

RE01 1500KB、 256KB グループ

PMIP ドライバ詳細仕様書

要旨

本書では、RE01 1500KB、256KB グループ向け CMSIS software package の PMIP ドライバ(以下、PMIP ドライバ)の仕様を説明します。

対象デバイス

マイコン : RE01 1500KB グループ、RE01 256KB グループ

パラレル MIP 液晶パネル(白黒) : TN0104ANVAANN-GN00 (Kyocera 社製)

本ドライバを他のマイコンまたは他のパラレル MIP 液晶パネルへ適用する場合、そのデバイスの仕様に合わせて変更し、十分評価してください。

目次

1. 概要	3
2. ドライバ構成	3
2.1 ファイル構成	3
2.2 コンフィグレーション	4
2.3 マクロノ型定義	5
2.3.1 型定義	5
2.3.2 PMIP エラーコード定義	6
2.3.3 PMIP イベント定義	7
2.4 構造体定義	7
2.4.1 st_transmode_t 構造体	7
2.4.2 st_mlcd_mode_info_t 構造体	7
3. 状態遷移図	7
4. ドライバ関数	8
4.1 関数仕様	8
4.1.1 R_PMIP_Open	8
4.1.2 R_PMIP_PowerOn	8
4.1.3 R_PMIP_Control	9
4.1.4 R_PMIP_Send	10
4.1.5 R_PMIP_SendTrim	11
4.1.6 R_PMIP_AllOne	11
4.1.7 R_PMIP_AllZero	12
4.1.8 R_PMIP_PowerOff	12
4.1.9 R_PMIP_Close	12
4.1.10 R_PMIP_GetVersion	12
4.1.11 R_PMIP_Reconfig	13

4.2	割り込み設定	14
5.	ドライバ使用例	14
6.	使用上の注意	16
6.1	SCLK の High 幅設定	16
6.2	関数の実行制限	16
7.	トラブルシューティング	16
7.1	MLCD_VCOM が出力されない	16
7.2	CMSIS ドライバの API をコールすると Hard Fault Error が発生する	16
7.3	API を呼び出しているが周辺機能が動作しない	16
7.4	API の戻り値は正常であるが、周辺機能の端子が機能しない	16
8.	参考ドキュメント	17

1. 概要

PMIP ドライバは、RE01 1500KB および 256KB グループにて MLCD を使用し、パラレル MIP 液晶パネルに画像データを出力するためのドライバです。

下記に本ドライバの概要を示します。

表 1-1 PMIP ドライバの概要

ドライバの動作概要	使用する周辺機能	主として使用するドライバ
MIP 液晶コントローラ (MLCD) を使用し、パラレル MIP 液晶パネルに画像データを出力します。 MLCD へのデータの入力には DMAC を使用します。	LPM、DMAC	R_LPM、R_DMCA

2. ドライバ構成

2.1 ファイル構成

PMIP ドライバは CMSIS Driver package の Device HAL に該当し、ベンダ独自ファイル格納ディレクトリ内の `r_pmip_api.c`、`r_pmip_api.h`、`r_pmip_cfg.h` の 3 個のファイルで構成されます。各ファイルの役割を表 2-1 に、ファイル構成を図 2-1 に示します。

表 2-1 R_PMIP ドライバ 各ファイルの役割

ファイル名	内容
<code>r_pmip_api.c</code>	ドライバソースファイルです。 ドライバ関数の実体を用意します。 PMIP ドライバを使用する場合は、本ファイルをビルドする必要があります。
<code>r_pmip_api.h</code>	ドライバヘッダファイルです。 ユーザが参照可能なマクロ／型／プロトタイプ宣言を用意します。 PMIP ドライバを使用する場合は、本ファイルをインクルードする必要があります。
<code>r_pmip_cfg.h</code>	コンフィグレーション定義ファイルです。 ユーザが設定可能なコンフィグレーション定義を用意します。

