

RX ファミリ

LCD モジュール Firmware Integration Technology

要旨

本アプリケーションノートは、Firmware Integration Technology (FIT)を使用した LCD モジュールについて説明します。本モジュールは LCD コントローラ/ドライバ(LCDC)を使用して、LCD に表示を行います。以降、本モジュールを LCDC FIT モジュールと称します。

対象デバイス

- ・ RX113 グループ
- ・ RX23E-B グループ

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、十分評価してください。

対象コンパイラ

- ・ Renesas Electronics C/C++ Compiler Package for RX Family
- ・ GCC for Renesas RX
- ・ IAR C/C++ Compiler for Renesas RX

各コンパイラの動作確認内容については 5.1 動作確認環境を参照してください。

関連ドキュメント

ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)

目次

1. 概要	3
1.1 LCDC FIT モジュールとは	3
1.2 LCDC FIT モジュールの概要	3
1.3 API の概要	3
1.4 状態遷移図	4
2. API 情報	5
2.1 ハードウェアの要求	5
2.2 ソフトウェアの要求	5
2.3 サポートされているツールチェーン	5
2.4 ヘッダファイル	5
2.5 整数型	5
2.6 コンパイル時の設定	6
2.7 コードサイズ	9
2.8 戻り値	10
2.9 FIT モジュールの追加方法	11
2.10 for 文、while 文、do while 文について	12
3. API 関数	13
R_LCDC_Open ()	13
R_LCDC_Write ()	15
R_LCDC_Modify ()	16
R_LCDC_Dispon ()	17
R_LCDC_DisponOff ()	18
R_LCDC_OperateCircuit ()	20
R_LCDC_Control ()	21
R_LCDC_AdjustContrast()	23
R_LCDC_Close ()	25
R_LCDC_GetVersion ()	26
4. 端子設定	27
5. 付録	28
5.1 動作確認環境	28
5.2 トラブルシューティング	30
6. 参考ドキュメント	31
テクニカルアップデートの対応について	31
改訂記録	32

1. 概要

1.1 LCDC FIT モジュールとは

本モジュールは API として、プロジェクトに組み込んで使用します。本モジュールの組み込み方については、「2.9FIT モジュールの追加方法」を参照してください。

1.2 LCDC FIT モジュールの概要

以下に本モジュールがサポートしている機能を列挙します。

- 駆動電圧生成回路として、外部抵抗分割方式、内部昇圧方式、容量分割方式を選択可能
- 表示バイアス法として、1/2 バイアス法、1/3 バイアス法、1/4 バイアス法を選択可能
- 表示時分割数として、スタティック、2 時分割、3 時分割、4 時分割、8 時分割を選択可能
- 表示波形として、A 波形、B 波形を選択可能
- 表示データ領域は、A パターン、B パターン、点滅が選択でき、表示切り替えが可能
- RTC の周期(PRD)割り込みを用いて A パターンと B パターンの点滅が可能
- 昇圧回路動作時に生成する基準電圧を 16 段階から選択可能 (コントラスト調整)

1.3 API の概要

表 1.1 に本モジュールに含まれる API 関数を示します。

表 1.1 API 関数一覧

関数	関数説明
R_LCDC_Open	LCDC の初期設定を行い、LCD が表示できる状態にします。
R_LCDC_Write	表示データを設定します。
R_LCDC_Modify	表示データを変更します。
R_LCDC_Dispon	LCD 表示をします。
R_LCDC_DisponOff	LCD 表示オフおよび内部昇圧/容量分割回路を停止にします。
R_LCDC_Control	駆動電圧方式、表示データ領域を変更します。
R_LCDC_AdjustContrast	コントラストを調整します。
R_LCDC_OperateCircuit	内部昇圧/容量分割回路の動作許可、停止を行います。
R_LCDC_Close	LCDC を停止します。
R_LCDC_GetVersion	本モジュールのバージョン番号を返します。

