

RZ/A2Mグループ

JPEGコーデックユニット(JCU) サンプルドライバ

要旨

本アプリケーションノートでは、RZ/A2Mの JPEG コーデックユニット機能を使用し、JPEG 圧縮画像のデコードおよび JPEG 圧縮画像へのエンコードを行うサンプルドライバについて説明します。

JPEG コーデックユニット(JCU) サンプルドライバの特長を以下に示します。

- ・ JPEG 圧縮画像を、RGB565、ARGB8888、YCbCr422 形式の画像に変換します。
- ・ YCbCr422 形式の画像を、JPEG 圧縮画像に変換します。

動作確認デバイス

RZ/A2M

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分に評価してください。

制限事項

カウントモード(分割処理)は、使用禁止とします。もし使用する場合は、十分に評価してください。

目次

1. 仕様	3
2. 動作確認条件	4
3. ソフトウェア説明	5
3.1 動作概要	5
3.2 使用割込み一覧	6
3.3 定数／型／クラス／関数	6
3.4 移植ガイド	7
4. 関数一覧	8
4.1 R_JCU_Initialize	9
4.2 R_JCU_TerminateAsync	9
4.3 R_JCU_SelectCodec	9
4.4 R_JCU_SetPauseForImageInfo	10
4.5 R_JCU_SetErrorFilter	10
4.6 R_JCU_StartAsync	10
4.7 R_JCU_GetAsyncError	10
4.8 R_JCU_ContinueAsync	11
4.9 R_JCU_SetDecodeParam	11
4.10 R_JCU_GetImageInfo	11
4.11 R_JCU_SetEncodeParam	11
4.12 R_JCU_SetQuantizationTable	12
4.13 R_JCU_SetHuffmanTable	12
4.14 R_JCU_GetEncodedSize	13
4.15 R_JCU_OnInterrupting	13
4.16 R_JCU_OnInitialize	13
4.17 R_JCU_OnFinalize	13
4.18 R_JCU_EnableInterrupt	14
4.19 R_JCU_DisableInterrupt	14
5. ドライバのインポート方法	15
5.1 e ² studio	15
5.2 e ² studio以外で作成されたプロジェクトの場合	15
6. 参考ドキュメント	16
改訂記録	17

1. 仕様

表1.1に使用する周辺機能と用途を、図1.1にサンプルコード実行時の動作環境を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
JPEG コーデックユニット(JCU)	画像データの変換
割込みコントローラ(INTC)	画像データ変換の終了通知に使用
FIFO 内蔵シリアルコミュニケーションインターフェース(SCIF) Ch(2)	メッセージ表示