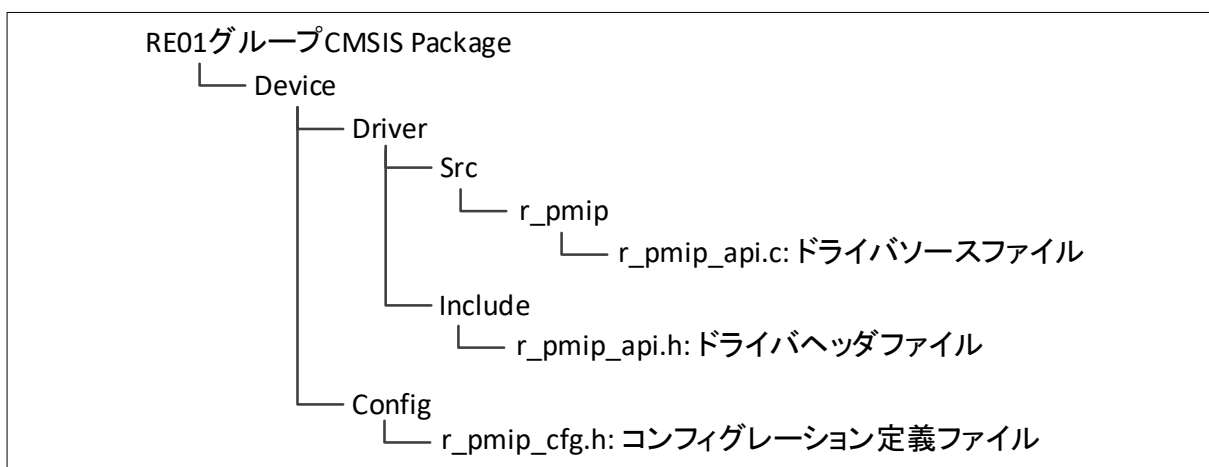


図 2-1 PMIP ドライバファイル構成

2.2 コンフィグレーション

PMIP ドライバは、ユーザが変更可能なコンフィグレーションを `r_pmip_cfg.h` ファイルおよび `r_pmip_api.h` ファイルに用意します。コンフィグレーションの一覧を表 2-2 および表 2-3 に示します。

表 2-2 `r_pmip_cfg.h` の設定内容

名称	設定内容	設定値	初期値
PMIP_CFG_DISP_WIDTH	MIP-LCD の水平方向サイズ (ピクセル数)	パラレル MIP 液晶パネルの仕様に従って設定	176
PMIP_CFG_DISP_HEIGHT	MIP-LCD の垂直方向サイズ (ピクセル数)	パラレル MIP 液晶パネルの仕様に従って設定	176
PMIP_CFG_SCLKH	送信クロックの High 幅(us)	パラレル MIP 液晶パネルの仕様に従って設定	1
PMIP_CFG_VCOM_CLK	VCOM 出力 High 幅設定	MLCDVCOMCTL.VCOMW[1:0] ビットの設定値から選択 0 : 500ms (VCOMW[1:0]=01b) 1 : 1000ms (VCOMW[1:0]=00b) 2 : 2000ms (VCOMW[1:0]=10b) 3 : 5000ms (VCOMW[1:0]=11b)	0
PMIP_CFG_PARAM_CHECK_EN	パラメータチェック機能	0 : 無効 1 : 有効	1
PMIP_CFG_ENB_TBL	イネーブル信号制御用テーブル選択	0 : Kyocera(TN0104ANVAANN-GN00) (注1) 1 : other	0
PMIP_CFG_INTERRUPT_LEVEL	MLCD の割り込み優先レベル	0(高)~3 (低)	0
PMIP_CFG_DISP_INI_DATA	PowerOn API 呼び出し時の MIP-LCD 初期化データ選択	パラレル MIP 液晶パネルの仕様に従って設定 0 : 0x00 を出力 1 : 0xFF を出力 ご使用のパネルが Normally White 時は"0"、Normally Black 時は"1"を設定します	1
PMIP_CFG_FMASK_US(注2)	データ転送前 VCOM マスク時間	データ転送前 VCOM マスク時間を設定(注4)	4000
PMIP_CFG_BMASK_US(注3)	データ転送後 VCOM マスク時間	データ転送後 VCOM マスク時間を設定(注4)	1000
PMIP_CFG_SUPPORT_DMAC	DMAC 有効/無効設定	0 : DMAC 無効 1 : DMAC 有効	1
PMIP_CFG_SUPPORT_DTC	DTC 有効/無効設定	0 : DTC 無効(注5)	0

注1. PMIP_CFG_ENB_TBL=0 を選択した場合、Kyocera 製 TN0104ANVAANN-GN00 で使用可能な最短のイネーブル信号幅が自動で設定されます。イネーブル信号幅を任意に設定する場合や他のパラレル MIP 液晶パネルを使用する場合は、PMIP_CFG_ENB_TBL=1 を設定してください。

注2. 256KB グループでのみ使用できます。設定値は MLCDVCOMCTL.FMASK[7:0] ビット設定値に変換されます。

注3. 256KB グループでのみ使用できます。設定値は MLCDVCOMCTL.BMASK[7:0] ビット設定値に変換されます。

注4. 実際のマスク時間には、設定値から 0~488us (サンプリング周期) の誤差が発生します。

[設定例]

設定値 : 4000us

マスク時間 : 4392us(488*9)

注5. DTC 転送はサポートしていません。必ず 0 を設定してください。

表 2-3 r_pmip_api.h の設定内容

名称	設定内容	設定値	初期値
OTHER_ENB (e_pmip_enb_tbl_other_t)	t _o ENB[US]および t _b ENB[US] ^(注)	使用するパラレル MIP-LCD の仕様に従って設定 ※PMIP_CFG_SCLKH より大きい値を設定	4
OTHER_ENBH (e_pmip_enb_tbl_other_t)	t _w ENBH[US] ^(注)	使用するパラレル MIP-LCD の仕様に従って設定	20

注. other 用テーブル選択時(PMIP_CFG_ENB_TBL = 1)のみ有効です。t_oENB などの詳細は、ユーザズマニュアル：ハードウェア 58.3.12 MLCD タイミング章を参照してください。

2.3 マクロ／型定義

PMIP ドライバは、ユーザが参照可能なマクロおよび型定義を r_pmip_api.h ファイルに用意します。

2.3.1 型定義

定義	値	内容
e_trans_mode_t	DMAC_TR	DMAC 転送
	CPU_TR	CPU 転送
e_trans_cmd_t	CMD0	データ送信時のビット配列
	CMD1	水平方向アドレスの自動更新方法
	CMD2	垂直方向アドレスの自動更新方法

