

# Renesas Synergy™ プラットフォーム JPEG デコードフレームワーク モジュールガイド

R11AN0093JU0100  
 Rev.1.00  
 2018.12.12

本資料は英語版を翻訳した参考資料です。本資料の第 6 章までは Renesas Synergy SSP v1.5.0 Users Manual 英語版の第 4 章 Module Guide を参照に翻訳されています。日本語版は、参考用としてご使用のうえ、最新および正式な内容については英語版のドキュメントを参照ください。

## 要旨 (Introduction)

本モジュールガイドは、ユーザがモジュールを効果的に使用してシステムが開発できるようになることを目的としています。このモジュールガイドを習得することで、開発システムへのモジュールの追加とターゲットアプリケーション向けの正確な設定 (configuration) ができ、さらに付属のアプリケーションプロジェクトコードを参照して、効率的なコード記述が行えるようになります。

より詳細な API や、より高度なモジュール使用法を記述した他のアプリケーションプロジェクト例もルネサス WEB サイト (本書末尾の「参考文献」の項を参照) から入手でき、より複雑な設計に役立ちます。

JPEG デコード HAL モジュールは、業界標準の JPEG 画像デコード処理用の高水準 API を提供し、Synergy™ JPEG コードックペリフェラルをサポートしています。JPEG デコードフレームワークモジュールは、ThreadX® RTOS に対応した実装をされており、Synergy MCU 上の Synergy JPEG ハードウェアへのスレッドセーフ (thread-safe) なアクセスを提供します。

## 目次 (Contents)

1. JPEG デコードフレームワークモジュールの機能 (JPEG Decode Framework Module Features) .....	3
2. JPEG デコードフレームワークモジュール API の概要 (JPEG Decode Framework Module APIs Overview) .....	4
3. JPEG デコードフレームワークモジュールの動作の概要 (JPEG Decode Framework Module Operational Overview) .....	5
3.1 JPEG デコードフレームワークモジュールの動作に関する重要な注意事項と制限事項 (JPEG Decode Framework Module Important Operational Notes and Limitations) .....	5
3.1.1 JPEG デコードフレームワークモジュールの動作に関する注意事項 (JPEG Decode Framework Module Operational Notes) .....	5
3.1.2 JPEG デコードフレームワークモジュールの制限事項 (JPEG Decode Framework Module Limitations) .....	6
4. アプリケーションへの JPEG デコードフレームワークモジュールの組み込み (Including the JPEG Decode Framework Module in an Application) .....	6
5. JPEG デコードフレームワークモジュールの構成 (Configuring the JPEG Decode Framework Module) .....	7
5.1 JPEG デコードフレームワークモジュールのクロック構成 (JPEG Decode Framework Module Clock	

Configuration) .....	9
5.2 JPEG デコードフレームワークモジュールの割り込み構成 (JPEG デコードフレームワークモジュールの割り込み構成) .....	9
5.3 JPEG デコードフレームワークモジュールのピン構成 (JPEG Decode Framework Module Pin Configuration) .....	9
6. アプリケーションでの JPEG デコードフレームワークモジュールの使用 (Using the JPEG Decode Framework Module in an Application) .....	9
7. JPEG デコードフレームワークモジュールのアプリケーションプロジェクト (The JPEG Decode Framework Module Application Project) .....	10
8. 対象アプリケーションの JPEG デコードフレームワークモジュールのカスタマイズ (Customizing the JPEG Decode Framework Module for a Target Application) .....	15
9. JPEG デコードフレームワークモジュールのアプリケーションプロジェクトの実行 (Running the JPEG Decode Framework Module Application Project) .....	15
10. JPEG デコードフレームワークモジュールのまとめ (JPEG Decode Framework Module Conclusion) .....	17
11. JPEG デコードフレームワークモジュールの次の手順 (JPEG Decode Framework Module Next Steps) .....	17
12. JPEG デコードフレームワークモジュールの参考情報 (JPEG Decode Framework Module Reference Information) .....	17

## 1. JPEG デコードフレームワークモジュールの機能 (JPEG Decode Framework Module Features)

- Synergy JPEG ハードウェアへのスレッドセーフ(thread-safe)なアクセスを提供。
- JPEG デコード HAL モジュールを使用した JPEG 伸張(decompression)をサポート。
- JPEG デコーダが完了するまでアプリケーションが待機できるようにするポーリングモードをサポート。
- ユーザーが指定したコールバック関数を使用した割り込みモードをサポート。
- 水平および垂直サブサンプル値、水平ストライド、デコードされたピクセルフォーマット、入力および出力データフォーマット、色空間などのパラメータを設定。
- デコード前にイメージサイズの取得が可能。
- デコードされたデータの入力バッファ (input buffer) への入力と、デコードされたイメージフレームの出力バッファへのストア (store) をサポート。
- JPEG デコーダモジュールへのデータのストリーム転送をサポート。この機能により、アプリケーションは、ファイルやネットワークから JPEG イメージのイメージ全体をバッファリングすることなく読み取ることができます。
- デコードするイメージ行数を設定。この機能により、アプリケーションはデコードされたイメージの処理を、フレーム全体をバッファリングすることなく実行可能。
- 入力デコードフォーマットとして YCbCr444、YCbCr422、YCbCr420、YCbCr411 をサポート。
- 出力フォーマットとして ARGB8888 と RGB565 をサポート。
- JPEG イメージのサイズ、高さ、幅が要件を満たさない場合にエラーをリターン。
- JPEG ハードウェアサポートイベントと同期するための、スレッドをサスペンドまたは再開するための wait API 関数をサポート。

