RZ/A2M グループ
MIPI ドライバ

R01AN4481JJ0101
Rev.1.01
2018.12.28

要旨

本アプリケーションノートでは、RZ/A2M の MIPI とビデオインプットモジュール(VIN)ドライバの機能仕様について説明します。

動作確認デバイス

RZ/A2M グループ

目次

1. 仕様

本ドライバは RZ/A2M グループ内蔵の MIPI CSI2 インタフェース(以下、MIPI とする)、およびビデオインプットモジュール(以下、VIN とする)を利用し、外部から入力される画像データを取り込み、メモリへ転送するドライバです。

MIPI ドライバで使用する周辺機能と用途を表 1-1 に示します。

表 1-1 MIPI ドライバで使用する周辺機能と用途

分類	項目	実現機能	説明	特記事項
カメラ	データ通信	データレーン切替	2 レーン並列動作、または 1 レーン動作	
		転送レート	80MHz～1GHz	
	データ補正	パケットヘッダ	ECC1 ビットエラー訂正 2 ビット以上のエラー検出	
		ペイロードデータ	CRC エラー検出	
	入力フォーマット	RAW 8bit	Bayer またはグレイスケール	
	キャプチャ画素数	任意	最大 2048 x 2048pixel	
	サイズクリッピング	任意	最大 2048 x 2048pixel	
	キャプチャモード	シングル/連続 切替	シングルフレーム、または 連続フレーム	
		フィールド	奇数フィールドキャプチャモード 偶数/奇数フィールドキャプチャモード 偶数フィールドキャプチャモード	
メモリ書き込み	出力フォーマット	RAW 8bit	Bayer またはグレイスケール	

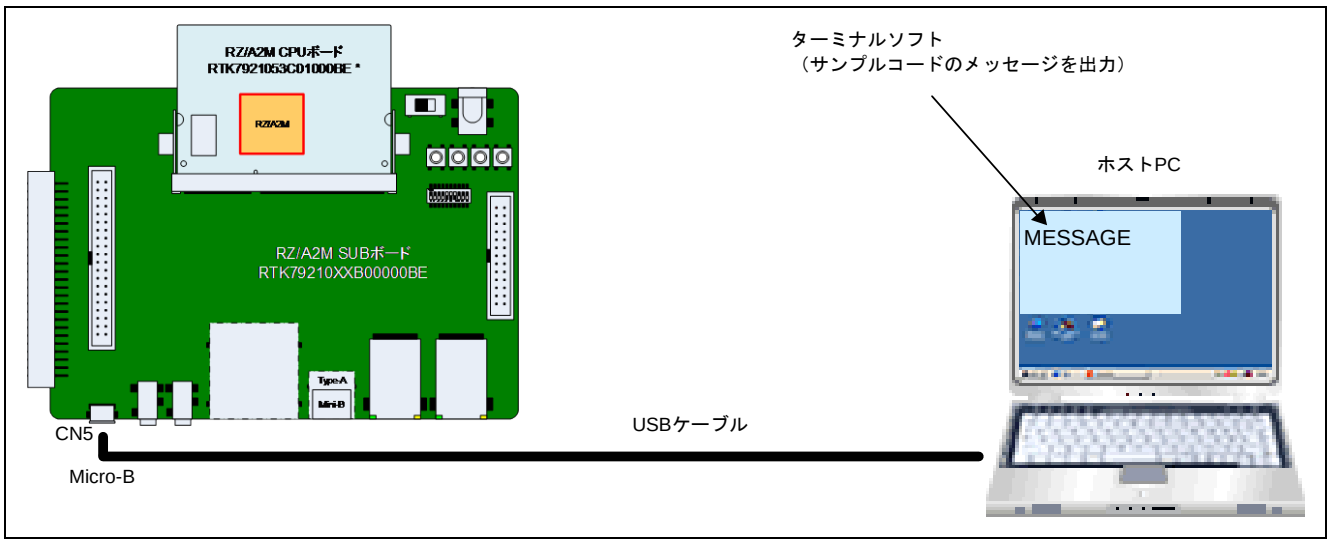

図 1.1 動作環境

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
使用マイコン	RZ/A2M
動作周波数（注）	CPU クロック（Iφ）：528MHz 画像処理クロック（Gφ）：264MHz 内部バスクロック（Bφ）：132MHz 周辺クロック 1（P1φ）：66MHz 周辺クロック 0（P0φ）：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 V7.3.0
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 -mthumb-interwork -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 -mthumb-interwork -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	• 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）	• 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. ハードウェア説明

3.1 使用端子

使用端子と機能を表 3-1 に示します。

表 3-1 使用端子一覧

端子名	入出力	内容	ターゲットボード接続
CSI_DATA0P	入力	CSI2 データレーン 0(差動ポジティブ)	専用端子
CSI_DATA0N	入力	CSI2 データレーン 0(差動ネガティブ)	専用端子
CSI_DATA1P	入力	CSI2 データレーン 1(差動ポジティブ)	専用端子
CSI_DATA1N	入力	CSI2 データレーン 1(差動ネガティブ)	専用端子
CSI_CLKP	入力	CSI2 クロックレーン(差動ポジティブ)	専用端子
CSI_CLKN	入力	CSI2 クロックレーン(差動ネガティブ)	専用端子

4. ソフトウェア説明

4.1 ファイル構成

本ドライバを構成するファイルを表 4-1 に示します。

表 4-1 構成ファイル一覧

ファイル名	説明
r_mipi_api.c	MIPI ドライバ API 関数 MIPI ドライバの API 関数を記述したソースファイル
r_mipi_api.h	MIPI ドライバ API 定義 MIPI ドライバの API 関数のプロトタイプや、API として定義したパラメータについて記述したヘッダファイル
r_mipi_userdef_api.c	MIPI ドライバユーザ定義関数 割り込み設定等の MIPI ドライバを動作させるうえで、ユーザ環境に依存する部分を記述するソースファイル
r_mipi_user.h	MIPI ドライバユーザ定義 ユーザ定義関数のプロトタイプ宣言や、ユーザが静的な定数を定義するヘッダファイル