2.3.2 PMIP エラーコード定義

PMIP のエラーコード定義です。

表 2-4 PMIP エラーコード定義一覧

定義	内容	解決方法
PMIP_OK	正常終了	-
PMIP_ERROR	MLCD モジュールクロック開始エラー	LPM ドライバのエラーが発生しています。LPM ドライバの R_LPM_ModuleStart 関数を参照してください。
	MLCD モジュールクロック停止エラー	LPM ドライバのエラーが発生しています。LPM ドライバの R_LPM_ModuleStop 関数を参照してください。
	MLCD システム初期処理エラー	1. 送信前に R_PMIP_Open 関数を実行してください 2. LPM ドライバのエラーが発生しています。LPM ドライバの仕様書を参照してください。
PMIP_ERROR_TRANS_MODE	転送方式設定エラー	R_PMIP_Open の引数 tr_mode.mode には、DMAC_TR, CPU_TR のどちらかを設定してください。
PMIP_ERROR_DMAC_TRANS	選択した DMAC チャネルでエラー発生	DMAC ドライバのエラーが発生しています。DMAC ドライバの仕様書を参照してください。
PMIP_ERROR_DMAC_CFG	DMAC のチャンネル設定エラー	R_PMIP_Open の引数 tr_mode.sel には、0~3 のいずれかを設定してください。
	選択した DMAC チャネルは使用中	選択した DMAC のチャンネルは使用中です。R_PMIP_Open の引数 tr_mode.sel に、未使用の DMAC チャネルを設定してください。
PMIP_ERROR_INPUT_CLK_OFFRANGE	送信クロック High 幅設定エラー	下記の条件を満たすように、設定してください。 PMIP_CFG_SCLKH(us)*システムクロック(Hz)/1000000 <256
PMIP_ERROR_ENBEG_CFG	ENBG 信号幅設定エラー	1. 下記の条件を満たすように、設定してください。 KYOCERA_ENB > PMIP_CFG_SCLKH または OTHER_ENB > PMIP_CFG_SCLKH 2. 算出された ENBG の幅は、オーバーフローしています。
PMIP_ERROR_CONTROL_CMD	コマンド設定エラー	R_PMIP_Control の引数 cmd には、CMD0、CMD1、CMD2 のいずれかを設定してください。
	コマンド設定値エラー	R_PMIP_Control の引数 set には、0 または 1 を設定してください。
PMIP_ERROR_SEND_CFG	送信パラメータの設定エラー	下記の条件を満たすように、設定してください。 st_h : 0~(PMIP_CFG_DISP_WIDTH-1), ただし 8 の倍数 st_v : 0~(PMIP_CFG_DISP_HEIGHT-1) size_h : 1~(PMIP_CFG_DISP_WIDTH/8) size_v : 1~PMIP_CFG_DISP_HEIGHT img_size_h : 2~ (PMIP_CFG_DISP_WIDTH/8)
PMIP_ERROR_SYSTEM_SETTING	割り込み設定エラー	割り込み優先レベル、または MLCD の割り込みを設定してください。割り込み設定とは Config フォルダの r_system_cfg.h ファイルの設定を意味します。
PMIP_ERROR_BUSY	ビジーエラー	PMIP 転送中によるビジーエラーが発生しています。このエラーが発生する処理は、PMIP 転送停止中のみ実行可能です。PMIP 転送完了後、再度実行してください。
PMIP_ERROR_RECONFIG	リコンフィグエラー	リコンフィグ処理中を示すエラーが発生しています。リコンフィグ処理中は、PMIP 転送を開始することはできません。リコンフィグ処理完了後、再度実行してください。

2.3.3 PMIP イベント定義

PMIP のイベント定義です。

定義	値	内容
PMIP_EVENT_SEND_COMPLETE	1	データ出力が正常に完了しました。
PMIP_EVENT_ERROR_DMACH_TRANS	2	データ出力で異常が発生しました。

2.4 構造体定義

PMIP ドライバでは、ユーザーが変更可能な構造体定義を `r_pmip_api.h` ファイルで定義しています。

2.4.1 `st_transmode_t` 構造体

要素名	型	内容
mode	<code>e_trans_mode_t</code>	転送モード ※DTC 転送はサポートしていません
sel	<code>uint8_t</code>	DMAC のチャンネル 0: DMAC0 1: DMAC1 2: DMAC2 3: DMAC3 ※CPU 転送選択時(mode=CPU_TR)は無効です

2.4.2 `st_mlcd_mode_info_t` 構造体

要素名	型	内容
cb_event	<code>pmip_cb_event_t</code>	コールバック関数
flags	<code>uint8_t</code>	DMA 転送時に使用するフラグ ※変更しないでください。