1.4 状態遷移図

図 1.1 に、本モジュールの状態遷移図を示します。各状態から遷移する際は、矢印横に記載されている関数を呼び出してください。

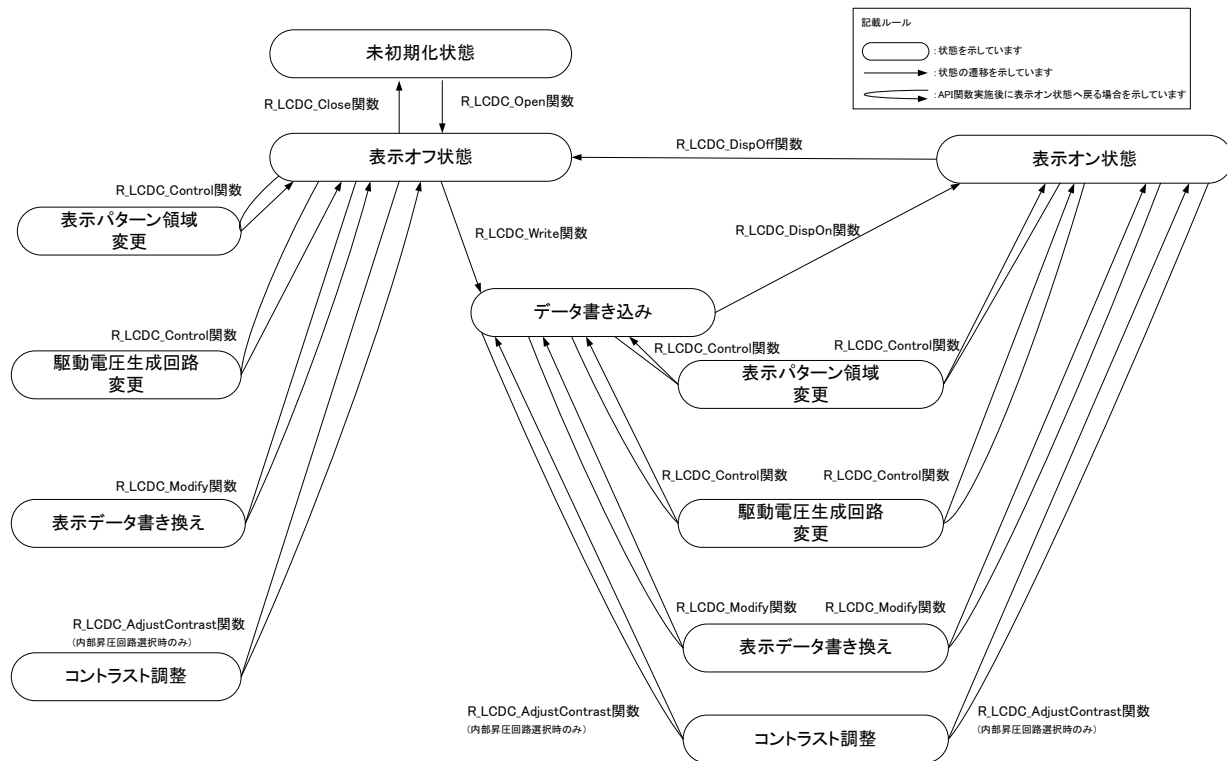


図 1.1 状態遷移図

2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

2.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- LCDC

2.2 ソフトウェアの要求

このドライバは以下の FIT モジュールに依存しています。

- ボードサポートパッケージ (r_bsp) v5.00 以上

2.3 サポートされているツールチェーン

本 FIT モジュールは「5.1 動作確認環境」に示すツールチェーンで動作確認を行っています。

2.4 ヘッダファイル

すべての API 呼び出しとそれをサポートするインタフェース定義は r_lcdc_rx_if.h に記載しています。

2.5 整数型

このドライバは ANSI C99 を使用しています。これらの型は stdint.h で定義されています。

2.6 コンパイル時の設定

本モジュールのコンフィギュレーションオプションの設定は、`r_lcdc_rx_config.h`で行います。
オプション名および設定値に関する説明を、下表に示します。

Configuration options in <code>r_lcdc_rx_config.h</code>	
#define LCDC_CFG_PARAM_CHECKING_ENABLE ※デフォルト値は“BSP_CFG_PARAM_CHECKING_ENABLE”	パラメータチェック処理をコードに含めるか選択できます。 “0”の場合、パラメータチェック処理をコードから省略します。 “1”の場合、パラメータチェック処理をコードに含めます。
#define LCDC_CFG_DRV_GENERATOR ※デフォルト値は“1”	LCD 駆動電圧生成回路を設定してください。 “0”の場合、外部抵抗分割方式を使用します。 “1”の場合、内部昇圧方式を使用します。 “2”の場合、容量分割方式を使用します。
#define LCDC_CFG_VOLTAGE_WAIT ※ デフォルト値は“1”	昇圧ウェイト時間を本モジュール内で行うかご自身のプログラムで行うかを選択してください。昇圧ウェイト時間はハードウェアマニュアルを参照してください。 “0”の場合、本モジュール内で昇圧ウェイト時間を待ちません。ご自身のプログラムで実施してください。 “1”の場合、本モジュール内で昇圧ウェイト時間を待ちます。
#define LCDC_CFG_BIAS ※デフォルト値は“3”	LCD 表示バイアス法を設定してください。 “2”の場合、1/2バイアス法を使用します。 “3”の場合、1/3 バイアス法を使用します。 “4”の場合、1/4 バイアス法を使用します。
#define LCDC_CFG_TIME_SLICES ※デフォルト値は“4”	LCD 表示時分割数を設定してください。 “1”の場合、スタティックを使用します。 “2”の場合、2 時分割を使用します。 “3”の場合、3 時分割を使用します。 “4”の場合、4 時分割を使用します。 “8”の場合、8 時分割を使用します。
#define LCDC_CFG_DRV_WAVEFORM ※デフォルト値は“0”	LCD表示波形を設定してください。 “0”の場合、A波形を使用します。 “1”の場合、B 波形を使用します。

