

## RL78/G16

### リアルタイム・クロック 2

#### 要旨

本アプリケーションノートでは、リアルタイム・クロック 2 (RTC2) の定周期割り込み機能、およびアラーム割り込み機能の使用例を示します。

定周期割り込み機能を利用して出力ポートの反転出力と LCD での時刻表示を行います。アラーム割り込み機能を利用して時計の設定日時から 5 秒後にアラーム割り込みを発生させます。

また、リセット期間中でも RTC2動作が継続していることをLCDの時刻表示で確認できます。

#### 動作確認デバイス

RL78/G16

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。

## 目次

1. 仕様 .....	3
1.1 仕様概要 .....	3
1.2 動作詳細 .....	4
2. 動作確認条件 .....	5
3. ハードウェア説明 .....	6
3.1 ハードウェア構成例 .....	6
3.2 使用端子一覧 .....	6
4. ソフトウェア説明 .....	7
4.1 オプション・バイトの設定一覧 .....	7
4.2 定数一覧 .....	7
4.3 変数一覧 .....	9
4.4 関数一覧 .....	9
4.5 関数仕様 .....	10
4.6 フローチャート .....	14
4.6.1 メイン処理 .....	14
4.6.2 RTC2 初期化処理 (ユーザ定義箇所) .....	15
4.6.3 アラーム割込み時の処理 .....	15
4.6.4 定周期割込み時の処理 .....	16
4.6.5 現在時刻初期設定 .....	16
4.6.6 アラーム時刻初期設定 .....	17
4.6.7 定周期割込みフラグ確認処理 .....	17
4.6.8 定周期割込みフラグクリア処理 .....	18
4.6.9 現在時刻表示処理 .....	19
4.6.10 BCD から文字への変換処理 .....	20
4.6.11 曜日から文字への変換処理 .....	21
4.6.12 IICA0 送信時のコールバック処理 .....	21
4.6.13 IICA0 エラー発生時のコールバック処理 .....	22
4.6.14 LCD モジュール初期化 .....	22
4.6.15 LCD モジュール 表示消去処理 .....	23
4.6.16 LCD モジュール 文字列送信処理 .....	23
4.6.17 LCD モジュール コマンド送信処理 .....	24
4.6.18 LCD モジュール データ送信処理 .....	24
4.6.19 LCD モジュール 通信終了フラグ設定 .....	25
4.6.20 LCD モジュール 通信終了待ち処理 .....	25
5. サンプルコード .....	26
6. 参考ドキュメント .....	26
改訂記録 .....	27

## 1. 仕様

### 1.1 仕様概要

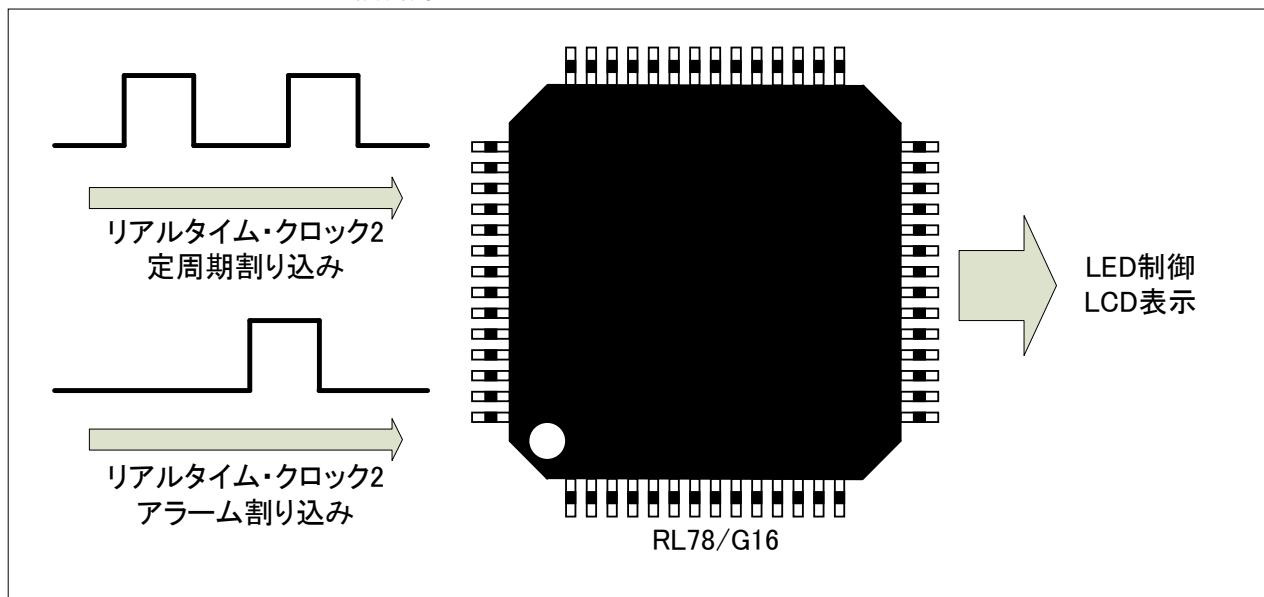
定周期割り込み機能を利用して出力ポートの反転出力とLCDでの時刻表示を行います。アラーム割り込み機能を利用して時計の設定日時から5秒後にアラーム割り込みを発生させます。