3. 状態遷移図

PMIP ドライバの状態遷移図を、図 3-1 に示します。

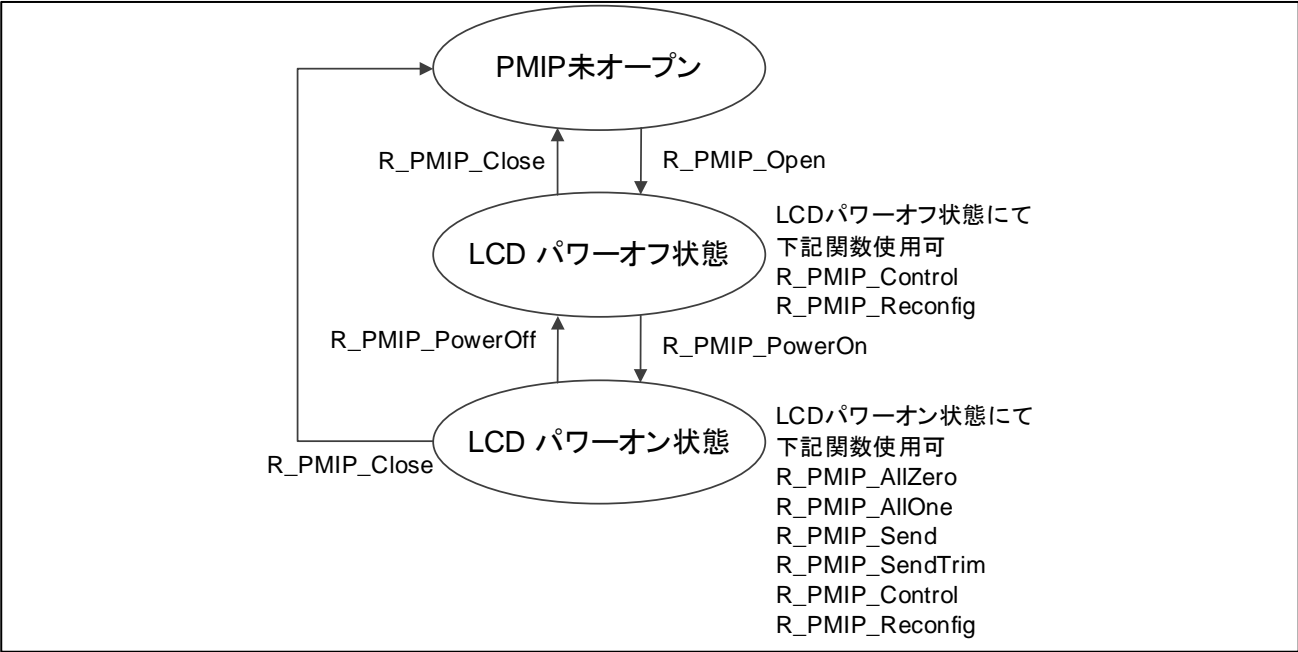


図 3-1 状態遷移図(注 1)

注 1. `R_PMIP_GetVersion` 関数はいずれの状態からでも呼び出せます

4. ドライバ関数

4.1 関数仕様

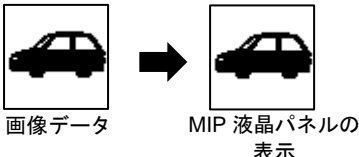
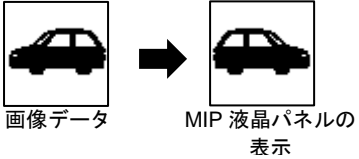
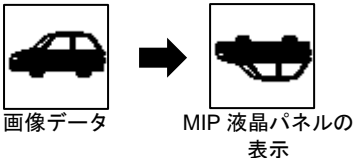
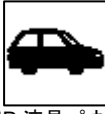


4.1.1 R_PMIP_Open

概要	システム初期処理	
書式	e_pmip_err_t R_PMIP_Open(pmip_cb_event_t cb, st_transmode_t* tr_mode)	
説明	MLCD のモジュールストップビットを解除し、焼き付け防止信号(VCOM)、メモリ書換え活性信号(ENB)、送信クロックの High 幅(us)、転送、およびコールバック関数の設定を行います。 送信クロックの High 幅設定では、本関数内で現在のシステムクロック周波数を取得し、クロック周波数の High 幅が PMIP_CFG_SCLKH と一致するように計算および設定を行います。	
引数	pmip_cb_event_t cb	コールバック関数
	tr_mode.mode	送信方法の選択
	tr_mode.sel	DMAC のチャネル選択
戻り値	PMIP_OK PMIP_ERROR PMIP_ERROR_TRANS_MODE PMIP_ERROR_DMACH_CFG PMIP_ERROR_INPUT_CLK_OFFRANGE PMIP_ERROR_ENBEG_CFG PMIP_ERROR_SYSTEM_SETTING	
備考	<ul style="list-style-type: none"> ・ R_PMIP_Open 関数を呼び出す前に<u>クロック補正回路 (CCC) を動作させてください</u>。(動作方法は 5 章参照) ・ t_{oENB} および t_{bENB} は以下の式で算出されます。 $t_{oENB}, t_{bENB} = (\text{SCLK の High 幅}) + (\text{ENBEG}[7:0] * \text{PCLKA 1cycle})$ 	

