･アービトレーション（1/2）

- 各チャネルには、プログラム可能なSW優先順位（4つの優先順位レベル）がある
  - 2つのチャネルのSW優先順位が同じ場合、小さいチャネル番号が優先
- MDMAチャネル･リクエスト間の新たなアービトレーションを実行せずにMDMAによって転送される最小データ･サイズは1回のバッファ転送
- ブロック転送の場合、個々のバッファの転送後、MDMAは、新しい外部リクエストと内部的に記憶されたリクエスト間での新たなアービトレーションのフェーズに移行
  - 他に優先順位の高いチャネル･リクエストがアクティブでない場合は、新しいバッファ転送が同じチャネルで開始される
  - 各バッファ転送後にチャネル･アービトレーションが実行されるため、上位のMDMAリクエストがバッファ転送期間よりも長い期間ブロックされることはない

life.augmented

各チャネルには、プログラム可能な優先順位があります。2つのチャネルのプログラム可能優先順位が同じ場合、小さいチャネル番号が優先されます。
アービタは、優先順位に基づいてMDMAチャネルリクエストを管理します。MDMAがアイドル状態の場合、各バッファの転送終了後に、すべてのMDMAリクエスト（ハードウェアまたはソフトウェア）がすべての有効チャネルで確認されます。
MDMAチャネルリクエスト間の新たなアービトレーションを開始せずにMDMAによって転送される最小データサイズは1回のバッファ転送です。
ブロック転送の場合、個々のバッファの転送後、MDMAは、新しい外部リクエストと内部的に記憶されたリクエスト間での新たなアービトレーションのフェーズに移行します。
他に優先順位の高いチャネルリクエストがアクティブでない場合は、新しいバッファ転送が同じチャネルで開始されます。
各バッファ転送後にチャネルアービトレーションが実行されるため、上位のMDMAリクエストがバッファ転送期間よりも長い期間ブロックされることはありません。

リクエスト･アービトレーション（2/2）

- 2つのデータ配列サイズ･パラメータがMDMAとアプリケーションの応答性に影響がある
  1. バッファ転送サイズ：他のチャネルのリクエストによりMDMAレベルでは中断できないデータ転送長
     - 他のチャネルのMDMAリクエストをチェックする前に、あるチャネルで、転送されるデータの長さ
  2. バースト･サイズ：バス･アービトレーション･レベルで中断できない最大データ転送長を定義
     - バースト･モードで転送できるデータ転送長
     - 他のマスタがバスへアクセスすることを阻止することがある
- 他のMDMAチャネルおよびマスタの「リアルタイム」要件を考慮し、バーストおよびバッファ転送サイズに関して適正な選択することが重要

life.augmented

以下の2つのデータ配列サイズパラメータがMDMAとアプリケーションの応答性に影響を与えます。

1. バッファ転送サイズ：他のチャネルのリクエストによってMDMAレベルでは中断できないデータ転送長
   - 他のチャネルのMDMAリクエストをチェックする前に、あるチャネルで、転送されるデータの長さです。
2. AXIバーストサイズ：これはバスアービトレーションレベルで中断できない最大データ転送長を定義します。
   - バーストモードで転送できるデータ転送長です。
   - 他のマスタがバスへアクセスすることを阻止することがあります。

他のMDMAチャネルおよびマスタの「リアルタイム」要件を考慮して、バーストおよびバッファ転送サイズに関して適正な選択をすることが重要です。

MDMAバッファ転送

- バッファ転送は、指定されたチャネルに対してMDMAリクエスト･イベントで転送される最小の論理データ量
  - MDMAリクエストに続いて現在のチャネルで転送されるデータの総量は、トリガ･モード（バッファ転送、ブロック転送、繰り返しブロック転送、全データ転送）によって決定
- 転送されるデータ項目の数、その幅（8ビット、16ビット、32ビット、または64ビット）、およびデータ転送に使われるバースト長はソフトウェアでプログラム可能
- DMA/ペリフェラル･リクエスト･イベントを処理した後、マスク･データ値をマスク･アドレスで指定されたアドレスに書き込むことによって、リクエストの確認が行われる
  - 主にリクエスト･イベント･ソースフラグのクリアに使用

life.augmented

バッファ転送は、1つのチャネルのMDMAリクエストイベントで転送される、論理的に最小のデータ量です（最大128バイト）。
MDMAバッファ転送は、一定数のデータ転送（シングルまたはバーストデータ転送）のシーケンスで構成されています。転送されるデータ項目の数とその幅（8ビット、16ビット、32ビット、または64ビット）は、ソフトウェアでプログラム可能です。データ転送に使用するバースト長も個別にプログラム可能です。
データ配列の転送を必要とするイベントの後、DMA/ペリフェラルのリクエスト信号がMDMAコントローラに送信されます。MDMAコントローラは、チャネルの優先順位に応じて、リクエストを処理します。これらのレジスタが設定されている場合に、マスクアドレスに指定されたアドレスにマスクデータ値を書き込むことによって、リクエストが確認応答されます。