4.2 状態遷移

本ドライバの状態遷移を図 4-1 に示します。

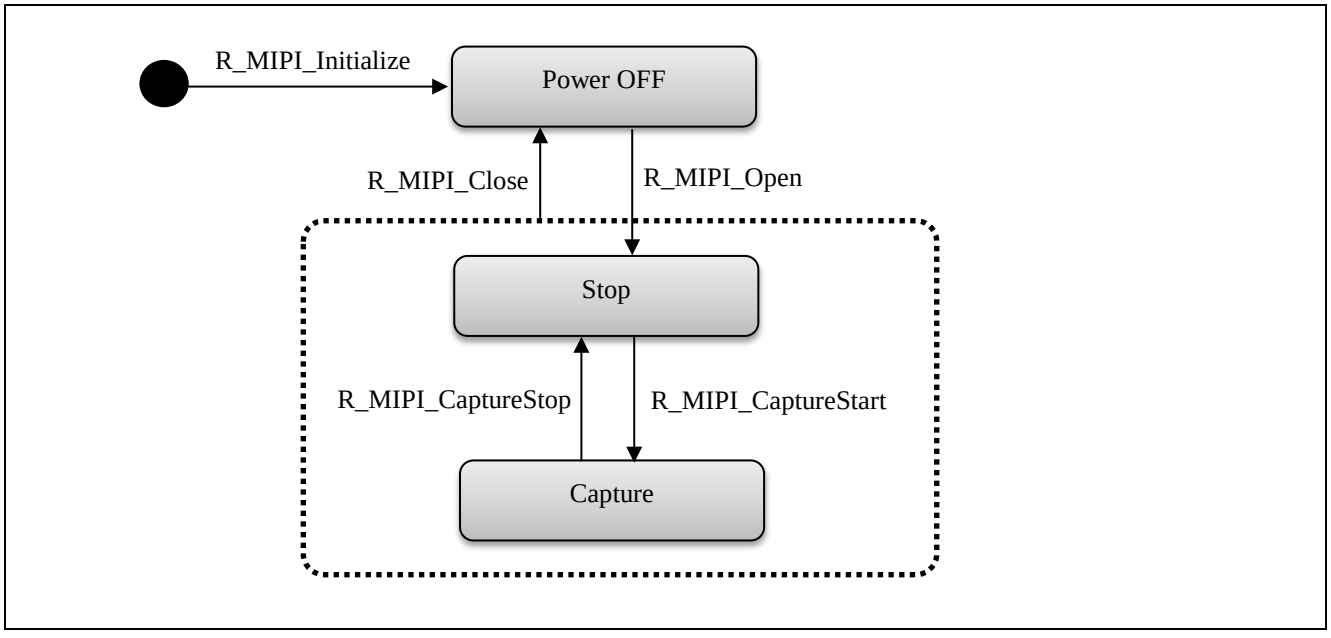

図 4-1 状態遷移図

4.3 列挙型定義

本ドライバが定義する列挙型について以下に記載します。エラーコードについては「4.4 エラーコード」を参照ください。

(1) e_mipi_capture_mode_t

e_mipi_capture_mode_t はフレームキャプチャモードを表す列挙型です。

```
typedef enum
{
    MIPI_SINGLE_MODE = 0,
    MIPI_CONTINUOUS_MODE,
} e_mipi_capture_mode_t;
```

列挙定数	説明
MIPI_SINGLE_MODE	シングルキャプチャモード
MIPI_CONTINUOUS_MODE	連続キャプチャモード

(2) e_mipi_inter_t

e_mipi_inter_t はフィールド検出制御を表す列挙型です。

```
typedef enum
{
    MIPI_PROGRESSIVE = 0,
    MIPI_INTERLACE,
} e_mipi_inter_t;
```

列挙定数	説明
MIPI_PROGRESSIVE	プログレッシブデータ
MIPI_INTERLACE	インターレースデータ

(3) e_vin_inputformat_t

e_vin_inputformat_t は入力フォーマットを表す列挙型です。

```
typedef enum
{
    VIN_INPUT_YCBCR422_8 = 0,
    VIN_INPUT_YCBCR422_8I,
    VIN_INPUT_RAW8,
} e_vin_inputformat_t;
```

列挙定数	説明
VIN_INPUT_YCBCR422_8	YUV(=YCbCr422 8bit)
VIN_INPUT_YCBCR422_8I	UYVY
VIN_INPUT_RAW8	RAW 8bit

(4) e_vin_outputformat_t

e_vin_outputformat_t は出力フォーマットを表す列挙型です。

```
typedef enum
{
    VIN_OUTPUT_YCBCR422_8   = (0x00),
    VIN_OUTPUT_Y8_CbCr8     = (0x02),
    VIN_OUTPUT_Y8           = (0x03),
    VIN_OUTPUT_RAW8         = (0x20),
} e_vin_outputformat_t;
```

列挙定数	説明
VIN_OUTPUT_YCBCR422_8	YUV (=YCbCr422 8bit)
VIN_OUTPUT_Y8_CbCr8	YC 分離, YCbCr422(Y 8bit, Cb/Cr 8bit 多重化)
VIN_OUTPUT_Y8	YC 分離, Y data(8bit)
VIN_OUTPUT_RAW8	RAW 8bit

(5) e_vin_outputendian_t

e_vin_outputendian_t は出力データのエンディアンタイプを表す列挙型です。

```
typedef enum
{
    VIN_OUUPUT_EN_LITTLE = 0,
    VIN_OUTPUT_EN_BIG,
} e_vin_outputendian_t;
```

列挙定数	説明
VIN_OUUPUT_EN_LITTLE	リトルエンディアン
VIN_OUTPUT_EN_BIG	ビッグエンディアン

(6) e_vin_interlace_t

e_vin_interlace_t はインタレースモードを表す列挙型です。

```
typedef enum
{
    VIN_INTERLACE_ODD = 0,
    VIN_INTERLACE_EVEN,
    VIN_INTERLACE_BOTH,
    VIN_DINTERLACE,
    VIN_PROGRESSIVE,
} e_vin_interlace_t;
```