<pre>#define LCDC_CFG_CLOCK_DIV</pre> ※デフォルト値は"23"	LCDのクロック周波数の分周を設定してください。 分周の求め方は以下のとおりです。 分周 = 使用されるクロック周波数÷時分割数÷フレーム周波数 もっとも近い値を以下から選択してください。 ・ LCDのクロックソースに"メインクロック、HOCO、LOCO"を選択した場合 : "17"の場合、256分周を設定します。 "18"の場合、512分周を設定します。 "19"の場合、1024分周を設定します。 "20"の場合、2048分周を設定します。 "21"の場合、4096分周を設定します。 "22"の場合、8192分周を設定します。 "23"の場合、16384分周を設定します。 "24"の場合、32768分周を設定します。 "25"の場合、65536分周を設定します。 "26"の場合、131072分周を設定します。 "27"の場合、262144分周を設定します。 "43"の場合、524288分周を設定します。 ・ LCDのクロックソースに"サブクロック"を選択した場合 : "1"の場合、4分周を設定します。 "2"の場合、8分周を設定します。 "3"の場合、16分周を設定します。 "4"の場合、32分周を設定します。 "5"の場合、64分周を設定します。 "6"の場合、128分周を設定します。 "7"の場合、256分周を設定します。 "8"の場合、512分周を設定します。 "9"の場合、1024分周を設定します。 ・ LCDのクロックソースに"IWDT専用オンチップオシレータ"を選択した場合 : "1"の場合、4分周を設定します。 "2"の場合、8分周を設定します。 "3"の場合、16分周を設定します。 "4"の場合、32分周を設定します。 "5"の場合、64分周を設定します。 "6"の場合、128分周を設定します。 "7"の場合、256分周を設定します。
<pre>#define LCDC_CFG_REF_VCC</pre> ※デフォルト値は"12"	基準電圧を設定してください。この定数は、駆動電圧生成回路に内部昇圧を選択した場合のみ有効です。基準電圧、1/3バイアス法、1/4バイアス法によって設定値が異なります。 ・ VL1電圧 (基準電圧)の場合 "4"の場合、1.00Vを設定します。 "5"の場合、1.05Vを設定します。 "6"の場合、1.10Vを設定します。 "7"の場合、1.15Vを設定します。 "8"の場合、1.20Vを設定します。 "9"の場合、1.25Vを設定します。 "10"の場合、1.30Vを設定します。 "11"の場合、1.35Vを設定します。

	“12”の場合、1.40Vを設定します。 “13”の場合、1.45Vを設定します。 “14”の場合、1.50Vを設定します。 “15”の場合、1.55Vを設定します。 “16”の場合、1.60Vを設定します。 “17”の場合、1.65Vを設定します。 “18”の場合、1.70Vを設定します。 “19”の場合、1.75Vを設定します。 ・ VL4電圧(1/3バイアス法時)の場合 “4”の場合、3.00Vを設定します。 “5”の場合、3.15Vを設定します。 “6”の場合、3.30Vを設定します。 “7”の場合、3.45Vを設定します。 “8”の場合、3.60Vを設定します。 “9”の場合、3.75Vを設定します。 “10”の場合、3.90Vを設定します。 “11”の場合、4.05Vを設定します。 “12”の場合、4.20Vを設定します。 “13”の場合、4.35Vを設定します。 “14”の場合、4.50Vを設定します。 “15”の場合、4.65Vを設定します。 “16”の場合、4.80Vを設定します。 “17”の場合、4.95Vを設定します。 “18”の場合、5.10Vを設定します。 “19”の場合、5.25Vを設定します。 ・ VL4電圧(1/4バイアス法時)の場合 “4”の場合、4.00Vを設定します。 “5”の場合、4.20Vを設定します。 “6”の場合、4.40Vを設定します。 “7”の場合、4.60Vを設定します。 “8”の場合、4.80Vを設定します。 “9”の場合、5.00Vを設定します。 “10”の場合、5.20Vを設定します。
--	---

2.7 コードサイズ

本モジュールの ROM サイズ、RAM サイズ、最大使用スタックサイズを下表に示します。

ROM (コードおよび定数) と RAM (グローバルデータ) のサイズは、ビルド時の「2.6 コンパイル時の設定」のコンフィギュレーションオプションによって決まります。

下表の値は下記条件で確認しています。

モジュールリビジョン: r_lcdc_rx rev2.00

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.04.00

(統合開発環境のデフォルト設定に"-lang = c99"オプションを追加)

GCC for Renesas RX 8.03.00.202202

(統合開発環境のデフォルト設定に"-std=gnu99"オプションを追加)

IAR C/C++ Compiler for Renesas RX version 4.20.3

(統合開発環境のデフォルト設定)

コンフィギュレーションオプション: デフォルト設定

ROM、RAM およびスタックのコードサイズ							
デバイス	分類	使用メモリ					
		Renesas Compiler		GCC		IAR Compiler	
		パラメータ チェックあり	パラメータ チェックなし	パラメータ チェックあり	パラメータ チェックなし	パラメータ チェックあり	パラメータ チェックなし
RX113 RX23E-B	ROM	1,035 バイト	959 バイト	1,385 バイト	1,301 バイト	1,351 バイト	1,268 バイト
	RAM	2 バイト		2 バイト		2 バイト	
	スタック	120 バイト		-		56 バイト	

2.8 戻り値

API 関数の戻り値を示します。この列挙型は API 関数のプロトタイプ宣言とともに `r_lcdc_rx_if.h` で記載されています。

```
typedef enum
{
    LCDC_ERR_ADJUST_PARAM = -2,      /* R_LCDC_AdjustContrast のパラメータが不正な場合 */
    LCDC_ERR_OUTSIDE_VOLTAGE = -1,   /* 指定した電圧が不正な場合 */
    LCDC_SUCCESS,                    /* 問題なく処理が終了した場合 */
    LCDC_ERR_NOT_CLOSED,              /* すでにロックされている場合 */
    LCDC_ERR_NOT_OPEN,                /* R_LCDC_Open 関数が実行されていない場合 */
    LCDC_ERR_INVALID_ARG,             /* 引数が不適切な場合 */
    LCDC_ERR_UNSUPPORTED,             /* 表示モードの組み合わせが不適切な場合 */
    LCDC_ERR_NOT_OPERATE,             /* 指定されたクロックソースが動作していない場合 */
    LCDC_ERR_RTC_NOT_OPERATE         /* 点滅選択時に RTC が動作していない場合 */
} lcdc_err_t;
```