MDMAブロック転送（1/2）

- ブロックは、最大64KBの連続したデータの配列で、連続したバッファ転送によって転送
- データの各ブロックは、「開始アドレス」と「ブロック長」で定義される
- 設定されたトリガモデルに応じて、1つのペリフェラル／DMAリクエストによってブロック全体の転送をトリガ
  - 優先順位の高い他のチャネルリクエストがアクティブでない場合は、新しいバッファ転送が同じチャネルで開始される

life.augmented

ブロックは、最大64キロバイトの「連続した」データの配列で、連続したバッファ転送によって転送されます。
データの各ブロックは、開始アドレスとブロック長で定義されます。

MDMAブロック転送（2/2）

- ブロック転送が完了すると、以下の3つのアクションが実行される可能性がある
  - ブロックが、繰り返しブロック転送の一部である場合：ブロック長を再ロードし、ブロック繰り返しアドレス更新レジスタの情報に基づいて新しいブロック開始アドレスを計算
  - 単一のブロックである場合や、繰り返し行われるブロック転送の最後のブロックである場合は、次のブロックの情報がメモリから読み込まれる
  - 現在のMDMAチャネルで転送する必要のある最後のブロックである場合：そのチャネルは無効となり、このチャネルでは今後MDMAリクエストは受け付けられない

life.augmented

ブロック転送が完了すると、以下の3つのアクションのいずれかが実行されます。
· ブロックが、繰り返しブロック転送の一部である場合：ブロック長を再ロードし、新しいブロック開始アドレスを計算します（CxBRURレジスタの情報に基づく）。
· 単一のブロックである場合、または繰り返しブロック転送の最後のブロックである場合：メモリから次のブロック情報をロードします（MDMA_CxLARのリンクされたリストアドレス情報を使用）。
· 現在のMDMAチャネルに転送する必要がある最後のブロックである場合（MDMA_CxLAR=0）：チャネルを無効にします。このチャネルに対するその後のMDMAリクエストは受け付けません。

MDMAブロック繰り返しモード

- ブロック繰り返しモードにより、転送先と転送元の「開始アドレス」が異なる場合でもブロック転送を繰り返すことが可能
- 繰り返しブロック･モードがアクティブな場合（繰り返しカウンタが“0”に等しくない場合）、現在のブロック転送の終了時に、ブロック･パラメータが以下のように更新
  - 転送するデータ･バイトのブロック数（BNDT）の値を再ロード
  - ブロック繰り返しの転送元／転送先アドレス更新モード（BRSUM/BRDUM）の設定に従ってブロックの転送元／転送先アドレス値を更新
  - 繰り返しカウンタを1だけデクリメント
- 繰り返しブロック･カウンタが0に達すると、最後のブロックは、単一のブロック転送として処理される

ブロック繰り返しモードにより、転送先と転送元の開始アドレスが異なる場合でもブロック転送を繰り返すことができます。
繰り返しブロックモードがアクティブな場合（繰り返しカウンタが“0”でない場合）、現在のブロック転送の終了時にブロックパラメータが更新され（BRSUM/BRDUMの設定に従ってBNDT値が再ロードされ、SAR/DAR値が更新されます）、繰り返しカウンタが1ずつデクリメントされます。
繰り返しブロックカウンタが0に達すると、この最後のブロックは、単一のブロック転送として処理されます。

MDMAのリンクされたリスト･モード

- リンクされたリスト･モードでは、チャネル･リンク･アドレス･レジスタ（CLAR）に指定されたアドレスから、新しいMDMAの設定をロード可能
- この操作後、前述のブロック／繰り返しブロック･モードで定義されたように、チャネルは新しいリクエストを受け付ける準備ができるか、またはトリガ･モードがチャネル･データ全体（TRGM="11"）に設定されている場合は転送を継続
- トリガ･ソースは、トリガおよびバス選択レジスタ（TBR）の値をロードする際に自動的に変更することが可能
- トリガ･モデルがTRGM="11"に設定されている場合、トリガ･モデルと選択したSWリクエストは変更してはいけない

life.augmented

リンクされたリストモードでは、CxLARレジスタに指定されたアドレスから、新しいMDMAの設定（CxTCR、CxBNDTR、CxSAR、CxDAR、CxBRUR、CxLAR、CxTBR、CxMAR、CxMDRレジスタ）をロードできます。このアドレスでは、AXIシステムバス上に配置されたメモリを処理する必要があります。
この操作後、前述のブロック／繰り返しブロックモードで定義されたように、チャネルは新しいリクエストを受け付ける準備ができるか、TRGM[1:0]が"11"の場合は転送を続けます。
トリガソースはCxTBR値をロードする際に自動的に変更することができます。
TRGMおよびSWRMの値は、TRGM[1:0]が"11"の場合は変更しないでください。