列挙定数	説明
VIN_INTERLACE_ODD	奇数フィールドキャプチャモード
VIN_INTERLACE_EVEN	偶数フィールドキャプチャモード
VIN_INTERLACE_BOTH	奇数/偶数フィールドキャプチャモード
VIN_PROGRESSIVE	プログレッシブ

(7) e_vin_input_align_t

e_vin_input_align_t は入力フォーマットが YCbCr422 形式の際の、データアラインメントを表す列挙型です。

```
typedef enum
{
    VIN_Y_UPPER = 0,
    VIN_CB_UPPER,
} e_vin_input_align_t;
```

列挙定数	説明
VIN_Y_UPPER	上位ビットに Y、下位ビットに CbCr を格納
VIN_CB_UPPER	上位ビットに CbCr、下位ビットに Y を格納

(8) e_vin_output_swap_t

e_vin_output_swap_t は出力データのバイトスワップを表す列挙型です。例えば、YUYV 形式のデータを、UYVY 形式で転送する場合、VIN_SWAP_ON を設定します。

```
typedef enum
{
    VIN_SWAP_OFF = 0,
    VIN_SWAP_ON,
} e_vin_output_swap_t;
```

列挙定数	説明
VIN_SWAP_OFF	バイトスワップあり
VIN_SWAP_ON	バイトスワップなし

(9) e_mipi_interrupt_type_t

e_mipi_interrupt_type_t は MIPI および VIN の割り込み要因を表す列挙型です。

```
typedef enum
{
    MIPI_INT_LESS_THAN_WC   = 0x00000001,
    MIPI_INT_AFIFO_OF       = 0x00000002,
    MIPI_INT_VD_START       = 0x00000004,
    MIPI_INT_VD_END         = 0x00000008,
    MIPI_INT_SHP_STB        = 0x00000010,
    MIPI_INT_FSFE           = 0x00000020,
    MIPI_INT_LNP_STB        = 0x00000040,
    MIPI_INT_CRC_ERR        = 0x00000080,
    MIPI_INT_HD_WC_ZERO     = 0x00000100,
    MIPI_INT_FRM_SEQ_ERR1   = 0x00000200,
    MIPI_INT_FRM_SEQ_ERR0   = 0x00000400,
    MIPI_INT_ECC_ERR        = 0x00000800,
    MIPI_INT_ECC_CRCT_ERR   = 0x00001000,
    MIPI_INT_ULPS_START     = 0x00002000,
    MIPI_INT_ULPS_END       = 0x00004000,
    MIPI_INT_ERRSOTHS       = 0x00008000,
    MIPI_INT_ERRSOTSYNCHS   = 0x00010000,
    MIPI_INT_ERRESC         = 0x00020000,
    MIPI_INT_ERRCONTROL     = 0x00040000,
    VIN_INT_FIELD2          = 0x00100000,
    VIN_INT_VSYNC_FALL      = 0x00200000,
    VIN_INT_VSYNC_RISE      = 0x00400000,
    VIN_INT_FIELD           = 0x00800000,
    VIN_INT_SCANLINE        = 0x01000000,
    VIN_INT_FRAME           = 0x02000000,
    VIN_INT_FIFO_OF         = 0x04000000
} e_mipi_interrupt_type_t;
```

列挙定数	説明
MIPI_INT_LESS_THAN_WC	ロングパケットのペイロードデータ長が、WC 値よりも小さいときのエラー割り込み
MIPI_INT_AFIFO_OF	PHY からの HS データが格納される非同期 FIFO のオーバフロー割り込み
MIPI_INT_VD_START	VD 信号出力の開始割り込み
MIPI_INT_VD_END	VD 信号出力の終了割り込み
MIPI_INT_SHP_STB	ショートパケット受信割り込み
MIPI_INT_FSFE	フレームパケット受信割り込み
MIPI_INT_LNP_STB	ロングパケット受信割り込み
MIPI_INT_CRC_ERR	CRC エラー割り込み
MIPI_INT_HD_WC_ZERO	WC ゼロ割り込み
MIPI_INT_FRM_SEQ_ERR1	フレームシーケンスエラー1 割り込み (不正なフレームエンドパケット受信)
MIPI_INT_FRM_SEQ_ERR0	フレームシーケンスエラー0 割り込み (不正なフレームスタートパケット受信)
MIPI_INT_ECC_ERR	ECC エラー割り込み
MIPI_INT_ECC_CRCT_ERR	ECC 1 ビット訂正割り込み
MIPI_INT_ULPS_START	ウルトラローパワー転送開始割り込み
MIPI_INT_ULPS_END	ウルトラローパワー転送終了割り込み
MIPI_INT_ERRSOTHS	同期化 SOT(転送開始)エラー割り込み
MIPI_INT_ERRSOTSYNCHS	非同期 SOT(転送開始)エラー割り込み
MIPI_INT_ERRESC	エスケープモードエントリエラー割り込み
MIPI_INT_ERRCONTROL	PHY 制御エラー割り込み
VIN_INT_FIELD2	フィールド割り込み
VIN_INT_VSYNC_FALL	VSYNC 立ち下りエッジ検出割り込み
VIN_INT_VSYNC_RISE	VSYNC 立ち上がりエッジ検出割り込み
VIN_INT_FIELD	フィールドスイッチング割り込み
VIN_INT_SCANLINE	スキャンライン割り込み
VIN_INT_FRAME	フレーム終了割り込み
VIN_INT_FIFO_OF	FIFO オーバフロー割り込み

4.4 エラーコード

本ドライバのエラーコード一覧を表 4-2 に示します。

表 4-2 MIPI ドライバのエラーコード一覧

エラーコード	値	説明
MIPI_OK	0	正常終了 呼び出された API 関数は正常に終了しました
MIPI_STATUS_ERR	1	ステータスエラー 許可されていない条件において、API 関数が呼び出されました
MIPI_PARAM_ERR	2	パラメータエラー API 関数が要求する引数において、許可されていないパラメータが指定されました。