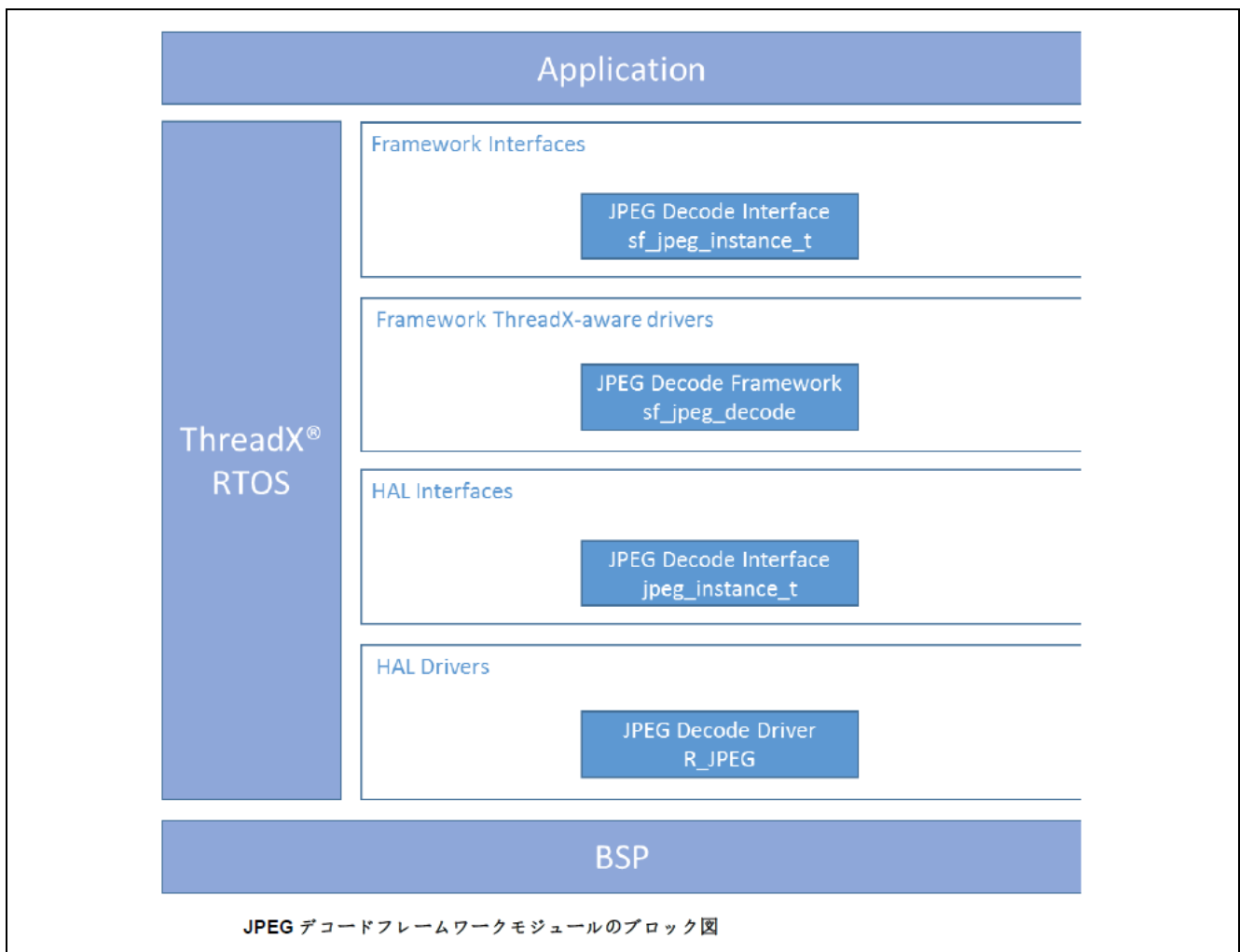


図 1 JPEG デコードフレームワークモジュールのブロック図

## 2. JPEG デコードフレームワークモジュール API の概要 (JPEG Decode Framework Module APIs Overview)

JPEG デコードフレームワークモジュールのすべての API のリスト、API コール例、API の簡単な説明を次の表に示します。ステータス戻り値の表はこの API 要約表の後に示します。

表 1 JPEG デコードフレームワークモジュール API の要約

関数名	API 呼び出しの例と説明
open	<pre>g_sf_jpeg_decode.p_api-&gt;open(g_sf_jpeg_decode.p_ctrl, g_sf_jpeg_decode.p_cfg);</pre> JPEG デコードフレームワークを開きます。
close	<pre>g_sf_jpeg_decode.p_api-&gt;close(g_sf_jpeg_decode.p_ctrl);</pre> JPEG デコードフレームワークを閉じます。
inputBufferSet	<pre>g_sf_jpeg_decode.p_api-&gt;inputBufferSet(g_sf_jpeg_decode.p_ctrl, &amp;buffer, size);</pre> JPEG コーデックにデータを供給します。
outputBufferSet	<pre>g_sf_jpeg_decode.p_api-&gt;outputBufferSet(g_sf_jpeg_decode.p_ctrl, &amp;buffer, size);</pre> JPEG コーデックからデコード済みのデータを読み取ります。
linesDecodedGet	<pre>g_sf_jpeg_decode.p_api-&gt;linesDecodedGet(g_sf_jpeg_decode.p_ctrl, &amp;lines);</pre> デコードされたライン数を取得します。
horizontalStrideSet	<pre>g_sf_jpeg_decode.p_api-&gt;horizontalStrideSet(g_sf_jpeg_decode.p_ctrl, stride);</pre> 垂直方向のストライド値を設定します。
imageSubsampleSet	<pre>g_sf_jpeg_decode.p_api-&gt;imageSubsampleSet(g_sf_jpeg_decode.p_ctrl, h_sub, v_sub);</pre> 水平および垂直のサブサンプル値を設定します。
wait	<pre>g_sf_jpeg_decode.p_api-&gt;wait(g_sf_jpeg_decode.p_ctrl,timeout);</pre> 現在の JPEG コーデック操作が、ThreadX ティックで与えられたタイムアウト値で終了するのを待ちます。
statusGet	<pre>g_sf_jpeg_decode.p_api-&gt;statusGet(g_sf_jpeg_decode.p_ctrl, &amp;status);</pre> JPEG コーデックのステータスを取得します。
imageSizeGet	<pre>g_sf_jpeg_decode0.p_api-&gt;imageSizeGet(g_sf_jpeg_decode0.p_ctrl, p_horizontal, p_vertical);</pre> デコード操作中の画像サイズを取得します。
pixelFormatGet	<pre>g_sf_jpeg_decode0.p_api-&gt;pixelFormatGet(g_sf_jpeg_decode0.p_ctrl, p_color_space);</pre> 入力ピクセル形式を取得します。
versionGet	<pre>g_sf_jpeg_decode.p_api-&gt;versionGet(&amp;version);</pre> バージョンポインタを使用して API バージョンを取得します。

注：関数データ構造体、型定義 (typedef)、定義 (define)、API データ、API 構造体、関数変数の動作と定義の詳細な説明については、『SSP ユーザーズマニュアル』で関連モジュールの API リファレンスを参照してください。

表 2 ステータス戻り値

Name	説明
SSP_SUCCESS	JPEG デコードドライバ (JPEG Decode driver) が正常に開かれました。
SSP_ERR_ASSERTION	アサーションエラーです。
SSP_ERR_IN_USE	モジュールは既に使用中です。
SSP_ERR_TIMEOUT	待機動作がタイムアウトし、ローレベルのドライバ (underlying driver) が時間内に応答しませんでした。
SSP_ERR_WAIT_ABORTED	システム内部エラー (System internal error) が発生しました。