2.9 FIT モジュールの追加方法

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、スマート・コンフィグレータを使用した(1)、(3)、(5)の追加方法を推奨しています。ただし、スマート・コンフィグレータは、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)、(4)の方法を使用してください。

- (1) e² studio 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
e² studio のスマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: e² studio 編 (R20AN0451)」を参照してください。
- (2) e² studio 上で FIT コンフィグレータを使用して FIT モジュールを追加する場合
e² studio の FIT コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。
- (3) CS+上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
CS+上で、スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: CS+編 (R20AN0470)」を参照してください。
- (4) CS+上で FIT モジュールを追加する場合
CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。
- (5) IAREW 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: IAREW 編 (R20AN0535)」を参照してください。

2.10 for 文、while 文、do while 文について

本モジュールでは、レジスタの反映待ち処理等で for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合は、「WAIT_LOOP」で該当の処理を検索できます。

以下に記述例を示します。

while 文の例：

```
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}
```

for 文の例：

```
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}
```

do while 文の例：

```
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

3. API 関数

R_LCDC_Open ()

LCDC の初期設定を行い、LCD 表示できる状態にする関数です。この関数は他の API 関数を使用する前に実行される必要があります。

Format

```
lcdc_err_t R_LCDC_Open(void)
```

Parameters

なし

Return Values

LCDC_SUCCESS	<i>/* 問題なく処理が完了した場合 */</i>
LCDC_ERR_NOT_CLOSED	<i>/* すでにロックされている場合 */</i>
LCDC_ERR_UNSUPPORTED	<i>/* 表示モードの組み合わせが不適切な場合 */</i>
LCDC_ERR_NOT_OPERATE	<i>/* 指定されたクロックソースが動作していない場合 */</i>

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

LCDC がロックされていない場合、LCDC を使用するために以下の初期設定を行います。

- LCDC をロック
- LCDC のモジュールストップ状態の解除
- 昇圧端子初期値切り替え制御ビットの設定 (駆動電圧生成回路が内部昇圧の場合のみ)
- バイアス法・時分割数・波形の設定
- 駆動電圧生成回路の設定
- 表示データ領域の設定 (デフォルトは A パターン領域)
- LCD のクロックソースの設定(クロックソースの選択は、BSP モジュールの定数 BSP_CFG_LCD_CLOCK_SOURCE で設定してください)
- 電圧レベルの設定 (駆動電圧生成回路が内部昇圧の場合のみ)
- 内部昇圧/容量分割回路を動作 (駆動電圧生成回路が内部昇圧方式、容量分割方式の場合のみ)

この関数を実行する前に、以下の処理を行ってください。

- LCD のクロックソースの発振
- PmnPFS.PSEL[4:0]ビットでコモン端子/セグメント端子、VL1、VL2、VL3、VL4、CAPH、CAPL 端子を設定
(上記を設定する前に PMR レジスタ、PDR レジスタ、PCR レジスタを“0”に設定してください。
また設定後に PMR レジスタを“1”に設定してください。)

Example

```
lcdc_err_t lcdc_err;  
lcdc_err = R_LCDC_Open();
```

Special Notes:

駆動電圧生成回路、表示波形、時分割数、バイアス法の設定は `r_lcdc_rx_config.h` で行ってください。その際、以下の組み合わせ表を参考にしてください。

表示モード			駆動電圧生成回路		
表示波形	時分割数	バイアス法	外部抵抗分割	内部昇圧	容量分割
A 波形	8	1/4	○	○	-
A 波形	4	1/3	○	○	○
A 波形	3	1/3	○	○	○
A 波形	3	1/2	○	-	-
A 波形	2	1/2	○	-	-
A 波形	スタティック		○	-	-
B 波形	8	1/4	○	○	-
B 波形	4	1/3	○	○	○

○：対応、-：非対応

R_LCDC_Write ()

表示データを設定する関数です。

Format

```
lcdc_err_t R_LCDC_Write(  
    uint8_t seg, /* 書き込むセグメントの指定 */  
    uint8_t data /* 書き込むデータ */  
)
```

Parameters

seg

書き込むセグメントを指定してください。

data

指定したセグメントに書き込む表示データを設定してください。

Return Values

<i>LCDC_SUCCESS</i>	<i>/* 問題なく処理が完了した場合 */</i>
<i>LCDC_ERR_INVALID_ARG</i>	<i>/* 引数が不適切な場合 */</i>

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

初期の表示データを設定します。8 時分割以外の場合、A パターン領域に書き込む際は下位 4 ビット、B パターン領域に書き込む際は上位 4 ビットに値を格納してください。

- 表示データレジスタに表示データを格納

Example

- 4 時分割表示で"2"を表示する場合

```
volatile lcdc_err_t lcdc_err;  
  
lcdc_err = R_LCDC_Write(10, 0x06);  
lcdc_err = R_LCDC_Write(11, 0x0b);
```

Special Notes:

なし

R_LCDC_Modify ()

表示データを書き換えるときに使用する関数です。

Format