4.5 ユーザカスタムパラメータ

本ドライバでは”r_mipi_user.h”において、ユーザが静的に変更可能なパラメータを定義します。

(1) 定数定義

以下に定数を記載します。MIPI D-PHY の各信号タイミングの詳細は MIPI CSI-2 の規格書を参照してください。

定数	値	説明
MIPI_INTERRUPT_PRIORITY	28u	MIPI の割り込み優先度
VIN_INTERRUPT_PRIORITY	28u	VIN の割り込み優先度
MIPI_1US_WAIT	528u	1us 待ち 本設定は CPU クロック=528MHz の場合の値

4.6 制限事項

(1) 再入可能性

本ドライバの関数は再入可能ではありません。本ドライバの関数を複数タスクや割り込み処理から非同期に呼び出した場合、予期しない動作をする可能性があります。

4.7 関数一覧

本ドライバの API 関数一覧を表 4-3 に示します。

表 4-3 API 一覧

関数名	概要	定義ヘッダ
R_MIPI_Initialize	初期化処理	r_mipi_api.h
R_MIPI_Open	MIPI コンフィグレーション、PHY の起動	r_mipi_api.h
R_MIPI_Close	MIPI および VIN の終了処理	r_mipi_api.h
R_MIPI_Setup	VIN コンフィグレーション	r_mipi_api.h
R_MIPI_SetBufferAdr	取り込みバッファ設定	r_mipi_api.h
R_MIPI_InterruptEnable	割り込み許可設定	r_mipi_api.h
R_MIPI_InterruptDisable	割り込み禁止設定	r_mipi_api.h
R_MIPI_GetInfo	キャプチャ情報取得	r_mipi_api.h
R_MIPI_CaptureStart	キャプチャ開始処理	r_mipi_api.h
R_MIPI_CaptureStop	キャプチャ停止処理	r_mipi_api.h
R_MIPI_InterruptHandler	MIPI 割り込みハンドラ	r_mipi_api.h
R_VIN_InterruptHandler	VIN 割り込みハンドラ	r_mipi_api.h
R_MIPI_CPUVAddrToSysPAddr	取り込みバッファアドレス変換処理	r_mipi_user.h
R_MIPI_OnInitialize	MIPI および VIN のスタンバイ解除、割り込みハンドラの登録サンプル	r_mipi_user.h
R_MIPI_OnFinalize	MIPI および VIN のスタンバイ設定、割り込みハンドラの解除サンプル	r_mipi_user.h

5. 関数リファレンス

5.1 R_MIPI_Initialize

R_MIPI_Initialize			
概　要	初期化処理		
ヘッダ	r_mipi_api.h		
宣　言	`void R_MIPI_Initialize(void (* const init_func)(uint32_t), const uint32_t user_num);`		
引　数	[IN]	void (* init_func)(uint32_t)	: コールバック関数の登録 必要がない場合、NULL を設定してください
	[IN]	uint32_t user_num	: コールバックの引数 用途に合わせて設定してください
リターン値	なし		
備考	なし		

(1) 説明

本関数は MIPI ドライバを初期化します。MIPI ドライバではモジュールスタンバイ解除や割り込みハンドラの登録処理を行っていないため、本関数のコールバック関数内でモジュールスタンバイ解除、割り込みハンドラの登録処理を行ってください。例として、「5.14 R_MIPI_OnInitialize」をユーザ定義関数として準備していますので、R_MIPI_OnInitialize を参考にユーザのシステムに合った設定を実装してください。

本関数では以下の処理を行います。

- 引数で登録されたコールバック関数の呼び出し
- ドライバ内で使用する内部変数の初期化

5.2 R_MIPI_Open

R_MIPI_Open			
概　要	MIPI コンフィグレーション、PHY の起動		
ヘッダ	r_mipi_api.h		
宣　言	`e_mipi_error_t R_MIPI_Open(const st_mipi_param_t * const mipi_data);`		
引　数	[IN]	const st_mipi_param_t * const mipi_data	: コンフィグレーションデータ NULL は設定しないでください
リターン値	MIPI_OK		: 正常終了
	MIPI_STATUS_ERR		: ドライバ内部ステータスが不正
	MIPI_PARAM_ERR		: mipi_data の設定が不正はたは範囲外
備考	R_MIPI_Initialize 関数実行後に呼び出し可能です		

(1) 説明

MIPI のキャプチャレーンや画像取り込み方式、PHY の設定を行います。

本関数では以下の処理を行います。

- コンフィグレーションデータのパラメータチェック
- MIPI のソフトウェアリセット
- インタレース / プログレッシブ取り込みの設定
- 仮想チャネル設定
- PHY の初期化
- ドライバ内ステータスのアップデート

(2) パラメータ詳細

(a) st_mipi_param_t

st_mipi_param_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint8_t  mipi_lanenum;
    uint8_t  mipi_vc;
    uint8_t  mipi_interlace;
    uint8_t  mipi_laneswap;
    uint16_t mipi_frametop;
    uint16_t mipi_outputrate;
    st_mipi_phy_timing_t mipi_phy_timing;
} st_mipi_param_t;
```

型 メンバ	説明
uint8_t mipi_lanenum	転送レーン数 1：1 レーン動作 2：2 レーン並列動作
uint8_t mipi_vc	仮想チャネル 0～3 有効な仮想チャネル番号
uint8_t mipi_interlace	入力方式 (T.B.D：現 Ver.では MIPI_PROGRESSIVE 固定) (注) MIPI_PROGRESSIVE：プログレッシブ MIPI_INTERLACE：インタレース
uint8_t mipi_laneswap	レーンスワップ (T.B.D：現 Ver.では 0 固定) (注) 0：レーンスワップなし 1：レーンスワップあり
uint16_t mipi_frametop	偶数フィールド番号 0x0000～0xFFFF インタレース入力画像のトップフィールドを検出するための値 先頭ライン同期パケットの ID を設定
uint16_t mipi_outputrate	MIPI 転送レート(MHz)　(T.B.D：現 Ver.では 80 固定) (注) 80～1000 MIPI の転送レートを設定します
st_mipi_phy_timing_t mipi_phy_timing	PHY タイミング設定 PHY のデータレーンとクロックレーンのタイミングを設定します 詳細は「st_mipi_phy_timing_t」を参照ください

【注】　現在のドライバ Ver.では、本パラメータは非対応です。各パラメータにおいては、説明欄に記載の固定値を使用してください。

偶数フィールド番号(mipi_frametop)は入力方式(mipi_interlace)が MIPI_INTERLACE の場合のみ有効です。

仮想チャネル(mipi_vc)とは、カメラからデータが流れるチャネルを意味します。

(b) st_mipi_phy_timing_t

st_mipi_phy_timing_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t mipi_ths_prepare;
    uint16_t mipi_ths_settle;
    uint16_t mipi_tclk_prepare;
    uint16_t mipi_tclk_settle;
    uint16_t mipi_tclk_miss;
    uint16_t mipi_t_init_slave;
} st_mipi_phy_timing_t;
```