注： ローレベルドライバ (Lower-level drivers) によって、共通のエラーコード (common error codes) を返すことがあります。すべての関連ステータス戻り値の定義については、『SSP ユーザーズマニュアル』で関連モジュールの API リファレンスを参照してください。

### 3. JPEG デコードフレームワークモジュールの動作の概要 (JPEG Decode Framework Module Operational Overview)

JPEG デコードフレームワークモジュールには、標準的な JPEG デコード機能が実装されています。入力バッファのデータを取得し、そのデータに対し JPEG デコードアルゴリズムを適用します。出力は、定義済みの出力バッファの場所に出力されます。JPEG ハードウェアサポートイベント (hardware supported events) と同期するためのスレッドのサスペンドや再開には wait API 関数が使用できます。

#### 3.1 JPEG デコードフレームワークモジュールの動作に関する重要な注意事項と制限事項 (JPEG Decode Framework Module Important Operational Notes and Limitations)

##### 3.1.1 JPEG デコードフレームワークモジュールの動作に関する注意事項 (JPEG Decode Framework Module Operational Notes)

- JPEG エンコードされたデータのデコードは、open API を呼び出すことで開始できます。モジュールを開くには、JPEG デコードフレームワークモジュールインスタンスを使用します。このインスタンスには、API 関数のポインタ、制御ブロック (control block) へのポインタ、ISDE の e2 studio の Synergy プロジェクトコンフィギュレータ (Synergy Project configurator) を通じて生成される静的構成 (static configuration) が含まれています。
- JPEG デコードフレームワークモジュールを停止するには、close API を呼び出します。
- 必要な場合は、入力バッファストリーミングモード (input buffer-streaming mode) が使用できます。
- 必要な場合は、出力バッファストリーミングモード (output buffer-streaming mode) が使用できます。
- 出力データのカラーフォーマット (output data-color formats) として、RGB565 と ARGB888 をサポートします。
- JPEG デコードフレームワークモジュールの制御ブロックにはステータスフラグ (status flag) があり、ユーザは statusGet API を通じてモジュールの現在のステータスを取得できます。また、モジュールで特定のイベントが発生すると、ユーザコールバック関数を通じてステータスがレポートされます。
- JPEG デコードフレームワークモジュールは、入力バッファがソースイメージファイルよりも小さい場合に備えて、入力バッファのバッファストリーミングモード (buffer-streaming mode) をサポートします。ハードウェアで生成された INPUT\_PAUSE 割り込みが発生するたびに、次の入力フレームが入力バッファとして設定されます。
- JPEG デコードフレームワークモジュールは、デコード結果のイメージが出力バッファサイズよりも大きい場合に備えて、出力バッファのバッファストリーミングモード (buffer-streaming mode) をサポートします。ハードウェアで生成された OUTPUT\_PAUSE 割り込みが発生するたびに、次のデータ用のスペースを空けるために出力バッファからデータが読み取られ、格納 (store) されます。
- JPEG デコードフレームワークモジュールが正常に動作するためには、入力および出力バッファが 8 バイトでアライン (8-bytes aligned) されている必要があります。8 バイトのアラインでない場合、API はエラーコードを返します。

### 3.1.2 JPEG デコードフレームワークモジュールの制限事項 (JPEG Decode Framework Module Limitations)

JPEG フレームワークモジュールは、JPEG エンコード処理をサポートしていません。このモジュールの動作に関するその他の制限事項については、最新の SSP リリースノートを参照してください。

## 4. アプリケーションへの JPEG デコードフレームワークモジュールの組み込み (Including the JPEG Decode Framework Module in an Application)

この章では、SSP コンフィギュレータを使用してアプリケーションに JPEG デコードフレームワークモジュールを組み込む方法について説明します。

注： この章を読むには、プロジェクトの作成、スレッドの追加、スレッドへのスタックの追加、およびスタック内でのブロックの構成について理解する必要があります。これらの項目に精通していない場合は、『SSP ユーザーズマニュアル』の最初のいくつかの章を参照して、SSP ベースのアプリケーション作成時の重要な各手順の取り扱い方を修得してください。

JPEG デコードフレームワークモジュールをアプリケーションに追加するには、次の表に示すスタック選択シーケンスを使用してスレッドに単純に追加します。(JPEG デコードフレームワークモジュールのデフォルト名は `g_sf_jpeg_decode0` です。この名前は、対応する [Properties] ウィンドウで変更できます。)

表 3 JPEG デコードフレームワークモジュール選択シーケンス

リソース	ISDE Tab	Stacks Selection Sequence
<code>g_sf_jpeg_decode0</code> JPEG Framework	Threads	New Stack> Framework> Graphics> JPEG Decode Framework on <code>sf_jpeg_decode</code>

次の図に示すように、`sf_jpeg_decode` の JPEG デコードフレームワークモジュールがスレッドスタックに追加されると、コンフィギュレータは必要なローレベルモジュールを自動的に追加します。