```
lcdc_err_t R_LCDC_Modify(
    uint8_t seg,          /* 書き込むセグメントの指定 */
    uint8_t data_mask,    /* 書き換え箇所の指定 */
    uint8_t data          /* 書き換えるデータ */
)
```

Parameters

seg

書き込むセグメントを指定してください。

data_mask

書き換えたいビットを“0”、それ以外を“1”に設定してください。1ビットだけを書き換えたい場合は、下表を参考にしてください。複数ビットの変更も可能です。

COM	COM3 (bit3)	COM2 (bit2)	COM1 (bit1)	COM0(bit0)
data_mask の設定値	0xf7	0xfb	0xfd	0xfe

data

指定したセグメントに書き込む表示データを設定します。

書き換えを行わないビット（data_mask で“1”を設定したビット）には“0”を設定してください。

Return Values

LCDC_SUCCESS

/ 問題なく処理が完了した場合 */*

LCDC_ERR_INVALID_ARG

/ 引数が不適切な場合 */*

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

LCD の表示データをビット単位で書き換えます。指定されたビット以外の値はそのまま保持します。この関数を実行する前に、R_LCDC_Open 関数を実行してください。

-表示されている表示データにマスク処理を実施

-変更するデータを設定

Example

- SEG0 の COM0(0bit)、COM1(1bit)を変更する場合

```
lcdc_err_t lcdc_err;
```

```
lcdc_err = R_LCDC_Modify(0,0xfc, 0x03);
```

Special Notes:

なし

R_LCDC_DispOn ()

示データを設定し、LCD 表示を行う関数です。

Format

lcdc_err_t R_LCDC_DispOn(void)

Parameters

なし

Return Values

LCDC_SUCCESS */* 問題なく処理が完了した場合 */*

LCDC_ERR_NOT_OPEN */* R_LCDC_Open 関数が実行されていない場合 */*

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

LCD 表示をオンにします。この関数を実行する前に、R_LCDC_Open 関数を実行してください。

- LCD の表示許可

Example

```
volatile lcdc_err_t lcdc_err;

/* LCD コントローラの設定 */
lcdc_err = R_LCDC_Open();

/* 表示データの書き込み */
lcdc_err = R_LCDC_Write(10, 0x06);
lcdc_err = R_LCDC_Write(11, 0x0b);

/* 表示オン */
lcdc_err = R_LCDC_DispOn();
```

Special Notes:

なし

R_LCDC_DispOff ()

LCD 表示オフおよび内部昇圧/容量分割回路を停止にする関数です。

Format

```
lcdc_err_t R_LCDC_DispOff(
    uint8_t stop_select /* 表示オフのみか、回路まで停止するかを選択します */
)
```

Parameters

stop_select

仕様に応じて、LCD の表示オフのみか、内部昇圧/容量分割回路も停止するかを選択してください。

LCDC_ONLY_DISPLAY_OFF : 表示のみオフにし、回路は停止しない場合に選択してください。

LCDC_CIRCUIT_STOP : 表示をオフしさらに内部昇圧/容量分割回路も停止する場合に選択してください。

駆動電圧生成回路に内部昇圧方式もしくは容量分割方式を選択している場合に選択できます。

表示のみオフにする場合、内部昇圧/容量分割回路も停止する場合の消費電流および再表示に要する時間を下表に示します。

	駆動電圧方式	消費電流	再表示に要する時間
表示のみオフにする場合 (LCDC_ONLY_DISPLAY_OFF)	すべて同じ	回路停止時より 低減しない	不要
表示をオフ、回路を停止する場合 (LCDC_CIRCUIT_STOP)	(選択不可) 外部抵抗分割	(選択不可 : 結果は表示オフ のみと同様)	(選択不可 : 結果は表示オフのみと同様)
	内部昇圧	表示オフのみ より低減する	基準電圧のセットアップ時間 + 昇圧ウェイト時間
	容量分割	表示オフのみ より低減する	昇圧ウェイト時間

※ 基準電圧のセットアップ時間および昇圧ウェイト時間はユーザーズマニュアルハードウェア編を参照してください。

Return Values

LCDC_SUCCESS /* 問題なく処理が完了した場合 */

LCDC_ERR_INVALID_ARG /* 引数が不適切な場合 */

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

LCD の表示オフのみを行う場合、表示オフと内部昇圧/容量分割回路を停止する場合の処理を行います。

- 表示オフのみの場合 : LCD 表示をオフする処理のみを行います。
- 表示オフと内部昇圧/容量分割回路を停止する場合 : LCD 表示のオフに加えて、回路の動作を停止します。また停止後、外部抵抗分割方式に切り替えます。停止した後、回路を再動作する場合、R_LCDC_OperateCircuit 関数(ここで回路停止前の駆動電圧方式に復帰します)、R_LCDC_DispOn 関数の順で実行してください。

Example

```
lcdc_err_t    lcdc_err;  
lcdc_err = R_LCDC_DispOff(LCDC_ONLY_DISPLAY_OFF);
```

Special Notes:

なし

R_LCDC_OperateCircuit ()

内部昇圧/容量分割回路の動作許可、停止を行う関数です。

Format

```
lcdc_data_t R_LCDC_OperateCircuit (  
    uint8_t select_operation    /* 内部昇圧/容量分割回路の動作を許可、または禁止を選択 */  
)
```

Parameters

select_operation

内部昇圧/容量分割回路の動作許可/禁止を以下から選択してください。

LCDC_CIRCUIT_ENABLE : 内部昇圧/容量分割回路の動作を許可する場合に選択してください。

LCDC_CIRCUIT_DISABLE : 内部昇圧/容量分割回路の動作を禁止する場合に選択してください。

Return Values

LCDC_SUCCESS /* 問題なく処理が完了した場合 */

LCDC_ERR_INVALID_ARG /* 引数が不適切な場合 */

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

駆動電圧生成回路で内部昇圧方式もしくは容量分割方式を選択している場合、内部昇圧/容量分割回路の許可/禁止を行います。許可した後、昇圧ウェイト時間を待つ必要があります。昇圧ウェイト時間は、本関数で行うか、ユーザご自身のプログラムで行うかを選択できます。定数

” LCDC_CFG_VOLTAGE_WAIT ” で設定してください。回路の動作を禁止にする場合、本関数で回路を停止したあと、外部抵抗分割方式に変更します。再度回路を動作する際は、本関数で停止前の駆動電圧回路方式に復帰します。

Example

- R_LCDC_DispOff 関数で表示オフおよび回路を停止したあとに、再動作する場合