4.1.2 R_PMIP_PowerOn

概要	MIP の power on シーケンス
書式	e_pmip_err_t R_PMIP_PowerOn(void)
説明	<p>パラレル MIP-LCD の電源 ON シーケンスを行います。</p> <p>PMIP_CFG_DISP_INI_DATA が 1 の場合、パラレル MIP-LCD のすべてのビットを"1"で初期化します。</p> <p>PMIP_CFG_DISP_INI_DATA が 0 の場合、パラレル MIP-LCD のすべてのビットを"0"で初期化します。</p>
引数	なし
戻り値	PMIP_OK PMIP_ERROR
備考	

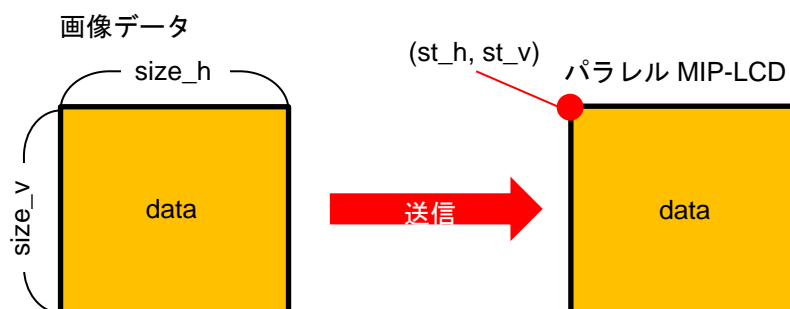
4.1.3 R_PMIP_Control

概要	送信設定	
書式	e_pmip_err_t R_PMIP_Control(e_trans_cmd_t cmd, void* set)	
説明	<p>下記の設定項目の内、いずれか 1 つを設定します</p> <ul style="list-style-type: none"> ・ データ送信時のビット配列(MLCDCR.BITSW) ・ 水平方向アドレスの自動更新方法(MLCDCR.HADDRDEC) ・ 垂直方向アドレスの自動更新方法(MLCDCR.VADDRDEC) <p>これらの設定を変更することで、画像データの左右反転表示・上下反転表示が可能です。</p>	
引数	e_trans_cmd_t cmd void* set	<p>設定項目の選択 設定内容の選択</p> <p>◆cmd = CMD0 のとき (uint8_t)set 0 : MSB 転送 1 : LSB 転送</p> <p>◆cmd = CMD1 のとき (uint8_t)set 0 : アドレスインクリメント 1 : アドレスデクリメント</p> <p>◆cmd = CMD2 のとき (uint8_t)set 0 : アドレスインクリメント 1 : アドレスデクリメント</p>
戻り値	PMIP_OK PMIP_ERROR PMIP_ERROR_CONTROL_CMD	
備考	<p>関数呼び出し例</p> <p>◆通常表示</p> <pre>R_PMIP_Control(CMD0,0); /* default */ R_PMIP_Control(CMD1,0); /* default */ R_PMIP_Control(CMD2,0); /* default */ R_PMIP_Send(0,0,32,256,img);</pre> <p>◆左右反転表示</p> <pre>R_PMIP_Control(CMD0,1); R_PMIP_Control(CMD1,1); R_PMIP_Send(248,0,32,256,img);</pre> <p>◆上下反転表示</p> <pre>R_PMIP_Control(CMD2,1); R_PMIP_Send(0,255,32,256,img);</pre> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="text-align: center; margin-right: 10px;">  画像データ </div> <div style="font-size: 2em; margin-right: 10px;">➡</div> <div style="text-align: center; margin-right: 10px;">  MIP 液晶パネルの 表示 </div> </div> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="text-align: center; margin-right: 10px;">  画像データ </div> <div style="font-size: 2em; margin-right: 10px;">➡</div> <div style="text-align: center; margin-right: 10px;">  MIP 液晶パネルの 表示 </div> </div> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="text-align: center; margin-right: 10px;">  画像データ </div> <div style="font-size: 2em; margin-right: 10px;">➡</div> <div style="text-align: center; margin-right: 10px;">  MIP 液晶パネルの 表示 </div> </div>	

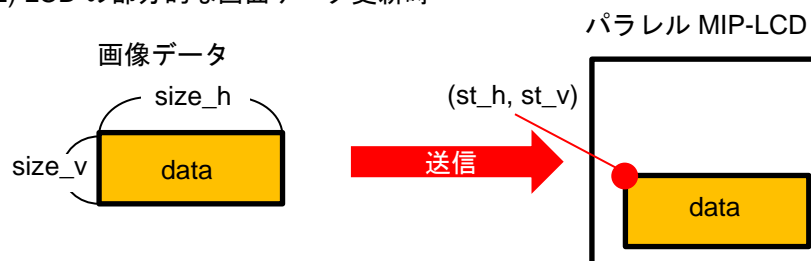
4.1.4 R_PMIP_Send

概要	データ送信
書式	e_pmip_err_t R_PMIP_Send(uint8_t st_h, uint8_t st_v, uint8_t size_h, uint16_t size_v, uint8_t* &data)
説明	パラレル MIP-LCD の座標(st_h,st_v)に画像データ(size_h×size_v ビット)を出力します。