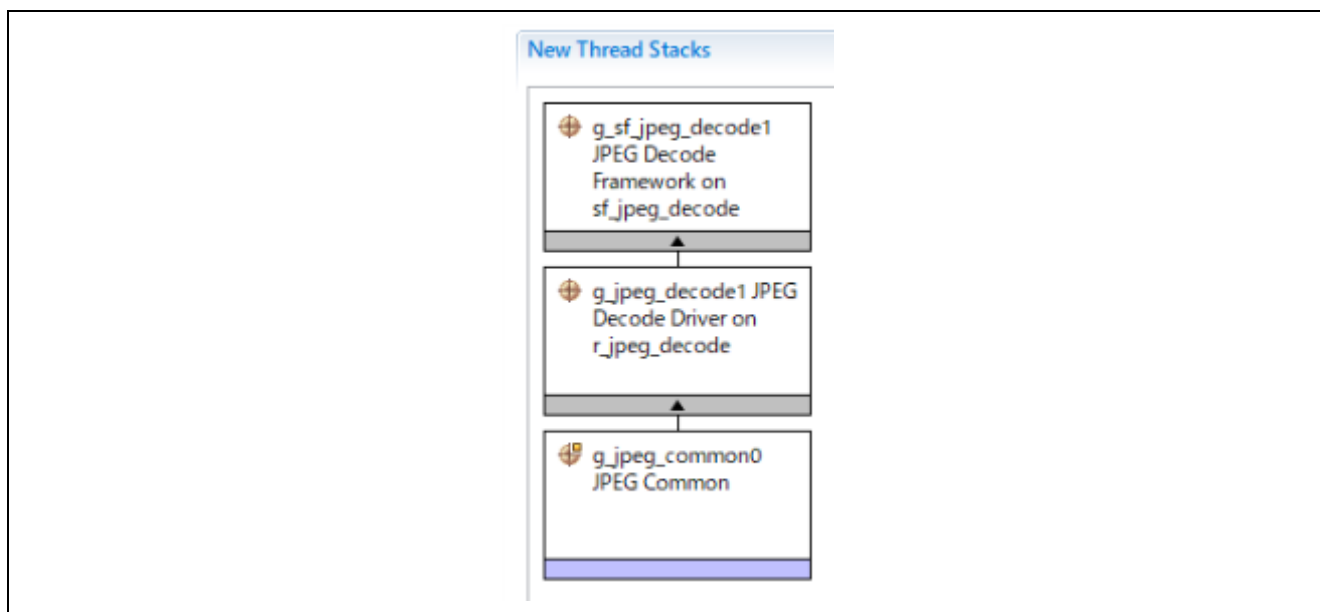


図 2 JPEG デコードフレームワークモジュールのスタック

### 伸張処理割り込み (JEDI)

JPEG 伸張処理割り込み (JPEG decompression-process interrupt) は、次のイベントが検出されたときに発生します。

- 現在の伸張プロセスが正常に完了した。
- 伸張プロセス中にエラーが発生した。
- イメージのサイズとピクセルフォーマットが正常に読み出された。



## データ転送割り込み (JDTI)

JPEG データ転送割り込み (JPEG data-transfer interrupt) は、次のイベントが検出されたときに発生します。

- JPEG コードされたデータがすべて正常に完了した。
- linesDecodedGet で指定された出力イメージデータの行数が転送された。
- inputBufferSet で指定された入力イメージデータの行数が転送された。

## 5. JPEG デコードフレームワークモジュールの構成 (Configuring the JPEG Decode Framework Module)

JPEG デコードフレームワークモジュールでは、ユーザが所望する動作 (operation) のための設定はユーザによって行われる必要があります。ユーザがアクセス可能なすべてのプロパティの利用可能な設定と値は、SSP コンフィギュレータの[Properties]タブに表示され、簡単に参照できるよう次の表に示されています。競合 (コンフリクト) を起こさずに変更できるプロパティのみが変更可能です。他のプロパティはロックされており、変更はできません。また、ISDE の[Properties]ウィンドウでロックされたプロパティのロックアイコンで識別されます。このアプローチは、設定プロセスを簡素化し、設定に対する従来の手動によるアプローチよりもエラーが起こりにくくします。ユーザがアクセス可能なすべてのプロパティの使用可能な設定とデフォルト値は、SSP コンフィギュレータ内の[Properties]タブに表示され、簡単に参照できるよう次の表に示されます。

注： ISDE を開き、JPEG デコードフレームワークを作成し、次の表に示す構成テーブル設定を確認すると並行してプロパティ設定を調べることができます。SSP を使用した開発に有効です。

表 3 JPEG Decode Framework on sf\_jpeg\_decode モジュールの設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled Default: Default (BSP)	パラメータエラーチェックを有効または無効にします。
Name	Default: g_jpeg_decode0	JPEG デコードモジュールインスタンスに使用する名前。

注：設定例とデフォルトは、Synergy S7G2 MCU ファミリーを使用しているプロジェクトのものであります。他の MCU では、異なるデフォルト値と設定の場合があります。

### JPEG デコードフレームワークモジュールの下位レベルモジュールの設定

通常、ローレベルドライバのデフォルト設定から変更する必要がある設定項目は少なく、これらはスレッドスタックブロックに赤いテキストで表示されます。一部の設定プロパティは、フレームワークが適切に作動するために特定の値に設定されており、それらはロックされていてユーザは変更できません。次の表に、ローレベルモジュールの [Properties] セクションのすべての設定を示します。

表 4 r\_jpeg 上の JPEG HAL モジュールの設定

ISDE のプロパティ	Value	説明
Parameter Checking	Default (BSP), Enabled, Disabled Default: Default (BSP)	パラメータエラーチェックを有効または無効にします。
Name	Default: g_jpeg_decode0	JPEG デコードモジュールインスタンスに使用する名前。
Byte Order for Input Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8), Byte Swap (2)(1)(4)(3)(6)(5)(8)(7), Word Swap (3)(4)(1)(2)(7)(8)(5)(6),	入力データのバイト順序を指定します。順序は 8 バイトごとに指定に従ってスワップされます。

ISDE のプロパティ	Value	説明
	Word-Byte Swap (4)(3)(2)(1)(8)(7)(6)(5),  Longword Swap (5)(6)(7)(8)(1)(2)(3)(4),  Longword-Byte Swap (6)(5)(8)(7)(2)(1)(4)(3),  Longword-Word Swap (7)(8)(5)(6)(3)(4)(1)(2),  Longword-Word Byte Swap (8)(7)(6)(5)(4)(3)(2)(1)  Default: Normal Byte order (1)(2)(3)(4)(5)(6)(7)(8)	
Byte Order for Output Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8),  Byte Swap (2)(1)(4)(3)(6)(5)(8)(7), Word Swap (3)(4)(1)(2)(7)(8)(5)(6),  Word-Byte Swap (4)(3)(2)(1)(8)(7)(6)(5),  Longword Swap (5)(6)(7)(8)(1)(2)(3)(4),  Longword-Byte Swap (6)(5)(8)(7)(2)(1)(4)(3),  Longword-Word Swap (7)(8)(5)(6)(3)(4)(1)(2),  Longword-Word Byte Swap (8)(7)(6)(5)(4)(3)(2)(1)  Default: Normal Byte order (1)(2)(3)(4)(5)(6)(7)(8)	出力データのバイト順序を指定します。順序は 8 バイトごとに指定に従ってスワップされます。
Output Data Color Format	Pixel Data RGB565 format, Pixel Data ARGB8888 format  Default: Pixel Data RGB565 format	出力データフォーマットを指定します。
Alpha value to be applied to decoded pixel data (only valid for ARGB8888 format)	Default: 255	出力データフォーマットのアルファ値を指定します (ARGB8888 フォーマットに対してのみ有効)。
Name of user callback function	(Locked) NULL	ユーザコールバック関数の名前を指定します。フレームワークで使用するためロックされています。
Decompression Interrupt Priority	Priority 0 (highest), Priority 1,2,3,4,5,6,7,8,9,10,11,12,13,14,	デコンプレッション割り込みの優先順位を選択。