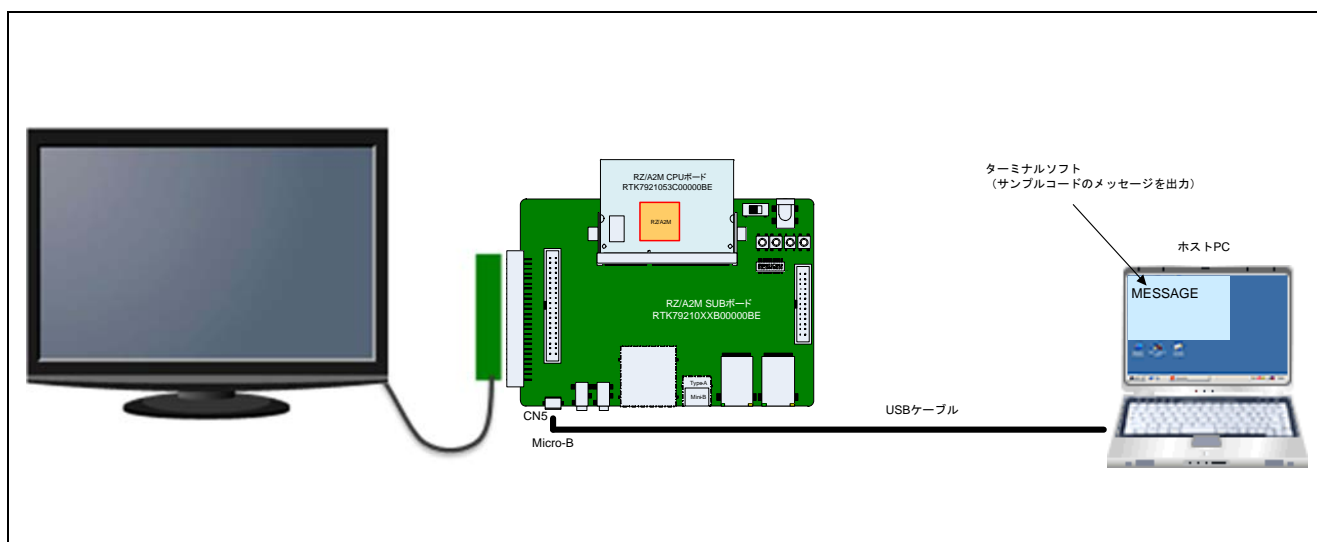


図1.1 動作環境

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用 MCU	RZ/A2M
動作周波数 (注)	CPU クロック (I ϕ) : 528MHz 画像処理クロック (G ϕ) : 264MHz 内部バスクロック (B ϕ) : 132MHz 周辺クロック 1 (P1 ϕ) : 66MHz 周辺クロック 0 (P0 ϕ) : 33MHz QSPI0_SPCLK : 66MHz CKIO : 132MHz
動作電圧	電源電圧 (I/O) : 3.3V 電源電圧 (1.8/3.3V 切替 I/O (PVcc_SPI)) : 3.3V 電源電圧 (内部) : 1.2V
統合開発環境	e2 studio V7.4.0
C コンパイラ	GNU Arm Embedded Toolchain 6-2017-q2-update コンパイラオプション (ディレクトリパスの追加は除く) Release: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Os -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -Wstack-usage=100 -fabi-version=0 Hardware Debug: -mcpu=cortex-a9 -march=armv7-a -marm -mlittle-endian -mfloat-abi=hard -mfpu=neon -mno-unaligned-access -Og -ffunction-sections -fdata-sections -Wunused -Wuninitialized -Wall -Wextra -Wmissing-declarations -Wconversion -Wpointer-arith -Wpadded -Wshadow -Wlogical-op -Waggregate-return -Wfloat-equal -Wnull-dereference -Wmaybe-uninitialized -g3 -Wstack-usage=100 -fabi-version=0
動作モード	ブートモード 3 (シリアルフラッシュブート 3.3V 品)
ターミナルソフトの通信設定	<ul style="list-style-type: none"> 通信速度 : 115200bps データ長 : 8 ビット パリティ : なし ストップビット長 : 1 ビット フロー制御 : なし
使用ボード	RZ/A2M CPUボード RTK7921053C00000BE RZ/A2M SUBボード RTK79210XXB00000BE
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none"> シリアルフラッシュメモリ (SPI マルチ I/O バス空間に接続) メーカー名 : Macronix 社、型名 : MX25L51245GXD RL78/G1C (USB 通信とシリアル通信を変換し、ホスト PC との通信に使用) LED1