```
volatile lcdc_err_t lcdc_err;
```

```
lcdc_err = R_LCDC_DispOff(LCDC_CIRCUIT_STOP); /* 表示オフおよび回路を停止 */
```

```
lcdc_err = R_LCDC_OperateCircuit(LCDC_CIRCUIT_ENABLE); /* 回路を再動作 */
```

Special Notes:

なし

R_LCDC_Control ()

駆動電圧回路方式および表示データ領域の変更を行う関数です。

Format

```
lcdc_data_t R_LCDC_Control(  
    uint8_t config_pattern,      /* 何を変更するかを選択 */  
    uint8_t select_drv_gen,      /* 駆動電圧方式を変更する場合、どのように変更するかを選択 */  
    uint8_t select_display_area /* 表示データ領域を変更する場合、どのように変更するかを選択 */  
)
```

Parameters

config_pattern

駆動電圧方式、表示データ領域で変更するものを設定します。どちらも変更する場合は、OR (” | ”)を用いてください。

LCDC_FIX_DRV_GENERATOR : 駆動電圧方式を修正したい場合に選択してください。

LCDC_FIX_DISPLAY_AREA : 表示データ領域を修正したい場合に選択してください。

select_drv_gen

第一引数で” LCDC_FIX_DRV_GENERATOR” を選択した場合、以下のいずれかを選択してください。

第一引数で選択していない場合、” FIT_NO_PTR” を設定してください。

LCDC_DRV_INTERNAL_TO_CAP : 内部昇圧から容量分割に変更したい場合に選択してください。

LCDC_DRV_CAP_TO_INTERNAL : 容量分割から内部昇圧に変更したい場合に選択してください。

select_display_area

第一引数で” LCDC_FIX_DISPLAY_AREA” を選択した場合、変更後の表示データ領域を設定します。

以下のいずれかを設定してください。第一引数で選択していない場合、” FIT_NO_PTR” を設定してください

LCDC_PATTERN_A : B パターンから A パターン領域に変更 したい場合に選択してください。

LCDC_PATTERN_B : A パターンから B パターン領域に変更 したい場合に選択してください。

LCDC_BLINKING : A パターン、B パターン領域の点滅に変更したい場合に選択してください。

Return Values

LCDC_SUCCESS /* 問題なく処理が完了した場合 */

LCDC_ERR_INVALID_ARG /* 引数が不適切な場合 */

LCDC_ERR_RTC_NOT_OPERATE /* 点滅指定時に RTC が動作していない場合 */

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

駆動電圧方式、表示データ領域を変更できます。第一引数でどちらを変更するか指定します。両方の同時選択も可能です。

・ 駆動電圧方式

容量分割方式から内部昇圧方式、内部昇圧方式から容量分割方式の 2 パターンのいずれかを行います。実施手順は以下のとおりです。

- LCD 表示オフ (R_LCDC_DispOff 関数)
- 駆動電圧方式を第二引数に応じて変更
- LCD 表示オン (R_LCDC_DispOn 関数)

・ 表示データ領域を変更する場合

現在の状態から、A パターン、B パターン、点滅のいずれかに変更可能です。

点滅を使用する前には、本関数の実行前に RTC を動作する必要があります。以下の設定をしてください。

[RTC の周期(PRD)割り込み設定]

- 1) RTC に使用するクロックの発振設定：ユーザーズマニュアルの手順を参照してください。
- 2) RTC の設定：ユーザーズマニュアルの手順を参照に以下の条件を設定してください。

- ・ 周期割り込み要求：許可
- ・ 周期割り込み周期：1/2 秒ごと

Example

```
lcdc_err_t  lcdc_err;
uint8_t     cfg_pattern;
uint8_t     sel_display_area;

/* 変更内容の設定 */
cfg_pattern = LCDC_FIX_DISPLAY_AREA; /* 表示データ領域を変更 */
sel_display_area = LCDC_PATTERN_B; /* 表示データ領域を B パターンに変更 */

lcdc_err = R_LCDC_Control(cfg_pattern, 0, cfg_display_area);
```

Special Notes:

なし

R_LCDC_AdjustContrast()

コントラストの調整を行う関数です。

駆動電圧生成回路に内部昇圧方式を選択している場合に使用できます。

Format