ISDE のプロパティ	Value	説明
	Priority15 (lowest, not valid if using Thread X),  Default: Priority 12	
Data Transfer Interrupt priority	Priority 0 (highest), Priority1,2,3,4,5,6,7,8,9,10,11,12,13,14, Priority15 (lowest, not valid if using Thread X),  Default: Priority 12	データ転送割り込み優先順位の選択

注：設定例とデフォルトは、Synergy S7G2 MCU ファミリを使用しているプロジェクトのものです。他の MCU では、異なるデフォルト値と設定の場合があります。

表 5 JPEG コモンモジュールの設定

ISDE のプロパティ	Value	説明
Name	Default: g_jpeg_common0	モジュール名

注：設定例とデフォルトは、Synergy S7G2 MCU ファミリを使用しているプロジェクトのものです。他の MCU では、異なるデフォルト値と設定の場合があります。

## 5.1 JPEG デコードフレームワークモジュールのクロック構成 (JPEG Decode Framework Module Clock Configuration)

JPEG フレームワークモジュールは、周辺モジュールクロック A (PCLKA) を使用して実行します。

## 5.2 JPEG デコードフレームワークモジュールの割り込み構成 (JPEG デコードフレームワークモジュールの割り込み構成)

割り込みを有効にするには、ISDE の JPEG デコードフレームワークモジュールの [Properties] ウィンドウで伸張割り込み(decompression interrupt)とデータ転送割り込み(data-transfer interrupt)のプライオリティを設定します。

## 5.3 JPEG デコードフレームワークモジュールのピン構成 (JPEG Decode Framework Module Pin Configuration)

JPEG デコードフレームワークモジュールはピン (ポート) を使用しません。

## 6. アプリケーションでの JPEG デコードフレームワークモジュールの使用 (Using the JPEG Decode Framework Module in an Application)

アプリケーションで JPEG デコードフレームワークモジュールを使用する際の一般的な手順は次のとおりです。

1. open API を使用して JPEG デコードペリフェラルを初期化します。
2. imageSubSampleSet API を使用してイメージサブサンプルを設定します。
3. horizontalStrideSet API を使用して水平ストライドを設定します。
4. outputBufferSet API を使用して出力バッファを設定します。
5. inputBufferSet API を使用して入力バッファを設定します。
6. wait API を使用してデコードの完了を待ちます。
7. statusGet API を使用してデコードステータスを確認します。
8. close API を使用してインスタンスを閉じます (必要な場合)。

これらの一般的な手順を、次の図の通常の動作フロー図に示します。

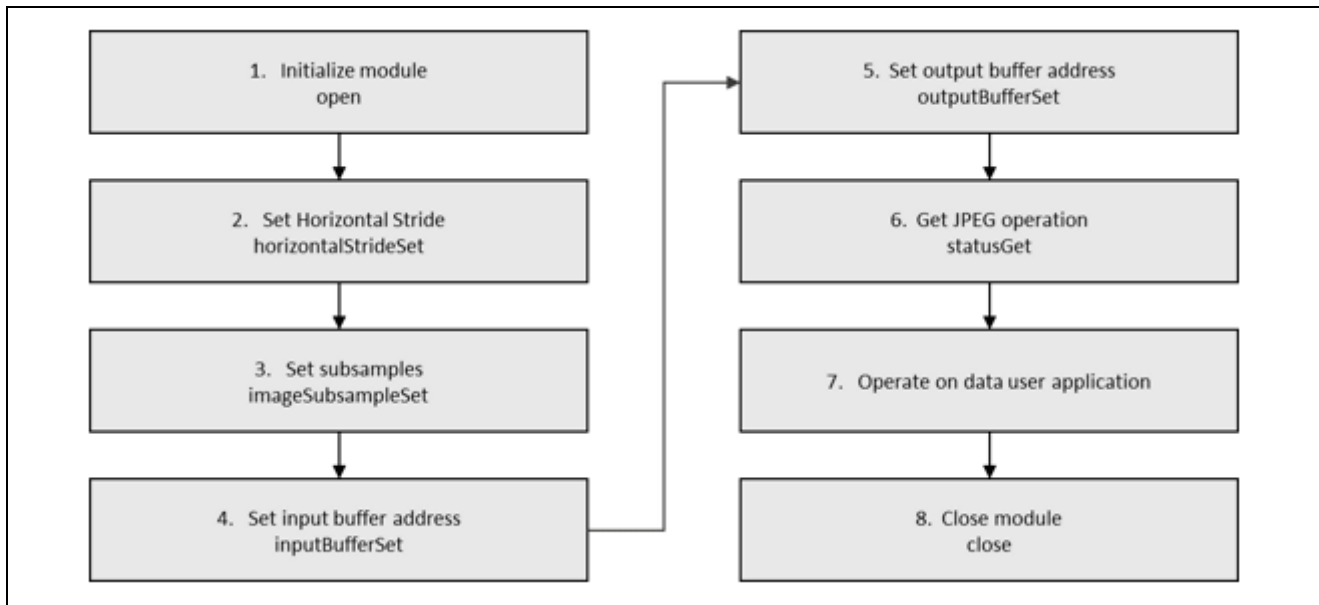


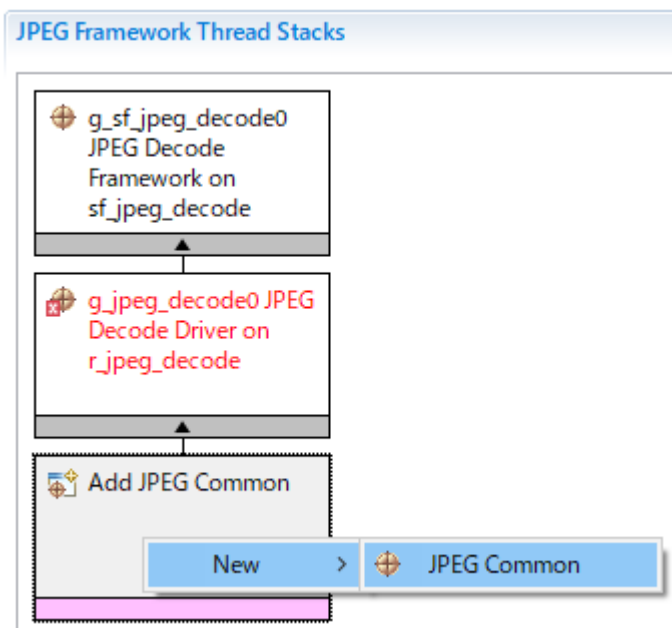
図 3 通常の JPEG HAL およびフレームワークモジュールアプリケーションのフロー図