表 1-1に使用する周辺機能と用途を示し、図 1-1にサンプルコードの動作概要を示します。

表 1-1 使用する周辺機能と用途

周辺機能	用途
リアルタイム・クロック 2	RTC2 の割り込み (INTRTC)
P12	定周期割り込み処理でポート出力に設定 (反転出力)
P15	アラーム割り込み処理でポート出力に設定 (初期値 : ハイ・レベル出力)
シリアル・インターフェース IICA0 P60/SCLA0、P61/SDAA0	LCD モジュールとの I2C 通信
RESET	外部リセット入力
P121/XT1 P122/XT2	RTC2 の動作クロック

図 1-1 サンプルコードの動作概要



## 1.2 動作詳細

本アプリケーションノートでは、RTC2の時刻を「2023/1/1 (Fri) 15 : 59 : 55」に設定し、アラーム時刻を「毎日16:00:00」に設定します。また、次の割り込み処理を行います。

- 定周期割り込み処理でP12 の出力反転とLCDでの時刻表示
- アラーム割り込み処理でP15 のロウ・レベル出力 (LED点灯)

(1) リアルタイム・クロック 2 (RTC2) の初期設定を行います。

- リアルタイム・クロック 2 の動作クロックにサブシステム・クロック ( $f_{sub}$ ) を選択
- 時刻表現は 24 時間制に設定
- RTC1HZ 端子の出力を禁止
- 定周期割り込み許可設定、割り込み周期 0.5s を使用
- アラーム割り込み許可
- INTRTC 割り込みを許可

(2) 入出力ポートを設定します。

- P12 を出力ポートに設定 (初期値：ロウ・レベル、LED 点灯状態)。定周期割り込み処理用。
- P15 を出力ポートに設定 (初期値：ハイ・レベル、LED 消灯状態)。アラーム割り込み処理用。

(3) シリアル・インターフェース IICA0 の初期設定を行います。

- IICA0 を使用 (P60 を SCLA0 に、P61 を SDAA0 に設定)
- 自局アドレスを 0x10 に設定
- 動作モードを標準に設定
- 転送クロックを 80000 bps に設定
- INTIICA0 割り込みを許可

(4) LCD モジュールの初期設定を行います。

- 8 ビット・バスモード、2 ライン表示、フォントタイプ 5x8 ドットに設定
- ディスプレイ表示オン、カーソル表示オフ、カーソル点滅オフに設定
- カーソルのシフト方向を右方向に設定

(5) RTC2 の割込みに応じて、LED 制御、LCD モジュールとの通信を行います。

注意 1. デバイス使用上の注意事項については、RL78/G16 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。

注意 2. RTC2 の初期設定は、データ保持電源電圧による内部リセットがあった場合のみ実行します。これを実現するため、ソースコードではスマート・コンフィグレータが生成するコードを一部書き換えており、スマート・コンフィグレータで再度コード生成をすると、この書き換えが無効となることに注意してください。また、RTC2 の時刻及び、アラーム時刻の初期設定値は、r\_cg\_userdefine.h 内にて定義した定数を使用しています。