型 メンバ	説明
uint16_t mipi_ths_prepare	MIPI D-PHY $T_{THS\_PREPARE}$ パラメータ 0x00～0x3F データレーンの LP-00 継続時間(HS-0 の直前まで)の設定
uint16_t mipi_ths_settle	MIPI D-PHY T_{THS_SETTLE} パラメータ 0x00～0x3F データレーンの $T_{THS\_PREPARE}$ 開始後、HS 遷移を無視すべき時間の設定
uint16_t mipi_tclk_prepare	MIPI D-PHY $T_{CLK\_PREPARE}$ パラメータ 0x00～0x3F クロックレーンの LP-00 継続時間(HS-0 の直前まで)の設定
uint16_t mipi_tclk_settle	MIPI D-PHY T_{CLK_SETTLE} パラメータ 0x00～0x3F クロックレーンの $T_{CLK\_PREPARE}$ 開始後、HS 遷移を無視すべき時間の設定
uint16_t mipi_tclk_miss	MIPI D-PHY T_{CLK_MISS} パラメータ 0x00～0x1F Clock レーンの Clock 欠如を検出し、HS-RX を無効にするまでの時間
uint16_t mipi_t_init_slave	MIPI D-PHY T_{INIT} パラメータ 0x0000～0xFFFF INIT ステートの最低持続時間

5.3 R_MIPI_Close

R_MIPI_Close			
概　要	MIPI および VIN の終了処理		
ヘッダ	r_mipi_api.h		
宣　言	`e_mipi_error_t R_MIPI_Close(void (* const init_func)(uint32_t), const uint32_t user_num);`		
引　数	[IN]	void (* init_func)(uint32_t)	: コールバック関数の登録 必要がない場合、NULL を設定してください
	[IN]	uint32_t user_num	: コールバックの引数 用途に合わせて設定してください
リターン値	MIPI_OK		: 正常終了
	MIPI_STATUS_ERR		: ドライバ内部ステータスが不正
備考	R_MIPI_Open 関数実行後に呼び出し可能です		

(1) 説明

キャプチャ動作を停止させ、MIPI のソフトウェアリセットを行います。MIPI ドライバではモジュールスタンバイ設定や割り込みハンドラの解除処理を行っていないため、本関数のコールバック関数内でモジュールスタンバイ設定、割り込みハンドラの解除処理を行ってください。

例として、「5.15 R_MIPI_OnFinalize」をユーザ定義関数として準備していますので、R_MIPI_OnFinalize を参考にユーザのシステムに合った設定を実装してください。

本関数では以下の処理を行います。

- MIPI および VIN の割り込み有効レジスタを無効に設定
- VIN のキャプチャ停止
- PHY の初期化および MIPI のソフトウェアリセット
- ドライバ内で使用する内部変数の初期化
- 引数で登録されたコールバック関数の呼び出し

5.4 R_MIPI_Setup

R_MIPI_Setup			
概　要	VIN コンフィグレーション		
ヘッダ	r_mipi_api.h		
宣　言	`e_mipi_error_t R_MIPI_Setup(const st_vin_setup_t * const vin_setup );`		
引　数	[IN]	const st_vin_setup_t * const vin_setup	: コンフィグレーションデータ NULL は設定しないでください
リターン値	MIPI_OK		: 正常終了
	MIPI_STATUS_ERR		: ドライバ内部ステータスが不正
	MIPI_PARAM_ERR		: mipi_data の設定が不正はたは範囲外
備考	R_MIPI_Open 関数実行後に呼び出し可能です また、キャプチャ動作停止中に本関数で VIN の設定を行ってください		