【注】 クロックモード 1 (EXTAL 端子からの 24MHz のクロック入力) で使用時の動作周波数です。

3. ソフトウェア説明

3.1 動作概要

図3.1に同期関数を使用して画像データを変換するシーケンスを示します。

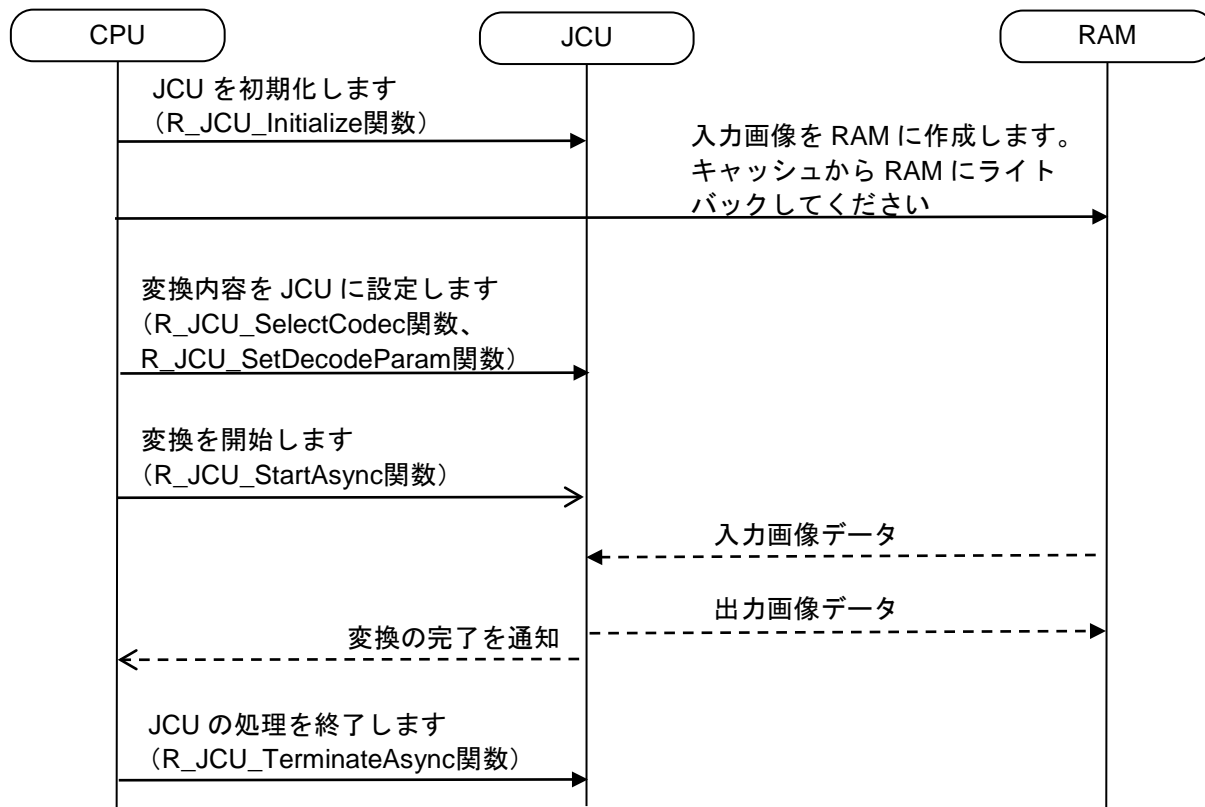


図3.1 JCU のシーケンス

このサンプルプログラムは「JPEG 画像の復号処理」(R_JCU_SampleDecode 関数)と「JPEG 画像の復号と符号化処理」(R_JCU_SampleDecodeEncode 関数)および「JPEG 画像の復号後、表示」(R_JCU_SampleDecodeAndShow 関数)の3種類で構成されています。

3.2 使用割込み一覧

表3.1にサンプルコードで使用する割込みを示します。

表3.1 サンプルコードで使用する割込み

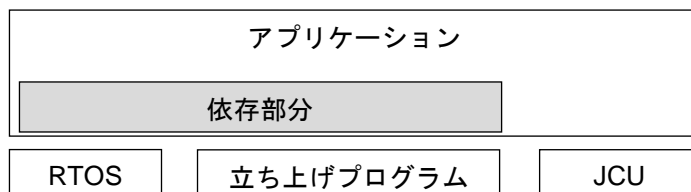
割込み(要因 ID)	優先度	処理概要
JEDI	JCU_INT_PRIORITY(=30)	圧縮伸長処理
JDTI	JCU_INT_PRIORITY(=30)	データ転送処理

3.3 定数／型／クラス／関数

プロジェクトに付属の HTML ファイルを参照してください。

3.4 移植ガイド

JCU が動作する環境から、RTOS や立ち上げプログラムを変更するときは、アプリケーションの中の RTOS や立ち上げプログラムに依存する部分を変更する必要があります。



JCU の非同期処理を開始する API 関数に指定するコールバック関数は、RTOS に依存します。付属のサンプル アプリケーションは、while 文でポーリングしており、CPU 使用率が 100%になってしまいます。

```

is_event = false;

e= R_JCU_StartAsync( (r_co_function_t) gs_SetTrue, &is_event );
if(e){goto fin;}
while ( ! is_event )
{ } /* Pooling */
e= R_JCU_GetAsyncError(); if(e){goto fin;}
  
```

CPU 使用率が 100%になることを避けるには、RTOS のバイナリー セマフォ、イベント グループ、スレッド付属イベントなどを使用するように、以下の箇所について変更してください。

- 非同期処理関数の引数に gs_SetTrue を指定する代わりに、RTOS の待ちを解除する API 関数を指定する。この関数は、割込み、またはスレッドから呼ばれる
- 非同期処理関数の引数に &is_event を指定する代わりに、RTOS の同期オブジェクトを指定する
- ポーリングする while 文を、RTOS の待ちを行う API 呼び出しに変更する
- R_JCU_StartAsync に指定する同期オブジェクトと、R_JCU_TerminateAsync に指定する同期オブジェクトは、別の同期オブジェクトを指定してください。 R_JCU_StartAsync と R_JCU_ContinueAsync は同じ同期オブジェクトでも構いません