## 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、以下の条件で動作を確認しています。

表 2-1 動作確認条件

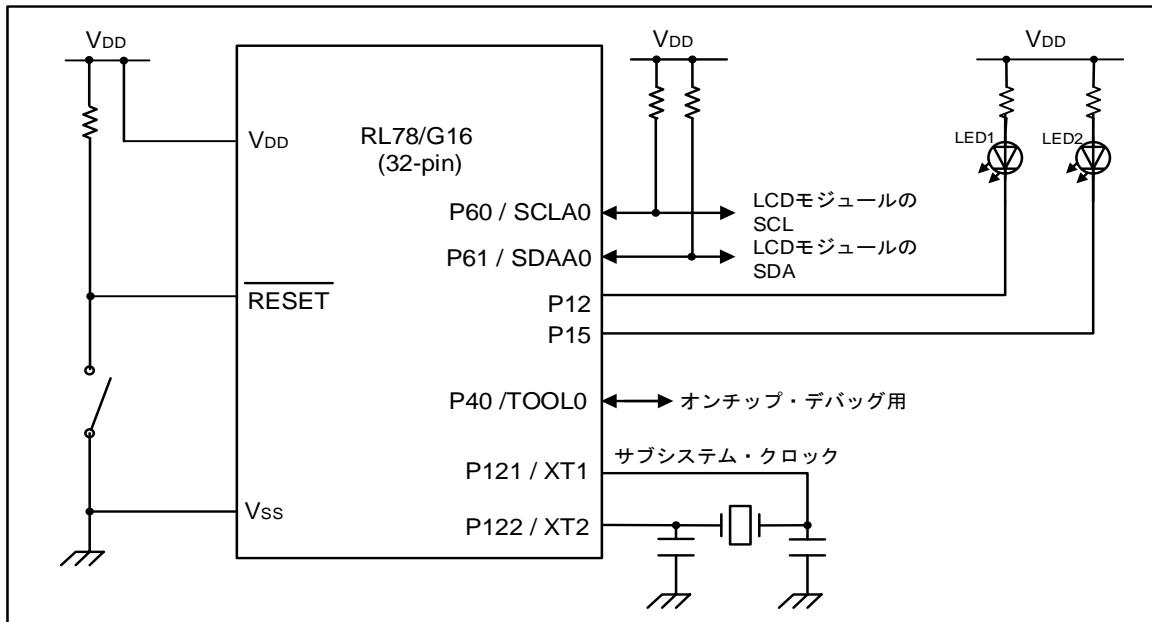
周辺機能	用途
使用マイコン	RL78/G16 (R5F121BCAFP)
使用ボード	RL78/G16 Fast Prototyping Board (RTK5RLG160C00000BJ)
動作周波数	高速オンチップ・オシレータ・クロック ( $f_{IH}$ ) : 16MHz
動作電圧	5.0V (2.4V~5.5V で動作可能) SPOR 検出電圧 立ち上がり時 TYP. 2.57V (2.44 V ~ 2.68 V) 立ち下がり時 TYP. 2.52V (2.40 V ~ 2.62 V)
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ for CC V8.10.00
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.12.01
統合開発環境 (e2studio)	ルネサス エレクトロニクス製 e2studio V2023-10 (23.10.0)
C コンパイラ (e2studio)	ルネサス エレクトロニクス製 CC-RL V1.12.01
統合開発環境 (IAR)	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 V5.10.1
C コンパイラ (IAR)	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V5.10.1
スマート・コンフィグレータ (SC)	ルネサス エレクトロニクス製 V1.8.0
ボードサポートパッケージ (BSP)	ルネサス エレクトロニクス製 V1.60
LCD モジュール	ACM1602NI-FLW-FBW-M01

### 3. ハードウェア説明

#### 3.1 ハードウェア構成例

図 3-1 に本アプリケーションノートで使用するハードウェア構成例を示します。

図 3-1 ハードウェア構成



注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して  $V_{DD}$  又は  $V_{SS}$  に接続して下さい）。

注意 2.  $V_{DD}$  は SPOR にて設定したリセット解除電圧( $V_{SPOR}$ ) 以上にしてください。

#### 3.2 使用端子一覧

表 3-1 に使用端子と機能を示します。

表 3-1 使用端子と機能

端子名	入出力	内容
P12	出力	LED1 の点灯制御
P15	出力	LED2 の点灯制御
P60 / SCLA0、P61 / SDAA0	入出力	LCD モジュールとの I2C 通信
RESET	入力	外部リセット入力
P121 / XT1、P122 / XT2	入力	RTC2 の動作クロック

注意 本アプリケーションノートは、使用端子のみを端子処理しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

## 4. ソフトウェア説明

### 4.1 オプション・バイトの設定一覧

表 4-1にオプション・バイト設定を示します。

表 4-1 オプション・バイト設定

アドレス	設定値	内容
000C0H	11101111B	ウォッチドッグ・タイマ動作停止 (リセット解除後、カウント停止)
000C1H	11111011B	SPOR 検出電圧： 立ち上がり時TYP. 2.57V (2.44 V ~ 2.68 V) 立ち下がり時 TYP. 2.52V (2.40 V ~ 2.62 V)
000C2H	11111001B	高速オンチップ・オシレータ・クロック :16MHz
000C3H	10000101B	オンチップ・デバッグ許可

### 4.2 定数一覧

表 4-2 にサンプルコードで使用する定数を示します。

表 4-2 サンプルコードで使用する定数 (1/2)