```
int8_t R_LCDC_AdjustContrast (  
    uint8_t    select_read_adjust, /* 電圧レベルのリードか電圧レベルの変更のどちらかを選択 */  
    int8_t     updown_level        /* 変更する電圧の段階 */  
)
```

Parameters

select_read_adjust

以下のいずれかを設定してください。

LCDC_VOL_READ : 現在の電圧レベルをリードしたい場合に選択してください。

LCDC_VOL_ADJUST : 電圧レベルを変更し、コントラストを調整したい場合に選択してください。

updown_level

第一引数で“LCDC_VOL_READ”を選択した場合：“0”を設定してください。

第一引数で“LCDC_VOL_ADJUST”を選択した場合：現在の電圧レベルから何段階電圧を変更するかを“-15~15”の範囲で設定してください。たとえば3段階上げる場合は“3”を、1段階下げる場合は“-1”を設定してください。

Return Values

LCDC_VOL_LEVEL	/* 問題なく処理が完了した場合は現在の電圧レベル (変更した場合は変更した後の電圧レベル)を返します */
LCDC_ERR_OUTSIDE_VOLTAGE	/* 電圧が範囲外の場合 */
LCDC_ERR_ADJUST_PARAM	/* パラメータが不正な場合 */

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

第一引数の設定に応じて電圧レベルのリードもしくは電圧レベルの変更(コントラストの調整)を行います。

- ・電圧レベルのリードを選択した場合
戻り値で現在の電圧レベルを返します。

- ・電圧レベルの変更を選択した場合

第一引数で渡された段階数に応じて現在の電圧レベルを変更します。以下の手順で実施します。

- 選択可能な電圧の上限および下限の範囲かをチェック
- 内部昇圧/容量分割回路の動作停止
- 引数の変更する段階数に応じて電圧レベルの変更
- 内部昇圧/容量分割回路の動作許可
- 基準電圧のセットアップ時間待ち

また、電圧レベルの設定範囲は以下のとおりです。

VL1 電圧(基準電圧)の場合	VL4 電圧(1/3 バイアス法)の場合	VL4 電圧(1/4 バイアス法)の場合
1.00V	3.00V	4.00V
1.05V	3.15V	4.20V
1.10V	3.30V	4.40V
1.15V	3.45V	4.60V
1.20V	3.60V	4.80V
1.25V	3.75V	5.00V
1.30V	3.90V	5.20V
1.35V	4.05V	-
1.40V	4.20V	-
1.45V	4.35V	-
1.50V	4.50V	-
1.55V	4.65V	-
1.60V	4.80V	-
1.65V	4.95V	-
1.70V	5.10V	-
1.75V	5.25V	-

Example

```
volatile lcdc_err_t lcdc_err;
```

```
/* 1/3 バイアス法を使用し、4.20V から 3.75V に 3 段階の変更をする場合 */
```

```
lcdc_err = R_LCDC_AdjustContrast(LCDC_VOL_ADJUST, -3);
```

Special Notes:

なし

R_LCDC_Close ()

LCDC を停止する関数です。

Format

lcdc_err_t R_LCDC_Close(void)

Parameters

なし

Return Values

```
LCDC_SUCCESS          /* 問題なく処理が完了した場合 */
```

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

LCD モジュールを停止します。以下の手順で実行します。必ず、R_LCDC_DispOff 関数もしくは R_LCDC_OperateCircuit 関数で回路を停止してから本関数を実行してください。

- LCDC のレジスタおよびフラグの値を初期化
- LCD ソースクロック出力ストップ状態へ遷移
- LCDC のモジュールストップ状態の遷移
- LCDC をアンロック

Example

```
volatile lcdc_err_t lcdc_err;

lcdc_err = R_LCD_CClose();
```

Special Notes:

なし

R_LCDC_GetVersion ()

API のバージョンを返す関数です。

Format

uint32_t R_LCDC_GetVersion(void)

Parameters

なし

Return Values

バージョン番号

Properties

r_lcdc_rx_if.h にプロトタイプ宣言されています。

Description

本 API のバージョン番号を返します。

Example