ミドルウェアから呼び出す場合、ミドルウェアの移植層を使用するように、以下の箇所について変更してください。

- 非同期処理関数の引数に gs_SetTrue を指定する代わりに、RTOS の待ちを解除する関数を抽象化した移植層の関数を指定する。この関数は、割込み、またはスレッドから呼ばれる
- 非同期処理関数の引数に &is_event を指定する代わりに、ミドルウェアのインスタンスを表すポインタを指定する
- ポーリングする while 文を、RTOS の待ちを行う関数を抽象化した移植層の関数呼び出しに変更する。その引数には、ミドルウェアのインスタンスを表すポインタを指定する
- R_JCU_StartAsync に指定する関数と、R_JCU_TerminateAsync に指定する関数は、別の関数を指定してください。 R_JCU_StartAsync と R_JCU_ContinueAsync は同じ関数でも構いません

RAM のミラー領域や物理アドレスは、立ち上げプログラムの MMU の設定に依存します。 ポインタに格納するアドレスは CPU の仮想アドレスですが、JCU ハードウェアがアクセスするアドレスは、物理アドレスであり、対応関係は MMU の設定に依存します。

4. 関数一覧

表 4-1 API 関数一覧

章	関数名	概要
4.1	R_JCU_Initialize	JCU ドライバを初期化します。
4.2	R_JCU_TerminateAsync	JCU ドライバを終了します。(非同期処理)
4.3	R_JCU_SelectCodec	圧縮または伸長を選択します。
4.4	R_JCU_SetPauseForImageInfo	画像情報取得可能になったら、一時停止するかどうかを選択します。
4.5	R_JCU_SetErrorFilter	有効とするデコードエラー検出の種類(jcu_int_detail_error_t)を設定する。
4.6	R_JCU_StartAsync	JCU の動作を開始します。(非同期処理)
4.7	R_JCU_GetAsyncError	非同期処理の中で発生したエラーを返します。
4.8	R_JCU_ContinueAsync	一時停止した JCU の動作を再開します。(非同期処理)
4.9	R_JCU_SetDecodeParam	デコード用のパラメータを設定します。
4.10	R_JCU_GetImageInfo	JPEG ファイルの画像情報を取得します。
4.11	R_JCU_SetEncodeParam	エンコード用のパラメータを設定します。
4.12	R_JCU_SetQuantizationTable	量子化テーブルを設定します。
4.13	R_JCU_SetHuffmanTable	ハフマンテーブルを設定します。
4.14	R_JCU_GetEncodedSize	圧縮したデータのサイズを取得します。
4.15	R_JCU_OnInterrupting	割込みサービス ルーチン(ISR)

表 4-2 ユーザ定義関数一覧

章	関数名	概要
4.16	R_JCU_OnInitialize	ユーザ定義部の初期化をします。
4.17	R_JCU_OnFinalize	ユーザ定義部の終了処理をします。
4.18	R_JCU_EnableInterrupt	割込み許可の設定を行います。
4.19	R_JCU_DisableInterrupt	割込み禁止の設定を行います。

4.1 R_JCU_Initialize

概要 JCU ドライバを初期化します。

ヘッダ r_jcu.h

宣言 jcu_errorcode_t R_JCU_Initialize (jcu_config_t* in_out_Config);

説明 ドライバ内部の状態(gs_jcu_internal_information)を初期化します。
ユーザ定義関数(R_JCU_OnInitialize)を実行します。
ユーザ定義関数では、以下の処理を行ってください。

- JCU モジュールへのクロック供給
- JCU に関する割込みの優先度の設定
- その他処理に必要な環境固有の設定

引 数	jcu_config_t* in_out_Config	NULL
リターン値	エラーコード。	

4.2 R_JCU_TerminateAsync

概要 JCU ドライバを終了します。

ヘッダ r_jcu.h

宣言 jcu_errorcode_t R_JCU_TerminateAsync(r_co_function_t in_OnFinalized, volatile void* in_OnFinalizedArgument);

説明 JCU ドライバを終了する処理です。終了処理が完了する前に、すぐにリターンする非同期関数です。
ドライバ内部の状態を変更します。
ユーザ定義関数(R_JCU_OnFinalize)を実行します。
ユーザ定義関数では、以下の処理を行ってください。