## 7. JPEG デコードフレームワークモジュールのアプリケーションプロジェクト (The JPEG Decode Framework Module Application Project)

注：

このアプリケーションプロジェクトは、**SSP V1.2.0**対応です。**SSP V1.5.0**で使用するときには**e2studio ISDE**の機能で**SSP v1.5.0**にマイグレートされます。また、次の操作を行ってください。

- ①下図のように、[JPEG Framework Thread Stacks]で[Add JPEG Common]を右クリックし [New]>[JPEG Common]を選択してください。



② 下図のjpeg\_decode\_framework\_thread\_entry.c 内の"jpeg\_Decode\_HAL\_run();"をコメントにしてください。

```

40  #ifndef SEMI_HOSTING
41  #ifdef __GNUC__
42      initialise_monitor_handles();
43  #endif
44  #endif
45
46  // jpeg_Decode_HAL_run();
47  tx_thread_sleep (10);
48  jpeg_Decode_Framework_run();
49
50  while (1)
51  {
52      tx_thread_sleep (1);
53  }
54
55

```

以下の内容は、JPEG Framework ModuleとJPEG HAL Driverを組み合わせて説明しています。

このモジュールガイドで説明するアプリケーションプロジェクトを使用して、実際の開発手順を体験することができます。このプロジェクトは、このドキュメントの末尾にある「参考情報」の章に掲載されているリンクから入手します。ユーザは ISDE でアプリケーションプロジェクトをインポートして開き、JPEG デコードフレームワークモジュールに対応する設定項目 (configuration setting) を表示することができます。さらに jpeg\_decodeimage\_mg.c/.h、jpeg\_decode\_hal\_api\_mg.h、jpeg\_decode\_framework\_api\_mg.h、jpeg\_decode\_framework\_thread\_entry.c 内のコードを確認することもできます。これらのファイルは、設計全体における JPEG デコードフレームワーク API を説明します。

このアプリケーションプロジェクトは、JPEG デコードフレームワーク API の一般的な使用方法を示します。このアプリケーションプロジェクトのメインスレッドのエントリ部分は、JPEG デコード HAL と複数のフレームワークモジュールをそれぞれのエントリ関数で初期化し、sample\_image1.c と sample\_image2.c に格納されている 2 種類のイメージをデコードします。デバッグ動作中に、ISDE のメモリタブで出力を表示できます。JPEG デコード中にイベントが発生した場合、ユーザコールバック関数を実行します。ユーザ指定のコールバック関数は、プログラムを実行するためのフラグを複数設定し、動作の流れを決定します。次の表は、このアプリケーションプロジェクトが使用する関連ソフトウェアとハードウェアの対象バージョンを示します。

表 5 このアプリケーションプロジェクトが使用するソフトウェアとハードウェアのリソース

リソース	リビジョン	説明
e² studio	5.3.1 またはそれ以降	統合ソリューション開発環境 (ISDE)
SSP	1.2.0 またはそれ以降	Synergy ソフトウェアパッケージ
IAR EW for Synergy	7.71.2 またはそれ以降	IAR Embedded Workbench® for Renesas Synergy™
SSC	5.3.1 またはそれ以降	Synergy Standalone Configurator
SK-S7G2	v3.0 と v3.1	スタータキット

次の図に、このアプリケーションプロジェクトの基本的なフローを示します。

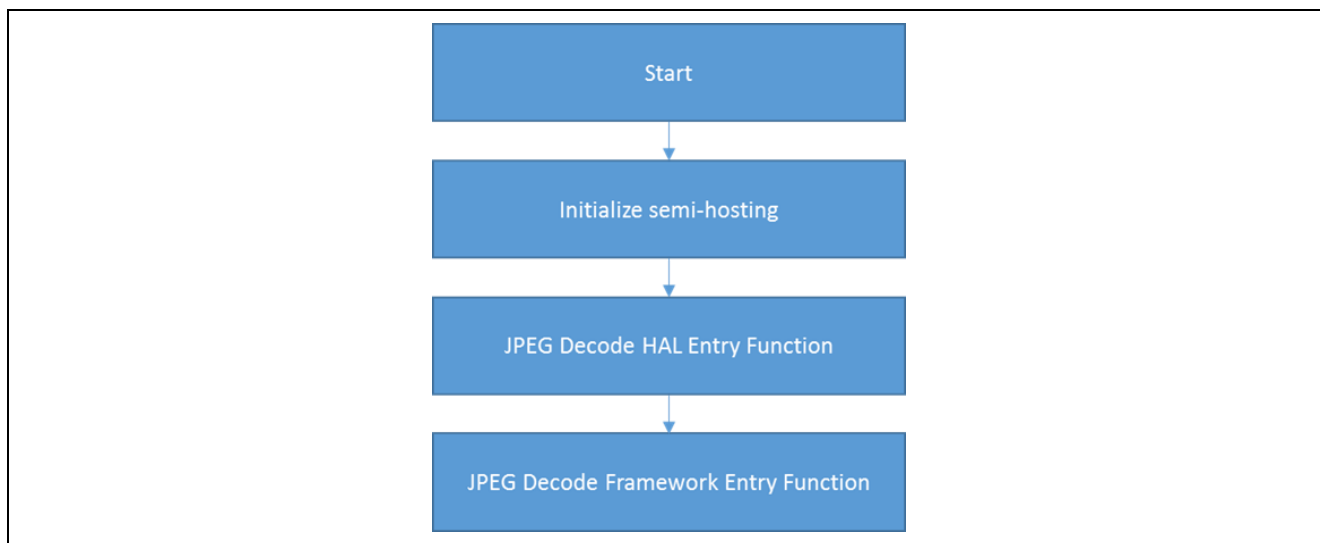


図 4 JPEG デコードフレームワークモジュールのアプリケーションプロジェクトのフロー図

アプリケーションプロジェクト一式は、このドキュメントの末尾にある「参考情報」の章に掲載されているリンクから入手します。jpeg\_decodeimage\_mg.c/.h、jpeg\_decode\_hal\_api\_mg.h、jpeg\_decode\_framework\_api\_mg.h、jpeg\_decode\_framework\_thread\_entry.c の各ファイルは、このプロジェクトを一旦 ISDE にインポートすると、プロジェクト内に格納されます。ISDE でこれらのファイルを開くことができますし、API の主な使い方を確認するための説明も参照できます。各ファイルの内容に関する説明も記述されています。