(1) 説明

キャプチャ画像のクリッピングエリア、入出力フォーマット、ストライドサイズを設定します。

本関数では以下の処理を行います。

- コンフィグレーションデータのパラメータチェック
- VIN の各種レジスタ設定

(2) パラメータ詳細

(a) st_vin_setup_t

st_vin_setup_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    st_vin_preclip_t   vin_preclip;
    uint8_t        vin_inputformat;
    uint8_t        vin_outputformat;
    uint8_t        vin_outputendian;
    uint8_t        vin_interlace;
    uint16_t       vin_stride;
    uint32_t       vin_ycoffset;
   e_vin_input_align_t  vin_input_align;
   e_vin_output_swap_t  vin_output_swap;
} st_vin_setup_t;
```

型 メンバ	説明
st_vin_preclip_t vin_preclip	プリクリップエリア設定 キャプチャ画像に対するクリップエリアを設定します 詳細は「st_vin_preclip_t」を参照ください
uint8_t vin_inputformat	入力フォーマット VIN_INPUT_YCBCR422_8：YUY(=YCbCr422 8bit) VIN_INPUT_YCBCR422_8I：UYVY VIN_INPUT_RAW8：RAW 8bit
uint8_t vin_outputformat	出力フォーマット VIN_OUTPUT_YCBCR422_8：YUY(=YCbCr422 8bit) VIN_OUTPUT_Y8_CbCr8：YC 分離, YCbCr422(Y 8bit, Cb/Cr 8bit) VIN_OUTPUT_Y8：YC 分離, Y data(8bit) VIN_OUTPUT_RAW8：RAW 8bit
uint8_t vin_outputendian	エンディアンタイプ VIN_OUUPUT_EN_LITTLE：リトルエンディアン VIN_OUTPUT_EN_BIG：ビッグエンディアン
uint8_t vin_interlace	インタレースモード VIN_INTERLACE_ODD：奇数フィールドキャプチャモード VIN_INTERLACE_EVEN：偶数フィールドキャプチャモード VIN_INTERLACE_BOTH：奇数/偶数フィールドキャプチャモード VIN_PROGRESSIVE：プログレッシブキャプチャモード
uint16_t vin_stride	イメージストライド 出力画像のストライドサイズ(Pixel 単位)を設定します
uint32_t vin_ycoffset	UV データアドレスオフセット 0～128 の倍数 出力フォーマットで YC 分離出力を指定した際の、UV の転送オフセットアドレスを指定します
e_vin_input_align_t vin_input_align	YCbCr422 入力データアライメント VIN_Y_UPPER：上位ビット(Y) 下位ビット(CbCr) VIN_CB_UPPER：上位ビット(CbCr) 下位ビット(Y)
e_vin_output_swap_t vin_output_swap	出力データバイトスワップモード VIN_SWAP_OFF：スワップしない VIN_SWAP_ON：スワップする

エンディアンタイプ(vin_outputendian)は外部メモリに出力する際のエンディアンタイプを指定します。

イメージストライド(vin_stride)は、st_vin_preclip_t 構造体で指定した水平プリクリッピングサイズ(vin_preclip_endx - vin_preclip_startx)以上の値を設定する必要があります。したがって、以下の式を満たすようにイメージストライドを設定してください。

vin_stride >= vin_afterclip_size_x

また、イメージストライドは出力フォーマット(vin_outputformat)により、以下のようにパラメータを設定する必要があります。

出力フォーマット	設定単位(Pixel)
VIN_OUTPUT_YCBCR422_8	64
VIN_OUTPUT_Y8_CbCr8	128
VIN_OUTPUT_Y8	128
VIN_OUTPUT_RAW8	64

イメージストライドで設定した値は、MIPI ドライバが VnIS レジスタへ設定値を書き込みます。H/W 仕様により、出力フォーマットが VIN_OUTPUT_RAW8 の場合は vin_stride を 2 で割った値を MIPI ドライバが VnIS レジスタへ書き込みます。

(b) st_vin_preclip_t

st_vin_preclip_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t vin_preclip_starty;
    uint16_t vin_preclip_endy;
    uint16_t vin_preclip_startx;
    uint16_t vin_preclip_endx;
} st_vin_preclip_t;
```

型 メンバ	説明
uint16_t vin_preclip_starty	スタートライン(垂直方向) 0～2046 (スケーリング使用時は 0～2044) 値 0 は最初の有効な行を意味します
uint16_t vin_preclip_endy	エンドライン(垂直方向) 1～2047 (スケーリング使用時は 3～2047)
uint16_t vin_preclip_startx	スタートピクセル(水平方向) 0～2042 までの偶数 値 0 は最初の有効ピクセルが指定されます
uint16_t vin_preclip_endx	エンドピクセル(水平方向) 5～2047 までの奇数