- JCU モジュールへのクロック供給停止
- JCU に関する割込みの優先度の設定クリア
- その他処理に必要な環境固有の設定

非同期処理が完了したときは、0
R_JCU_GetAsyncError を呼び出してください。

引 数	r_co_function_t in_OnFinalized	終了処理が完了したときに呼び出すコールバック関数。 割込み、またはスレッドから呼ばれる。エラー発生時は呼ばれない
	volatile void* in_OnFinalizedArgument	終了処理が完了したときに呼び出すコールバック関数に渡す値
リターン値	エラーコード。	

4.3 R_JCU_SelectCodec

概要 圧縮または伸長を選択します。

ヘッダ r_jcu.h

宣言 jcu_errorcode_t R_JCU_SelectCodec(const jcu_codec_t codec);

説明 JCU の処理状態を選択(伸長または圧縮)します。
この関数を呼び出すと、デコードとエンコードに使うパラメータと、カウントモードの設定が初期化されるので、全て再設定してください。

引 数	jcu_codec_t codec	コーデック種別。
リターン値	エラーコード。	

4.4 R_JCU_SetPauseForImageInfo

概 要	画像情報取得可能になったら、一時停止するかどうかを選択します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_SetPauseForImageInfo(const bool_t is_pause);	
説 明	<ul style="list-style-type: none"> R_JCU_GetImageInfo 関数で、JPEG ファイルの画像情報が取得できる状態になったら、一時停止するかどうかを、引数で指定します。 	

引 数	bool_t is_pause	TRUE:一時停止する FALSE:一時停止しない
リターン値	エラーコード。	

4.5 R_JCU_SetErrorFilter

概 要	有効とするデコードエラー検出の種類(jcu_int_detail_error_t)を設定する。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_SetErrorFilter(jcu_int_detail_error_t filter);	
説 明	有効とするデコードエラー詳細情報を設定します。 <ul style="list-style-type: none"> 有効としたデコードエラーが発生した場合は、割込みが発生します。 	

引 数	jcu_int_detail_error_t filter	有効とするエラー検出の種類(jcu_int_detail_error_t)のビット・フラグ値。
リターン値	エラーコード。	

4.6 R_JCU_StartAsync

概 要	JCU の動作を開始します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_StartAsync(r_co_function_t in_OnFinished, volatile void* in_OnFinishedArgument);	
説 明	<ul style="list-style-type: none"> JCU をスタートします。デコードまたはエンコード処理が、完了または中断する前に、すぐにリターンする非同期関数です。 スタートする前に、R_JCU_SetDecoderParam または R_JCU_SetEncoderParam の API 関数で、パラメータを設定して下さい。 JCU をスタートした後は、キャンセルできません。 非同期処理が完了したときは、0 R_JCU_GetAsyncError を呼び出してください。 	

引 数	r_co_function_t in_OnFinalized	JCU の動作が完了したときに呼び出すコールバック関数。割込み、またはスレッドから呼ばれる。エラー発生時は呼ばれない
	volatile void* in_OnFinalizedArgument	JCU の動作が完了したときに呼び出すコールバック関数に渡す値
リターン値	エラーコード。	

4.7 R_JCU_GetAsyncError

概 要	非同期処理の中で発生したエラーを返します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_GetAsyncError(void);	
説 明	<ul style="list-style-type: none"> 	

引 数	なし	
リターン値	エラーコード。	

4.8 R_JCU_ContinueAsync

概 要	一時停止した JCU の動作を再開します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_ContinueAsync(const jcu_continue_type_t type, r_co_function_t in_OnFinished, volatile void* in_OnFinishedArgument);	
説 明	<ul style="list-style-type: none"> 中断した JCU の動作（モード）を再開させます。再開した動作が、完了または中断する前に、すぐにリターンする非同期関数です。 非同期処理が完了したときは、0 R_JCU_GetAsyncError を呼び出してください。 	
引 数	jcu_continue_type_t type	JCU が再開するモード。
	r_co_function_t in_OnFinalized	JCU の動作が完了したときに呼び出すコールバック関数。割込み、またはスレッドから呼ばれる。エラー発生時は呼ばれない
	volatile void* in_OnFinalizedArgument	JCU の動作が完了したときに呼び出すコールバック関数に渡す値
リターン値	エラーコード。	