MDMAのリンクされたリスト・モード

- チャネル設定（チャネル・リンク・アドレスLAR）はAXIアドレス空間になければいけない
- LAR値は、LAR[2:0]=0x0などのダブルワード・アドレスにアラインされている必要あり

レジスタ（32ビットワード）	リンク・アドレス・レジスタからのオフセット	説明
CTCR	0x00	転送設定レジスタ
CBNDTR	0x04	データのブロック数レジスタ
CSAR	0x08	転送元アドレス・レジスタ
CDAR	0x0C	転送先アドレス・レジスタ
CBRUR	0x10	ブロック繰り返しアドレス更新レジスタ
CLAR	0x14	リンク・アドレス・レジスタ：次のディスクリプタ
CTBR	0x18	トリガおよびバス選択レジスタ
CMAR	0x20	マスク・アドレス・レジスタ
CMDR	0x24	マスク・データ・レジスタ

life.augmented

チャネル設定（チャネルリンクアドレスLAR）はAXIアドレス空間になければなりません。
LAR値は、LAR[2:0]=0x0などのダブルワードアドレスにアラインされている必要があります。

MDMAリクエスト・マッピング

MDMAリクエスト	リクエスト・ソース	説明
mdma_str0 – 7	dma1_tcf0 – 7	DMA1ストリーム0-7転送完了フラグ
mdma_str8 – 15	dma2_tcf0 – 7	DMA2ストリーム0-7転送完了フラグ
mdma_str16	LTDC	LTDCライン割込み
mdma_str17	JPEG	JPEG入力FIFOのしきい値
mdma_str18		JPEG入力FIFOがフルではない
mdma_str19		JPEG出力FIFOのしきい値
mdma_str20		JPEG出力FIFOがフルではない
mdma_str21		JPEG変換終了
mdma_str22	QUADSPI	QUADSPIのFIFOのしきい値
mdma_str23		QUADSPI転送終了
mdma_str24	DMA2D	DMA2D CLUT転送終了
mdma_str25		DMA2D転送終了
mdma_str26		DMA2D電子透かし
mdma_str29	SDMMC1	SDMMC1データの終了

この表は、MDMAリクエストと、デバイスへのそれらのマッピングを示しています。MDMAチャネルは、DMA1ストリームの転送の終了によってトリガできます。このトリガに応答して、MDMAは次のことができます。

- SRAM D2からD1 RAMへのデータ転送を行う。
- または、新しい転送のためにDMA1ゼロフローを再プログラムする。

MDMAによってCPU負荷を効率的に軽減できるようにするには、MDMAチャネルをデバイスの割込みによってトリガして、データ交換と処理を自動化することができます。ペリフェラルのトリガの例をこの表に示します。

割込み

- 各チャネルの割込みイベント

割込みイベント	説明
CTCIF	MDMAチャネル転送完了 これはチャネル·イネーブル·ビットに 0 を書き込むことによってもセットされる
BTIF	MDMAブロック転送完了
BRTIF	MDMAブロック繰り返し転送完了
TCIF	MDMAバッファ転送完了 これはデバッグ機能として使用可能（割込みなし） 最後のフラグリセット以降に（少なくとも）MDMAバッファ転送が生成されたことを示している
TEIF	MDMA転送エラー

life.augmented

各MDMAチャネルに対し、次のイベントにて割込みを生成することができます。

・チャネル転送完了
・ブロック転送完了
・ブロック転送繰り返し完了
・バッファ転送完了
・転送エラー

低電力モードでのMDMA

モード	説明
RUN	アクティブ
SLEEP	アクティブ MDMA割込みはSTM32H7を起動することが可能
STOP	停止 MDMAレジスタの内容は保持（STOPのD1ドメイン）
STANDBY	パワーダウン STANDBYモードを終了した後、MDMAを再初期化する必要あり

MDMAは RUNモードおよびSLEEPモードでアクティブです。DMA割込みはSTM32H7をSLEEPモードから起動します。STOPモードでは、DMAは停止し、DMAレジスタの内容は保持されます。STAMDBYモードでは、DMAはパワーダウンし、STAMDBYモードを終了した後は、DMAレジスタを再初期化する必要があります。