jpeg\_decode\_hal\_api\_mg.h ファイルは、JPEG デコード用の HAL API を使用するインライン関数を記述したヘッダファイルです。同様に、jpeg\_decode\_framework\_api\_mg.h は、このモジュールガイドの最初の表で記載したフレームワーク API を使用するインライン関数を記述したヘッダファイルです。

sample\_image1.c and sample\_image2.c イメージファイルは、16 進配列フォーマット (hexadecimal array format) で保存済みです。JPEG イメージを 16 進の C 配列に変換するには、拡張されたダンプ (dump) とコードのユーティリティソフトウェアツールを使用します (リンクは「参考情報」の章に掲載されています)。sample\_image1.c データは HAL API によって伸張されており、sample\_image2.c データはこのこのアプリケーションプロジェクト内のフレームワークデコードアプリケーションによって伸張されています。

jpeg\_decode\_image\_mg.h ファイルは、コールバック関数が使用する関数プロトタイプ、マクロ、フラグ宣言を記述しています。

jpeg\_decode\_image\_mg.c ファイルは、JPEG イメージをデコードするために HAL とフレームワーク API を使用するアプリケーションコードを記述しています。このファイルは、JPEG デコードイベントが発生した際に複数のフラグを設定するコールバック関数も記述しています。このファイルは、HAL とフレームワークモジュールを使用して JPEG を伸張するコードを記述しています。jpeg\_Decode\_HAL\_run() 関数は、JPEG HAL デコードのエントリポイントです。次の図で、この関数のフローを説明します。



図 5 JPEG デコードフレームワークモジュールのアプリケーションプロジェクトのフロー図

jpeg\_decode\_framework\_run() 関数は、JPEG HAL デコードのエントリポイントです。次の図で、この関数のフローを説明します。



図 6 JPEG デコードフレームワークモジュールのアプリケーションプロジェクトのフロー図

注： この説明は、Synergy ソフトウェアパッケージ内のデバッグコンソールで printf() を使用する方法をユーザが理解していることを想定しています。このような経験がない場合、このドキュメントの末尾にある「参考情報」の章で紹介しているナレッジベースの記事、「How do I Use Printf() with the Debug Console in the Synergy Software Package」(Synergy ソフトウェアパッケージのデバッグコンソールで Printf() を使用する方法)を参照してください。代わりに、デバッグモードで変数ウォッチ機能を使用して結果を表示することもできます。



対象ボードや MCU の必要な動作と物理プロパティをサポートするために、このアプリケーションプロジェクトではいくつかの重要なプロパティを設定しています。次の表に、それらのプロパティと、このプロジェクトで設定した値を示します。練習として、このアプリケーションプロジェクトを開き、[Properties] ウィンドウでこれらの設定を表示することもできます。

表 6 アプリケーションプロジェクトに対応する JPEG デコードフレームワークモジュールの設定項目

リソース	ISDE のプロパティ	設定済みの値
g_sf_jpeg_decode0 JPEG Decode Framework on r_jpeg_decode	Parameter Checking	Enable
g_jpeg_decode0 JPEG Decode Driver on r_jpeg_decode	Parameter Checking	Enabled
	Name	g_jpeg_decode0
	Byte Order for Input Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8)
	Byte Order for Output Data Format	Normal byte order (1)(2)(3)(4)(5)(6)(7)(8)
	Output Data Color Format	Pixel Data RGB565 format
	Alpha value to be applied to decoded pixel data (only valid for ARGB8888 format)	255
	Name of user callback function	(Locked) NULL
	Decompression Interrupt Priority	Priority 3
	Data Transfer Interrupt priority	Priority 3

## 8. 対象アプリケーションの JPEG デコードフレームワークモジュールのカスタマイズ (Customizing the JPEG Decode Framework Module for a Target Application)

いくつかの設定項目は通常、アプリケーションプロジェクトが示された値に対し、ユーザが変更を加えます。たとえば、ユーザは入出力バッファのバイト順序や、出力データフォーマット、デコード後のピクセルデータに適用するアルファ値を容易に変更できます。また、ユーザは API を使用して、水平ストライド値 (horizontal stride value)、イメージのサブサンプル値 (image subsample value)、入出力イメージサイズ (input/output image size) を変更することもできます。

## 9. JPEG デコードフレームワークモジュールのアプリケーションプロジェクトの実行 (Running the JPEG Decode Framework Module Application Project)

ISDE にこのプロジェクトをインポートし、コンパイルしてデバッグを実行するだけで、JPEG フレームワークモジュールのアプリケーションプロジェクトを実行させ、対象キットでその動作を観察することができます。「Renesas Synergy プロジェクトのインポート」を参照してください。このパッケージに付属している 11an0023eu0116-synergy-ssp-import-guide.pdf で、ISDE または IAR EW for Synergy にプロジェクトをインポートし、アプリケーションをビルド/実行する方法に関する手順を参照しています。

注： Synergy 開発プロセスの基本的な流れを経験したことのあるユーザにとって、以下の手順は十分な内容になっていますが、これらの手順をまだ理解していない場合、これらの手順を説明した、『SSP ユーザーズマニュアル』を参照してください。

## Renesas Synergy™ プラットフォーム JPEG デコードフレームワークモジュールガイド

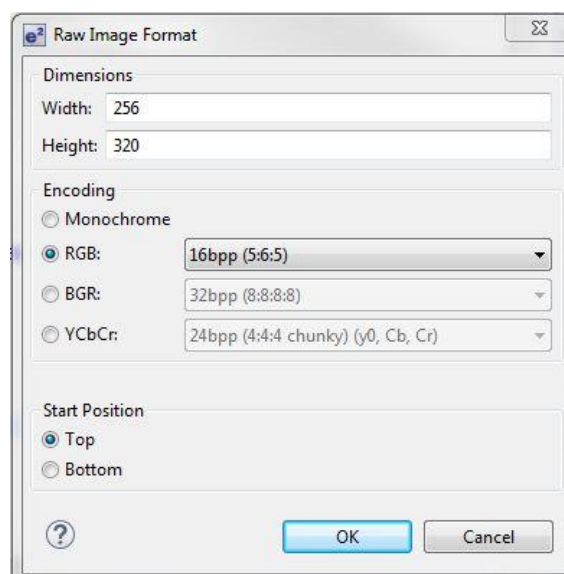
JPEG デコードフレームワークモジュールのアプリケーションプロジェクトを作成し、実行するために、以下の手順に従ってください。