1) LCD の全画面データ更新時



2) LCD の部分的な画面データ更新時

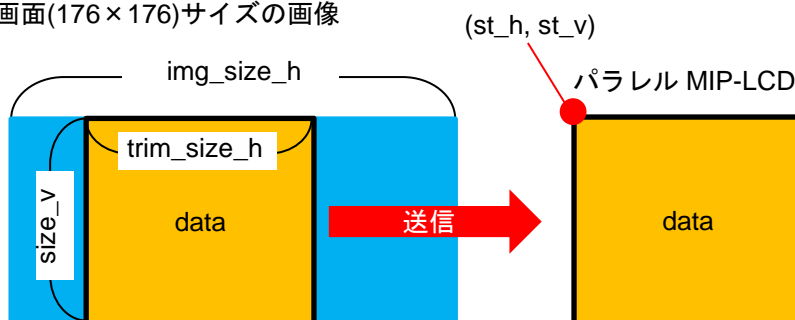


引数	uint8_t st_h	送信開始ピクセル(水平方向)(SL)
	uint8_t st_v	送信開始ピクセル(垂直方向)(GL)
	uint8_t size_h	送信データサイズ(水平方向バイト数)
	uint16_t size_v	送信データサイズ(垂直方向行数)
	uint8_t* &data	送信データのメモリアドレス
戻り値	PMIP_OK	
	PMIP_ERROR	
	PMIP_ERROR_DMAC_TRANS	
	PMIP_ERROR_SEND_CFG	
	PMIP_ERROR_SYSTEM_SETTING	
備考	送信開始ピクセル(SL)は 8 の倍数となるように設定してください。	

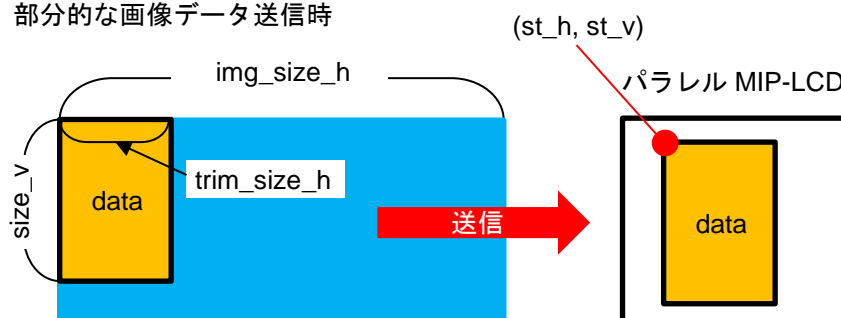
4.1.5 R_PMIP_SendTrim

概要	データ送信
書式	e_pmip_err_t R_PMIP_SendTrim(uint8_t st_h, uint8_t st_v, uint16_t img_size_h, uint8_t trim_size_h, uint16_t size_v, uint8_t* &data)
説明	パラレル MIP-LCD の座標(st_h,st_v)に画像データ(trim_size_h×size_v ビット)を出力します。画像データの水平方向サイズはimg_size_hであり、切り抜きサイズ trim_size_h 部分を切り出して出力します。

1) 全画面(176×176)サイズの画像



2) 部分的な画像データ送信時



引数	uint8_t st_h	送信開始ピクセル(水平方向)(SL)
	uint8_t st_v	送信開始ピクセル(垂直方向)(GL)
	uint16_t img_size_h	送信データサイズ(水平方向バイト数)
	uint8_t trim_size_h	送信データ切り抜きサイズ(水平方向バイト数)
	uint16_t size_v	送信データサイズ(垂直方向行数)
	uint8_t* &data	送信データのメモリアドレス
戻り値	PMIP_OK PMIP_ERROR PMIP_ERROR_DMAC_TRANS PMIP_ERROR_SEND_CFG PMIP_ERROR_SYSTEM_SETTING	
備考	送信開始ピクセル(SL)は 8 の倍数となるように設定してください。	

4.1.6 R_PMIP_AllOne

概要	全画面 1 データ送信
書式	e_pmip_err_t R_PMIP_AllOne(void)
説明	パラレル MIP-LCD のすべてのビットに"1"を出力します。
引数	なし
戻り値	PMIP_OK PMIP_ERROR
備考	

4.1.7 R_PMIP_AllZero

概要	全画面 0 データ送信
書式	e_pmip_err_t R_PMIP_AllZero(void)
説明	パラレル MIP-LCD のすべてのビットに"0"を出力します。
引数	なし
戻り値	PMIP_OK PMIP_ERROR
備考	

4.1.8 R_PMIP_PowerOff

概要	MIP の power off シーケンス
書式	e_pmip_err_t R_PMIP_PowerOff(void)
説明	パラレル MIP-LCD の電源 OFF シーケンスを行います。
引数	なし
戻り値	PMIP_OK PMIP_ERROR
備考	

4.1.9 R_PMIP_Close

概要	システム終了処理
書式	e_pmip_err_t R_PMIP_Close(void)
説明	R_SMIP_Open で設定した値を初期化し、MLCD のモジュールストップビットをストップ状態に設定し、システム終了処理を行います。
引数	なし
戻り値	PMIP_OK PMIP_ERROR
備考	CCC は停止させません。停止させる場合は、ユーザのプログラムにて停止させてください。