垂直方向のライン数は、プリクリッピング後のライン数が 2 以上となる必要があるため、

(vin_preclip_endy - vin_preclip_starty) >= 1

となるように設定してください。また、垂直または水平スケーリングを使用する場合は、

(vin_preclip_endy - vin_preclip_starty) >=3

となるように設定してください。

水平方向のピクセル数は、プリクリッピング後のピクセル数が 6 よりも大きな偶数となる必要があるため、

(vin_preclip_endx - vin_preclip_startx) >=5

を満たし、かつ奇数となるように設定してください。

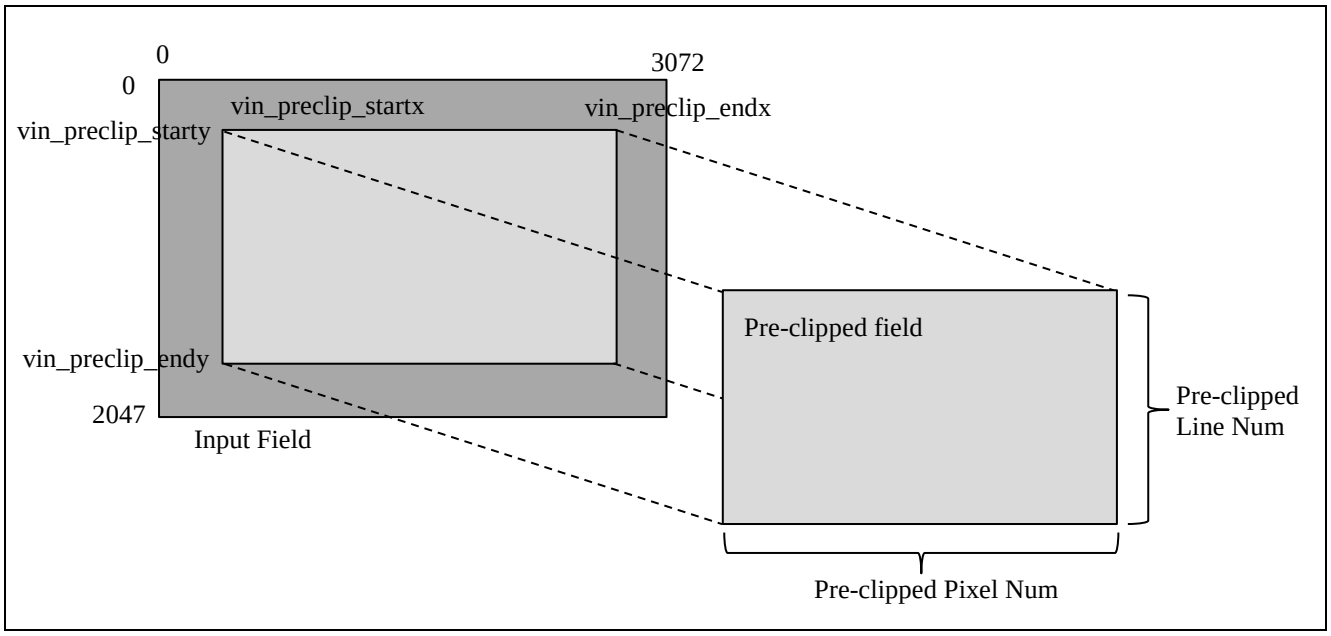

図 5-1 プリクリッピングエリアのイメージ

5.5 R_MIPI_SetBufferAdr

R_MIPI_SetBufferAdr			
概　要	取り込みバッファ設定		
ヘッダ	r_mipi_api.h		
宣　言	`e_mipi_error_t R_MIPI_SetBufferAdr(const uint8_t buffer_no, const uint8_t * const bufferBase);`		
引　数	[IN]	const uint8_t buffer_no	: VIN のメモリベース番号 0 : MB1 を指定 1 : MB2 を指定 2 : MB3 を指定
	[IN]	const uint8_t * const bufferBase	: データ格納先アドレスの指定 NULL は設定しないでください
リターン値	MIPI_OK		: 正常終了
	MIPI_STATUS_ERR		: ドライバ内部ステータスが不正
	MIPI_PARAM_ERR		: mipi_data の設定が不正はたは範囲外
備考	R_MIPI_Open 関数実行後に呼び出し可能です		

(1) 説明

指定されたデータ格納先アドレス(bufferBase)を、VIN のメモリベースレジスタへ割り当てます。VIN は 3 つのメモリベースレジスタ、MB1、MB2、MB3 を搭載しています。第 1 引数で指定されたメモリベースレジスタへ、第 2 引数で指定されたアドレスを設定します。

連続フレームキャプチャモードを使用する場合、キャプチャシーケンスは、MB1 -> MB2 -> MB3 の順にデータが格納されます。シングルキャプチャモードを使用する場合は、MB1 へ設定されたアドレスへキャプチャデータを格納します。

本関数では以下の処理を行います。

- ドライバ内ステータスチェック
- メモリベースレジスタへデータ格納先アドレスを設定

5.6 R_MIPI_InterruptEnable

R_MIPI_InterruptEnable			
概　要	割り込み許可設定		
ヘッダ	r_mipi_api.h		
宣　言	`void R_MIPI_InterruptEnable(const st_mipi_int_t * const param );`		
引　数	[IN]	const st_mipi_int_t * const param	: 割り込み設定 NULL は設定しないでください
リターン値	なし		
備考	なし		

(1) 説明

割り込み設定で指定された内容に従い、MIPI および VIN の割り込み許可設定を行います。

本関数では以下の処理を行います。

- MIPI および VIN の割り込み許可

(2) パラメータ詳細

(a) st_mipi_int_t

st_mipi_int_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    e_mipi_interrupt_type_t type;
    void (* p_mipiCallback) (e_mipi_interrupt_type_t interrupt_flag);
    void (* p_vinCallback)  (e_mipi_interrupt_type_t interrupt_flag);
    uint32_t line_num;
} st_mipi_int_t;
```

型 メンバ	説明
e_mipi_interrupt_type_t type	MIPI および VIN の割り込み要因 「4.3 列挙型定義」e_mipi_interrupt_type_t の定義で、割り込み許可する要因を設定します
void (* p_mipiCallback) (e_mipi_interrupt_type_t interrupt_flag)	MIPI 割り込みコールバック関数 MIPI 割り込みが発生した際に呼び出される関数です NULL は設定しないでください
void (* p_vinCallback) (e_mipi_interrupt_type_t interrupt_flag)	VIN 割り込みコールバック関数 MIPI 割り込みが発生した際に呼び出される関数です NULL は設定しないでください
uint32_t line_num	スキャンライン割り込みのライン指定 0x0000～0x07FF VIN_INT_SCANLINE が type で指定された際の、割り込み発生ラインを指定します

5.7 R_MIPI_InterruptDisable

R_MIPI_InterruptDisable	
概　要	割り込み禁止設定
ヘッダ	r_mipi_api.h
宣　言	`void R_MIPI_InterruptDisable(void);`
引　数	なし
リターン値	なし
備考	なし

(1) 説明

MIPI および VIN の割り込み禁止設定を行います。また、本関数の実行によって R_MIPI_InterruptEnable で設定した MIPI および VIN の割り込みコールバック関数の登録をクリアします。

本関数では以下の処理を行います。

- MIPI および VIN の割り込み禁止

5.8 R_MIPI_GetInfo

R_MIPI_GetInfo			
概　要	キャプチャ情報取得		
ヘッダ	r_mipi_api.h		
宣　言	e_mipi_error_t R_MIPI_GetInfo(st_vin_info_type_t * infoType);		
引　数	[IN]	st_vin_info_type_t * infoType	：キャプチャ情報格納アドレス NULL は設定しないでください
リターン値	MIPI_OK		：正常終了
	MIPI_STATUS_ERR		：ドライバ内部ステータスが不正
備考	R_MIPI_Open 関数実行後に呼び出し可能です また、キャプチャ動作中に本関数を呼び出した場合はエラーとなります		

(1) 説明

現在のキャプチャフィールドと取り込みライン位置、キャプチャが完了した有効なフレームバッファ(メモリベースレジスタ)を、指定されたアドレスへ格納します。

本関数では以下の処理を行います。

- ドライバ内部ステータスチェック
- キャプチャ情報を指定アドレスへ格納

(2) パラメータ詳細

(a) st_vin_info_type_t

st_vin_info_type_t 構造体のメンバは以下の通りです。