4.9 R_JCU_SetDecodeParam

概 要	デコード用のパラメータを設定します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_SetDecodeParam(const jcu_decode_param_t *const decode, const jcu_buffer_param_t *const buffer);	
説 明	<ul style="list-style-type: none"> デコード用のパラメータを設定します。 ピクセルフォーマットが ARGB8888 以外のときは、decode.alpha には 0 を設定してください。 ピクセルフォーマットが YCbCr 以外のときは、decode.outputCbCrOffset には JCU_CBCR_OFFSET_0 を設定してください。 	
引 数	jcu_decode_param_t *decode	デコード用のパラメータ。
	jcu_buffer_param_t *buffer	入出力バッファ。
リターン値	エラーコード。	

4.10 R_JCU_GetImageInfo

概 要	JPEG ファイルの画像情報を取得します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_GetImageInfo(jcu_image_info_t *const buffer);	
説 明	<ul style="list-style-type: none"> JPEG ファイルの画像情報（幅、高さ、エンコードフォーマット）を取得します。 画像情報取得の割込み発生前は、不定値となります。 取得した JPEG ファイル画像情報のピクセルフォーマットが、jcu_jpeg_format_t の範囲外であった場合は、エラーと判定して、デコードは行わないでください。 	
引 数	jcu_image_info_t * buffer	画像情報格納用メモリのポインタ。
リターン値	エラーコード。	

4.11 R_JCU_SetEncodeParam

概 要	エンコード用のパラメータを設定します。	
ヘッダ	r_jcu.h	

宣 言	jcu_errorcode_t R_JCU_SetEncodeParam(const jcu_encode_param_t *const encode, const jcu_buffer_param_t *const buffer);	
説 明	<ul style="list-style-type: none"> エンコード用のパラメータを設定します。 	
引 数	jcu_encode_param_t *encode	エンコード用のパラメータ。
	jcu_buffer_param_t *buffer	入出力バッファ。
リターン値	エラーコード。	

4.12 R_JCU_SetQuantizationTable

概 要	量子化テーブルを設定します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_SetQuantizationTable(const jcu_table_no_t tableNo, const uint8_t *const table);	
説 明	<ul style="list-style-type: none"> 量子化テーブルの設定を行います。 量子化テーブルの設定値は、「RZ/A2M グループ ユーザーズマニュアル ハードウェア編」の、45.3.1 (4)章を参照してください。 付属の QuantizationTable_Generator.html ファイルで量子化テーブルの設定値のサンプルを計算することができます。 	
引 数	jcu_table_no_t tableNo	データをセットするテーブル番号。
	uint8_t * table	設定する量子化テーブル。
リターン値	エラーコード。	

4.13 R_JCU_SetHuffmanTable

概 要	ハフマンテーブルを設定します。	
ヘッダ	r_jcu.h	
宣 言	jcu_errorcode_t R_JCU_SetHuffmanTable(const jcu_table_no_t tableNo, const jcu_huff_t type, const uint8_t *const table);	
説 明	<ul style="list-style-type: none"> ハフマンテーブルの設定を行います。 ハフマンテーブルの設定値は、「RZ/A2M グループ ユーザーズマニュアル ハードウェア編」の、45.3.1 (4)章を参照してください。 	
引 数	jcu_table_no_t tableNo	データをセットするハフマンテーブル番号。
	jcu_huff_t type	ハフマンテーブルに設定する成分
	uint8_t * table	設定するハフマンテーブル
リターン値	エラーコード。	

4.14 R_JCU_GetEncodedSize

概 要	エンコードしたデータのサイズを取得します。
ヘッダ	r_jcu.h
宣 言	jcu_errorcode_t R_JCU_GetEncodedSize(size_t *const out_Size);
説 明	<ul style="list-style-type: none"> エンコードして圧縮したデータのサイズを取得します。 エンコード完了の割込み発生前は、不定値となります。

引 数	size_t * out_Size	サイズ情報格納用メモリのポインタ。
リターン値	エラーコード。	

4.15 R_JCU_OnInterrupting

概 要	割込みサービス ルーチン(ISR)
ヘッダ	r_jcu.h
宣 言	errnum_t R_JCU_OnInterrupting(void);
説 明	<ul style="list-style-type: none"> R_JCU_OnInitialize 関数で登録したすべての JCU の割込みコールバック関数から呼び出してください。