4.1.10 R_PMIP_GetVersion

概要	API バージョン取得
書式	uint32_t R_PMIP_GetVersion(void)
説明	API バージョンを取得します
引数	なし
戻り値	PMIP ドライバのバージョン
備考	

4.1.11 R_PMIP_Reconfig

概要	PMIP クロック再設定処理
書式	e_pmip_err_t R_PMIP_Reconfig(void)
説明	現在のシステムクロック周波数を取得し、メモリ書換え活性信号(ENB)、送信クロックの High 幅(us)を再設定します。 クロック周波数の High 幅が PMIP_CFG_SCLKH と一致するように計算および設定を行います。
引数	なし
戻り値	PMIP_OK PMIP_ERROR PMIP_ERROR_RECONFIG PMIP_ERROR_BUSY PMIP_ERROR_INPUT_CLK_OFFRANGE PMIP_ERROR_ENBEG_CFG
備考	MLCD データ送信動作中は使用できません (PMIP_ERROR_BUSY が返ります)。

4.2 割り込み設定

通信制御で使用する割り込みは、r_system_cfg.hにてNVICに登録する必要があります。割り込み(NVIC)の詳細は「RE01 1500KB、256KB グループ CMSIS パッケージを用いた開発スタートアップガイド (r01an4660)」の「割り込み制御」を参照してください。PMIP ドライバで使用する割り込み定義を表 4-1 に、割り込み登録例を図 4-1 に示します。

表 4-1 NVIC の登録定義

NVIC 登録定義
SYSTEM_CFG_EVENT_NUMBER_MLCD_TEI
SYSTEM_CFG_EVENT_NUMBER_MLCD_TEMI

```
#define SYSTEM_CFG_EVENT_NUMBER_SCI5_TXI
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 1/9/17/25 only */
#define SYSTEM_CFG_EVENT_NUMBER_MLCD_TEI
    (SYSTEM_IRQ_EVENT_NUMBER9) /*!< Numbers 1/9/17/25 only */
#define SYSTEM_CFG_EVENT_NUMBER_PORT_IRQ5
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 5/13/21/29 only */
. . .
#define SYSTEM_CFG_EVENT_NUMBER_SPI1_SPII
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 2/10/18/26 only */
#define SYSTEM_CFG_EVENT_NUMBER_MLCD_TEMI
    (SYSTEM_IRQ_EVENT_NUMBER10) /*!< Numbers 2/10/18/26 only */
#define SYSTEM_CFG_EVENT_NUMBER_PORT_IRQ9
    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED) /*!< Numbers 2/10/18/26 only */
. . .
```

図 4-1 r_system_cfg.h での NVIC への割り込み登録例

5. ドライバ使用例

PMIP ドライバを使用して画像を出力する場合の設定例を図 5-1 に示します。

```
#include "r_pmip_api.h"
#include "r_system_api.h"
#include "r_lpm_api.h"

static void callback(uint32_t event);
void set_ccc(void);
extern const uint8_t image[3872]; /* 176×176bit の白黒画像データ */
main()
{
    st_transmode_t tmd;
    tmd.mode = DMAC_TR;
    tmd.sel = 0; // DMAC channel select

    set_ccc();
```

```
(void)R_PMIP_Open((pmip_cb_event_t)&callback, &tmd); // Initialization of system
(void)R_PMIP_PowerOn(); // MIP power ON sequence
(void)R_PMIP_Control(CMD0, (void*)1);

While(1)
{
    R_SYS_SoftwareDelay(1000, SYSTEM_DELAY_UNITS_MILLISECONDS);
    (void)R_PMIP_Send(0, 0, 22, 176, image);
    R_SYS_SoftwareDelay(1000, SYSTEM_DELAY_UNITS_MILLISECONDS);
    (void)R_PMIP_AllZero();
}

static void callback(uint32_t event)
{
    /* すべての転送完了した際の処理を記述 */
    switch(event)
    {
        case PMIP_EVENT_SEND_COMPLETE:
        {
            /* 送信が正常に完了した場合の処理を記載 */
        }
        break;
        case PMIP_EVENT_ERROR_DMAC_TRANS:
        {
            /* 送信でエラーが発生した場合の処理を記載 */
        }
        break;
    }
}

void set_ccc(void)
{
    /* PMIP(MLCD) ドライバの VCOM 動作には CCC の設定が必要 */
    R_LPM_ModuleStart(LPM_MSTP_CCC); // release of CCC module stop
    R_SYS_RegisterProtectEnable(SYSTEM_REG_PROTECT_CGC);
    R_SYS_SubOscSpeedClockStart();
    R_SYS_SoftwareDelay(1000, SYSTEM_DELAY_UNITS_MILLISECONDS); // wait 1s
    CCC->R128CTRL_b.CADJUSCEN = 1; /* Start CCC */
}
```

図 5-1 PMIP ドライバ設定および送信例

6. 使用上の注意

6.1 SCLK の High 幅設定

MLCDCR.SCKCR ビットの設定値は、PMIP_CFG_SCLKH と PCLKA より算出されます。MLCDCR.SCKCR ビットが 1 のとき、実際に MLCD_SCLK 端子から出力される信号の High 幅は、NOP の時間が加算され 2cycle(PCLKA 換算)となります。PMIP_CFG_SCLKH で設定した High 幅と一致しないため、ご注意ください。詳細は、ユーザズマニュアル：ハードウェア 「58.3.12 MLCD タイミング」を参照してください。