```
uint32_t version;
```

```
version = R_LCDC_GetVersion();
```

Special Notes:

なし

4. 端子設定

LCDC FIT モジュールを使用するためには、マルチファンクションピンコントローラ（MPC）で周辺機能の入出力信号を端子に割り付ける（以下、端子設定と称す）必要があります。

端子設定は、R_LCDC_Open 関数を呼び出す前に行ってください。

e² studio の場合は「FIT Configurator」または「Smart Configurator」の端子設定機能を使用することができます。FIT Configurator、Smart Configurator の端子設定機能を使用すると、端子設定画面で選択したオプションに応じて、ソースファイルが出力されます。そのソースファイルで定義された関数を呼び出すことにより端子を設定できます。詳細は表 4.1 FIT コンフィグレータが出力する関数一覧を参照してください。

表 4.1 FIT コンフィグレータが出力する関数一覧

使用マイコン	出力される関数名	備考
RX113 RX23E-B	R_LCDC_PinSet()	

5. 付録

5.1 動作確認環境

本 FIT モジュールの動作確認環境を以下に示します。

表 5.1 動作確認環境 (Rev.1.01)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio V7.3.0 IAR Embedded Workbench for Renesas RX 4.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.01.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 4.8.4.201801 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.10.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev1.01
使用ボード	Renesas Starter Kit for RX113 (型名：R0K505113S000BE)

表 5.2 動作確認環境 (Rev.1.02)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio V7.7.0 IAR Embedded Workbench for Renesas RX 4.13.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.02.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.03.00.201904 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 4.13.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev1.02
使用ボード	Renesas Starter Kit for RX113 (型名：R0K505113xxxxxx)

表 5.3 動作確認環境 (Rev.2.00)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio Version 2022-10 IAR Embedded Workbench for Renesas RX 4.20.3
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.04.00 コンパイルオプション：コンパイルオプションはスマート・コンフィグレータを使用した際のデフォルト設定
	GCC for Renesas RX 8.3.0.2022.02 コンパイルオプション：コンパイルオプションはスマート・コンフィグレータを使用した際のデフォルト設定
	IAR C/C++ Compiler for Renesas RX version 4.20.3 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.00
使用ボード	Renesas Solution Starter Kit for RX23E-B (型名：RTK0ES1001C00001BJ)

表 5.4 動作確認環境 (Rev.2.01)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio Version 2025-01 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.07.00 コンパイルオプション：コンパイルオプションはスマート・コンフィグレータを使用した際のデフォルト設定
	GCC for Renesas RX 8.3.0.202411 コンパイルオプション：コンパイルオプションはスマート・コンフィグレータを使用した際のデフォルト設定
	IAR C/C++ Compiler for Renesas RX version 5.10.1 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのリビジョン	Rev2.01
使用ボード	-

5.2 トラブルシューティング

- (1) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「Could not open source file "platform.h"」エラーが発生します。

A : FIT モジュールがプロジェクトに正しく追加されていない可能性があります。プロジェクトへの追加方法をご確認ください。

- CS+を使用している場合
アプリケーションノート RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」
- e² studio を使用している場合
アプリケーションノート RX ファミリ e² studio に組み込む方法 Firmware Integration Technology (R01AN1723)」

また、本 FIT モジュールを使用する場合、ボードサポートパッケージ FIT モジュール(BSP モジュール)もプロジェクトに追加する必要があります。BSP モジュールの追加方法は、アプリケーションノート「ボードサポートパッケージモジュール(R01AN1685)」を参照してください。

- (2) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「This MCU is not supported by the current r_lcdc_rx module.」エラーが発生します。

A : 追加した FIT モジュールがユーザプロジェクトのターゲットデバイスに対応していない可能性があります。追加した FIT モジュールの対象デバイスを確認してください。

- (3) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「Parameter error in configures file.」エラーが発生します。

A : “r_lcdc_rx_config.h” ファイルの設定値が間違っている可能性があります。“r_lcdc_rx_config.h” ファイルを確認して正しい値を設定してください。詳細は「2.7 コンパイル時の設定」を参照してください。

- (4) Q : LCD パネルに文字が表示されません。

A : 正しく端子設定が行われていない可能性があります。本 FIT モジュールを使用する場合は端子設定が必要です。詳細は「4 端子設定」を参照してください。

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

（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。）

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

RX ファミリ CC-RX コンパイラ ユーザーズマニュアル（R20UT3248）

（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。）

テクニカルアップデートの対応について

本モジュールは以下のテクニカルアップデートの内容を反映しています。

対応しているテクニカルアップデートはありません。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Dec.01.14	-	初版作成
1.01	May.20.19	—	以下のコンパイラに対応 ・ GCC for Renesas RX ・ IAR C/C++ Compiler for Renesas RX
		1	対象コンパイラを追加
		9	2.7 コードサイズ を追加
		12	2.10 for 文、while 文、do while 文について を追加
		26	「R_LCDC_GetVersion()」章を更新
		27	「4.端子設定」章を追加
		27	「5.1 動作確認環境」章を追加
		プログラム	機能関連 Smart Configurator での GUI によるコンフィグオプション設定機能に対応 ■内容 GUI によるコンフィグオプション設定機能に対応するため、設定ファイルを追加。
		プログラム	R_LCDC_GetVersion 関数のインライン展開を削除
		—	API 関数のコメントを Doxygen スタイルに変更
1.02	Jun.10.20	1	関連ドキュメントの、R01AN1833 を削除
		9	2.7 コードサイズ を変更
		11	2.9 「FIT モジュールの追加方法」を更新
		13~26	API 説明ページの、「Reentrant」項目を削除
		28	表 5.2 動作確認環境 (Rev.1.02)を追加
		プログラム	以下を修正しました。 ■対象デバイス 全デバイス ■内容 複数の周辺機能から同時にアクセスされる可能性があるレジスタがあり、そのレジスタへの書き込みのアトミック性が確保できるように処理を変更。
		all	RX23E-B に対応
		6	LCDC_CFG_PARAM_CHECKING_ENABLE の初期値を変更
2.00	May.31.23	9	2.7 コードサイズ を変更
		10	戻り値の定義変更
		16	「R_LCDC_Modify()」章の data 引数の内容を更新
		20	「R_LCDC_OperateCircuit()」章の Format 内容の変更
		22	「R_LCDC_AdjustContrast()」章の戻り値見直し
		プログラム	■概要 モジュール内のフォルダ構成を変更。 いくつかの問題を修正。 ■内容 targets フォルダを削除し、ソースファイルを共通化。 R_LCDC_AdjustContrast 関数の戻り値を変更 R_LCDC_Control 関数のパラメータチェック方法、レジスタ設定方法の変更

Rev.	発行日	改訂内容	
		ページ	ポイント
2.00	May.31.23	プログラム	lcdc_delay 関数の処理を変更
			LCDM1 レジスタに対して SCOC ビットが"1"の状態を設定する可能性がある処理を修正
			lcdc_close 関数で LCD カウントソースが停止していなかった問題を修正
			LCDC_CFG_PARAM_CHECKING_ENABLE の初期値を変更
			LCDC_SEL_CIRCUIT_ENABLE, LCDC_SEL_CIRCUIT_DISABLE 定義の削除
2.01	Mar.20.25	29	表 5.4 動作確認環境 (Rev.2.01)を追加
		プログラム	プログラムの免責事項を変更

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、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。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）、交通管制（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。

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、これに限りません。）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いたします。
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。