1. ISDE または IAR EW for Synergy に、付属のサンプルプロジェクト JPEG\_Decode\_MG\_AP をインポートします。サンプルプロジェクトをインポートする手順について、「Renesas Synergy プロジェクトのインポート (Importing a Renesas Synergy Project)」を参照してください。
2. アプリケーションをコンパイルします。
3. micro USB ケーブルを SK-S7G2 スタータキットの J19 につなぎ、ホスト PC に接続します。
4. アプリケーションのデバッグを開始します。
5. 通信中は LED1-3 が定期的に点滅します。デコード後のイメージを表示するには、以下の手順に従ってください。
  - A. サンプルアプリケーションをデバッグモードで実行しているとき、次の図に示す [Memory] タブに移動し、[+] 記号をクリックして jpeg\_decoded\_img1 と jpeg\_decoded\_img2 のアドレス位置を追加します。(出力バッファの開始アドレス)。



- B. 各メモリアドレスの [New Renderings] タブをクリックし、[Raw Image] オプションをダブルクリックして、[Raw Image Format] ボタンをクリックします。

Raw Image Format...



- C. 次の図に表示されている情報を入力します。
- D. 次の各出力を観察します (それぞれ output 1 と output 2)。

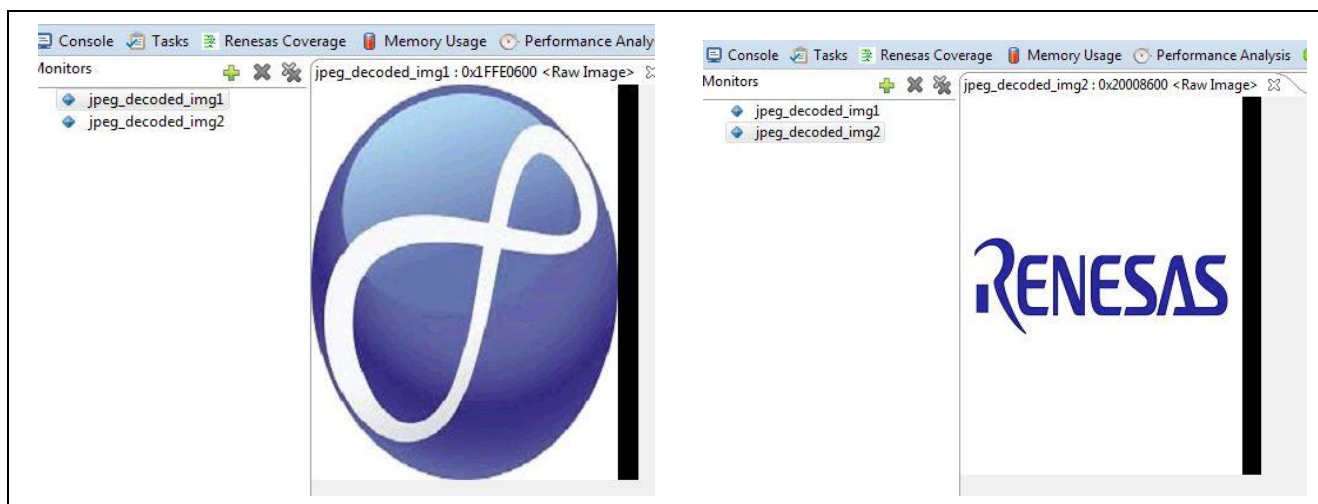


図 7 JPEG デコードフレームワークモジュールのアプリケーションプロジェクトのサンプル出力

## 10. JPEG デコードフレームワークモジュールのまとめ (JPEG Decode Framework Module Conclusion)

このモジュールガイドは、サンプルプロジェクトで JPEG フレームワークモジュールの選択、追加、設定、使用を行うために必要な背景となる情報全般を説明しました。従来の組み込みシステムでは、これらの手順を理解することに多くに時間を必要とし、また間違いが起りやすい操作でした。Renesas Synergy プラットフォームにより、これら手順の所要時間が短くなり、設定項目の競合や、ローレベルドライバの誤った選択など、誤りが防止できるようになりました。アプリケーションプロジェクトで示したように、高レベル API を使用することで高いレベルの開発からスタートし、ローレベルドライバを作成するような従来の開発環境で必要とされる時間が不要になり、開発時間を短縮できます。

## 11. JPEG デコードフレームワークモジュールの次の手順 (JPEG Decode Framework Module Next Steps)

JPEG デコードプロジェクトをマスターした後、API を使用して、JPEG ファイルをデコードし、伸張したイメージを LCD で表示するサンプルを作成することができます。また、ストリーミング入出力バッファ、高度な伸張設定、他のカラーフォーマットのイメージなどと組み合わせて、このサンプルを使用する方法も試してください。

## 12. JPEG デコードフレームワークモジュールの参考情報 (JPEG Decode Framework Module Reference Information)

『SSP ユーザーズマニュアル』：SSP distribution package の一部として html 形式を入手できるほか、Synergy WEB のソフトウェア、SSP のパート (<https://www.renesas.com/jp/ja/products/synergy/software/ssp.html>) から pdf を入手することもできます。

本プロジェクト (ソフトウェア) は、以下のサイトから入手可能です。

<https://www.renesas.com/jp/ja/software/D6001420.html>

また、本モジュールの参考資料やリソースに関する最新の全ての情報は、以下の Synergy WEB のサポートコーナーの、ドキュメント、ナレッジベースから入手可能です。

<https://www.renesas.com/jp/ja/products/synergy/support.html>

## ホームページとサポート窓口

サポート: <https://synergygallery.renesas.com/support>

テクニカルサポート:

- アメリカ: [https://renesas.zendesk.com/anonymous\\_requests/new](https://renesas.zendesk.com/anonymous_requests/new)
- ヨーロッパ: <https://www.renesas.com/en-eu/support/contact.html>
- 日本: <https://www.renesas.com/ja-jp/support/contact.html>

すべての商標および登録商標はそれぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.12.12	-	<ul style="list-style-type: none"><li>第 1.00 版 発行</li><li>第 2 章～6 章：SSP v1.5.0 UM 英文版から和訳</li><li>第 x7 章～：英文版（資料番号 r11an0093eu0100-synergy-jpeg-decode-fw-mod-guide Rev1.00、発効日 2017 年 8 月 1 日）を翻訳</li><li>第 12 章：シナジーナレッジデータベースの HTTP 変更</li></ul>

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
  5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、  
家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、  
金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じて、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシートにおいて高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
  9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
  11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>