6.2 関数の実行制限

本ドライバにおける関数の実行制限一覧を表 6-1 に示します。

表 6-1 関数実行制限

関数名	実行制限
R_PMIP_Open	PMIP 未オープン状態でのみ使用可能
R_PMIP_PowerOn	PMIP オープン状態かつ Reconfg 未実行状態でのみ使用可能
R_PMIP_Control	
R_PMIP_Send	PMIP オープン状態、LCD パワーオン状態 Reconfg 未実行状態でのみ使用可能
R_PMIP_SendTrim	
R_PMIP_AllOne	
R_PMIP_AllZero	
R_PMIP_PowerOff	
R_PMIP_Close	PMIP オープン状態でのみ使用可能
R_PMIP_GetVersion	-
R_PMIP_Reconfig	PMIP オープン状態かつ 画像データ出力停止中にのみ使用可能

7. トラブルシューティング

7.1 MLCD_VCOM が出力されない

- A) CCC クロックが設定されているか確認してください。

R_PMIP_Open 関数の実行前に、CCC クロックの出力設定を実施してください。

詳細は、「5 ドライバ使用例」をご参照ください。

7.2 CMSIS ドライバの API をコールすると Hard Fault Error が発生する

- A) API の RAM 展開が出来ていない可能性があります。

RAM 上に配置した API をコールする前に R_SYS_CodeCopy 関数にて API を RAM 展開しているか確認してください。詳細は、関連ドキュメント No. R01AN4660 をご参照ください。

7.3 API を呼び出しているが周辺機能が動作しない

- A) API の戻り値を確認し、エラー値が返っていないかをご確認ください。特に r_system_cfg.h の割り込み設定がされていないことでエラー値が返っている事例が多く発生しています。詳細は、関連ドキュメント No. R01AN4660 をご参照ください。

7.4 API の戻り値は正常であるが、周辺機能の端子が機能しない

- A) Pin.c の中にある関数で端子設定が正しく行えているか確認してください。

詳細は、関連ドキュメント No. R01AN4660 をご参照ください。

8. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

RE01 1500KB グループ ユーザーズマニュアル ハードウェア編 R01UH0796

RE01 256KB グループ ユーザーズマニュアル ハードウェア編 R01UH0894

(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。)

RE01 グループ CMSIS Package スタートアップガイド

RE01 1500KB、256KB グループ CMSIS パッケージを用いた開発スタートアップガイド R01AN4660

RE01 1500KB、256KB グループ CMSIS ドライバ R_LPM 仕様書 R01AN4838

(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。)

RE01 1500KB、256KB グループ CMSIS ドライバ DMAC 仕様書 R01AN4730

(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。)

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：開発環境

(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
1.00	Aug.30.2019	—	初版
1.01	Nov.26.2019	4 14 プログラム	液晶パネルの型名を修正 参照先を修正 r_pmip_cfg.h の RAM/ROM 配置不備を修正。 以下の内部関数が RAM に設定しても RAM に配置されない問題を改修。 e_mlcd_cpu_normal_trans 関数
1.03	Feb.21.2020	4 8 プログラム	以下のコンフィグレーション定義を追加 ・ PMIP_CFG_DISP_INI_DATA R_PMIP_PowerOn 関数の説明を追加 ・ PowerOn API 呼び出し時の MIP-LCD 初期化データ選択用定義(PMIP_CFG_DISP_INI_DATA)の追加 ・ R_PMIP_Send 関数の RAM/ROM 配置不備を修正 ・ 内部変数（リソース）の RAM/ROM 配置不備を修正 ・ R_PMIP_PowerOff 関数での VCOM 待ち時間を 2ms に変更
1.04	Mar.5.2020	— 4 プログラム (256KB)	256KB グループに対応 以下のコンフィグレーション定義を追加 ・ PMIP_CFG_FMASK_US ・ PMIP_CFG_BMASK_US ・ 256KB の IO デファインにあわせて修正 ・ VCOM マスク制御をソフトウェア制御からハードウェア制御に変更 ・ VCOM マスク時間設定用コンフィグレーション定義を追加 - PMIP_CFG_FMASK_US - PMIP_CFG_BMASK_US
1.05	Apr.23.2020	4 プログラム	以下の定義を追加 ・ PMIP_CFG_SUPPORT_DMAC ・ PMIP_CFG_SUPPORT_DTC PMIP_CFG_VCOM_CLK の記述を UMH に合わせて修正 DMAC および DTC ドライバがない状態でビルドできる形に構成を変更 asm("nop");を__NOP();に置き換え
1.06	Nov.05.2020	—	誤記修正
1.07	Jan.12.2021	13 16 プログラム	R_PMIP_Reconfig 関数を追加 「表 6-1 関数実行制限」を追加 R_PMIP_Reconfig 関数を追加

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. 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ーやマガジンケース、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、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プ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うと、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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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しします。
13. 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。
14. 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.5.0-1 2020.10)

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。