引 数	なし
リターン値	エラーコード。

4.16 R_JCU_OnInitialize

概 要	ユーザ定義部の初期化をします。
ヘッダ	r_jcu_pl.h
宣 言	errnum_t R_JCU_OnInitialize (void);
説 明	<ul style="list-style-type: none"> JCU ドライバの初期化処理からコールバックされるユーザ定義関数です。 必要ならば、以下の処理を行ってください。 <ul style="list-style-type: none"> JCU モジュールへのクロック供給 JCU に関係する割込みの優先度の設定 その他処理に必要な環境固有の設定

引 数	なし
リターン値	エラーコード。

4.17 R_JCU_OnFinalize

概 要	ユーザ定義部の終了処理をします。
ヘッダ	r_jcu_pl.h
宣 言	errnum_t R_JCU_OnFinalize (void);
説 明	<ul style="list-style-type: none"> JCU ドライバの終了処理からコールバックされるユーザ定義関数です。 必要ならば、以下の処理を行ってください。 <ul style="list-style-type: none"> JCU モジュールへのクロック供給の停止 JCU に関係する割込みの優先度のクリア その他処理に必要な環境固有の設定

引 数	なし
リターン値	エラーコード。

4.18 R_JCU_EnableInterrupt

概 要	JCU のすべての割込みを許可します。	
ヘッダ	r_jcu_pl.h	
宣 言	void R_JCU_EnableInterrupt(void);	
説 明	• JCU ドライバからコールバックされるユーザ定義関数です。	
引 数	なし	
リターン値	なし	

4.19 R_JCU_DisableInterrupt

概 要	JCU のすべての割込みを禁止します。	
ヘッダ	r_jcu_pl.h	
宣 言	bool_t R_JCU_DisableInterrupt(void);	
説 明	• JCU ドライバからコールバックされるユーザ定義関数です。	
引 数	なし	
リターン値	今まで JCU のすべての割込みを許可していたかどうか	

5. ドライバのインポート方法

5.1 e² studio

Smart Configurator ツールを使用して e2 studio のプロジェクトにドライバをインポートする方法の詳細については、RZ/A2M Smart Configurator ユーザーガイド : e2 studio R20AN0583JJ を参照してください。

5.2 e² studio 以外で作成されたプロジェクトの場合

このセクションでは、ドライバをプロジェクトにインポートする方法について説明します。

一般的に、どの IDE にも 2 つのステップがあります。

- 1) プロジェクトに必要なソースツリー内の場所にドライバをコピーします。
- 2) ドライバをコピーした場所へのリンクをコンパイラに追加します。

他に必要なドライバがある場合（例えば r_cbuffer など）、同様にインポートする必要があります。

6. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RZ/A2M グループ ユーザーズマニュアル ハードウェア編

(最新版をルネサス エレクトロニクスホームページから入手してください。)

RTK7921053C00000BE (RZ/A2M CPU ボード) ユーザーズマニュアル

(最新版をルネサス エレクトロニクスホームページから入手してください。)

RTK79210XXB00000BE (RZ/A2M SUB ボード) ユーザーズマニュアル

(最新版をルネサス エレクトロニクスホームページから入手してください。)

Arm Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

(最新版を Arm ホームページから入手してください。)

Arm Cortex™-A9 Technical Reference Manual Revision: r4p1

(最新版を Arm ホームページから入手してください。)

Arm Generic Interrupt Controller Architecture Specification - Architecture version2.0

(最新版を Arm ホームページから入手してください。)

Arm CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual Revision: r3p3

(最新版を Arm ホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ユーザーズマニュアル：統合開発

統合開発環境 e2 studio のユーザーズマニュアルは、ルネサス エレクトロニクスホームページから入手してください。

(最新版をルネサス エレクトロニクスホームページから入手してください。)

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.10	2019. 5. 17	p4	表 6.1 動作確認条件 コンパイラオプション"-mthumb-interwork"を削除
		p15	「5.ドライバのインポート方法」追加
1.03	2019. 4. 15	p4	動作確認を行った統合開発環境のバージョンを更新
1.02	2018.12.28	—	修正：スタンバイ関連を修正。R_JCU_OnInitialize 関数と R_JCU_OnFinalize 関数の内部で STBACK レジスタをチェッ クするように修正。
1.00	2018.9.14	—	新規発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレイやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力ブルアップ電源を入れないでください。入力信号や入出力ブルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違っていると、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

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まれます。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、データシート等において高信頼性、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)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。