```
typedef struct
{
    uint16_t vin_nowcaptureline;
    uint8_t vin_nowcapturefield;
    uint8_t vin_nowcapturebase;
} st_vin_info_type_t;
```

型 メンバ	説明
uint16_t vin_nowcaptureline	ラインカウント 現在のキャプチャフィールドのライン位置
uint8_t vin_nowcapturefield	現在のキャプチャフィールド 0：奇数フィールド 1：偶数フィールド
uint8_t vin_nowcapturebase	有効フレームバッファ 0：有効なフレームバッファは MB1 1：有効なフレームバッファは MB2 2：有効なフレームバッファは MB3 3：有効なフレームバッファなし

5.9 R_MIPI_CaptureStart

R_MIPI_CaptureStart			
概　要	キャプチャ開始処理		
ヘッダ	r_mipi_api.h		
宣　言	`e_mipi_error_t R_MIPI_CaptureStart(const e_mipi_capture_mode_t captureMode);`		
引　数	[IN]	const e_mipi_capture_mode_t captureMode	：キャプチャモード MIPI_SINGLE_MODE：シングルキャプチャ MIPI_CONTINUOUS_MODE：連続キャプチャ
リターン値	MIPI_OK		：正常終了
	MIPI_STATUS_ERR		：ドライバ内部ステータスが不正
備考	R_MIPI_Open 関数実行後に呼び出し可能です また、キャプチャ動作開始前に R_MIPI_Setup 関数、R_MIPI_SetBufferAdr 関数で各種キャプチャ設定を実行してください		

(1) 説明

キャプチャモードを設定し、キャプチャ動作を開始します

本関数では以下の処理を行います。

- ドライバ内部ステータスチェック
- キャプチャモード設定
- キャプチャ動作開始

5.10 R_MIPI_CaptureStop

R_MIPI_CaptureStart		
概　要	キャプチャ停止処理	
ヘッダ	r_mipi_api.h	
宣　言	`e_mipi_error_t R_MIPI_CaptureStop(void);`	
引　数	なし	
リターン値	MIPI_OK	：正常終了
	MIPI_STATUS_ERR	：ドライバ内部ステータスが不正
備考	R_MIPI_CaptureStart 関数を実行し、キャプチャ動作中のみ実行可能です。	

(1) 説明

キャプチャ動作を停止します。

本関数では以下の処理を行います。

- ドライバ内部ステータスチェック
- キャプチャ動作停止

5.11 R_MIPI_InterruptHandler

R_MIPI_InterruptHandler			
概　要	MIPI 割り込みハンドラ		
ヘッダ	r_mipi_api.h		
宣　言	`void R_MIPI_InterruptHandler( uint32_t int_sense );`		
引　数	[IN]	uint32_t int_sense	：割り込み要求 エッジ/レベル
リターン値	なし		
備考	なし		

(1) 説明

本関数は MIPI の割り込みハンドラです。割り込みハンドラの登録処理の例として準備している「5.14 R_MIPI_OnInitialize」で本関数を割り込みハンドラとして登録しています。

5.12 R_VIN_InterruptHandler

R_VIN_InterruptHandler			
概　要	VIN 割り込みハンドラ		
ヘッダ	r_mipi_api.h		
宣　言	`void R_VIN_InterruptHandler( uint32_t int_sense );`		
引　数	[IN]	uint32_t int_sense	：割り込み要求 エッジ/レベル
リターン値	なし		
備考	なし		

(1) 説明

本関数は VIN の割り込みハンドラです。割り込みハンドラの登録処理の例として準備している「5.14 R_MIPI_OnInitialize」で本関数を割り込みハンドラとして登録しています。

5.13 R_MIPI_CPUVAddrToSysPAddr

R_MIPI_CPUVAddrToSysPAddr			
概　要	取り込みバッファアドレス変換処理		
ヘッダ	r_mipi_user.h		
宣　言	`uint32_t R_MIPI_CPUVAddrToSysPAddr(uint32_t vaddr);`		
引　数	[IN]	uint32_t vaddr	：仮想アドレス
リターン値	uint32_t 型の整数		：物理アドレス
備考	なし		

(1) 説明

本関数は仮想アドレスを物理アドレスに変換します。R_MIPI_SetBufferAdr 関数内の処理で、メモリベースレジスタへアドレスを設定する際に、本関数が呼び出されます。

5.14 R_MIPI_OnInitialize

R_MIPI_OnInitialize			
概　要	MIPI および VIN のスタンバイ解除、割り込みハンドラの登録サンプル		
ヘッダ	r_mipi_user.h		
宣　言	`void R_MIPI_OnInitialize (const uint32_t user_num);`		
引　数	[IN]	const uint32_t user_num	：ユーザパラメータ
リターン値	なし		
備考	なし		

(1) 説明

MIPI および VIN のモジュールスタンバイ解除や割り込みハンドラ登録処理の例として、準備している関数です。必要に応じて、ユーザ環境に適した処理を実装してください。

本関数では以下の処理を行います。

- MIPI および VIN のスタンバイ解除
- 割り込みハンドラ登録
- 割り込み優先度設定

5.15 R_MIPI_OnFinalize

R_MIPI_OnFinalize			
概　要	MIPI および VIN のスタンバイ設定、割り込みハンドラの解除サンプル		
ヘッダ	r_mipi_user.h		
宣　言	`void R_MIPI_OnFinalize( const uint32_t user_num );`		
引　数	[IN]	const uint32_t user_num	：ユーザパラメータ
リターン値	なし		
備考	なし		

(1) 説明

MIPI および VIN のモジュールスタンバイ設定や割り込みハンドラ解除処理の例として、準備している関数です。必要に応じて、ユーザ環境に適した処理を実装してください。

本関数では以下の処理を行います。

- MIPI および VIN のスタンバイ設定
- 割り込みハンドラ解除

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。）

RTX921053C00000BE（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。）

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ
http://japan.renesas.com/

お問合せ先
http://japan.renesas.com/contact/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2018.9.14	ー	初版発行
1.01	2018.12.28	5.4	R_MIPI_Setup パラメータ追加

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. 未使用端子の処理
   【注意】未使用端子は、本文の「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。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。
2. 電源投入時の処置
   【注意】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. リザーブアドレス（予約領域）のアクセス禁止
   【注意】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、将来の機能拡張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。
4. クロックについて
   【注意】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。
5. 製品間の相違について
   【注意】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うと、内部ROM、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、ノイズ輻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も、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て、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/