定数名	設定値	内容
_0xA0_LCM_SLAVE_ADDR	0xA0	LCD モジュールのスレーブ アドレス
_0x00_LCM_SLAVE_ADDR_RW_LOW	0x00	LCD モジュールへの書き込みフラグ
_0x80_LCM_CONTROL_BYTE_RS_HIGH	0x80	LCD モジュールへのデータ転送フラグ
_0x00_LCM_CONTROL_BYTE_RS_LOW	0x00	LCD モジュールへのコマンド転送フラグ
_0x01_LCM_COMMAND_CLEAR_DISPLAY	0x01	ディスプレイをクリアするフラグ
_0x04_LCM_COMMAND_ENTRY_MODE_SET	0x04	表示位置設定フラグ
_0x02_LCM_COMMAND_ENTRY_MODE_SET_ID_HIGH	0x02	表示位置を右シフト (インクリメント)
_0x00_LCM_COMMAND_ENTRY_MODE_SET_S_LOW	0x00	表示シフトオフフラグ
_0x08_LCM_COMMAND_DISPLAY_ONOFF	0x08	ディスプレイ表示有効フラグ
_0x04_LCM_COMMAND_DISPLAY_ONOFF_D_HIGH	0x04	ディスプレイ表示オンフラグ
_0x00_LCM_COMMAND_DISPLAY_ONOFF_C_LOW	0x00	カーソル表示無効フラグ
_0x00_LCM_COMMAND_DISPLAY_ONOFF_B_LOW	0x00	カーソル点滅無効フラグ
_0x20_LCM_COMMAND_FUNCTION_SET	0x20	コマンド機能有効フラグ
_0x10_LCM_COMMAND_FUNCTION_SET_DL_HIGH	0x10	LCD モジュールへの転送ビット単位 : 8-bit
_0x08_LCM_COMMAND_FUNCTION_SET_N_HIGH	0x08	ディスプレイ表示 : 2 行
_0x00_LCM_COMMAND_FUNCTION_SET_F_LOW	0x00	フォント表示 : 5x8 ドット
_0x80_LCM_COMMAND_SET_DDRAM_ADDRESS	0x80	フォント表示位置設定

表 4-3 サンプルコードで使用する定数 (2/2)

定数名	設定 値	内容
LCM_COMMAND_EXEC_WAIT	26600	LCD モジュール コマンド実行待ち時間 1.5ms (16MHz 動作時)
LCM_CONFIG_FUNCTION_SET_PARAMS	0x18	ファンクション セット コマンド パラメータ _0x10_LCM_COMMAND_FUNCTION_SET_DL_HIGH   _0x08_LCM_COMMAND_FUNCTION_SET_N_HIGH   _0x00_LCM_COMMAND_FUNCTION_SET_F_LOW
LCM_CONFIG_ENTRY_MODE_SET_PARAMS	0x20	エントリー モード セット コマンド パラメータ _0x02_LCM_COMMAND_ENTRY_MODE_SET_ID_HIGH   _0x00_LCM_COMMAND_ENTRY_MODE_SET_S_LOW
LCM_CONFIG_DISPLAY_ONOFF_PARAMS	0x40	ディスプレイ On/Off 制御 コマンドパラメータ _0x04_LCM_COMMAND_DISPLAY_ONOFF_D_HIGH   _0x00_LCM_COMMAND_DISPLAY_ONOFF_C_LOW   _0x00_LCM_COMMAND_DISPLAY_ONOFF_B_LOW
LCM_CONFIG_MAX_CHAR_PER_LINE	16	1 行の最大文字数
LCM_CONFIG_WAIT_COUNT	13	IIIA0 ウエイト カウント
LCM_POSITION_TOP	0x80	LCD モジュール表示ライン 上段
LCM_POSITION_BOTTOM	0xC0	LCD モジュール表示ライン 下段
R_RTC2_INIT_SEC	0x55	現在時刻の初期値 (秒)
R_RTC2_INIT_MIN	0x59	現在時刻の初期値 (分)
R_RTC2_INIT_HOUR	0x15	現在時刻の初期値 (時)
R_RTC2_INIT_WEEK	0x06	現在時刻の初期値 (曜日)
R_RTC2_INIT_DAY	0x01	現在時刻の初期値 (日)
R_RTC2_INIT_MONTH	0x01	現在時刻の初期値 (月)
R_RTC2_INIT_YEAR	0x23	現在時刻の初期値 (年)
R_RTC2_ALARM_MIN	0x00	アラーム発生時刻の設定値 (分)
R_RTC2_ALARM_HOUR	0x16	アラーム発生時刻の設定値 (時)
R_RTC2_ALARM_WEEK	0xFF	アラーム発生時刻の設定値 (曜日)
R_INTERRUPT_OFF	0	割込みフラグクリア状態
R_INTERRUPT_ON	1	割込みフラグオン状態

### 4.3 変数一覧

表 4-4 にグローバル変数を示します。

表 4-4 グローバル変数

型	変数名	内容	使用関数
uint8_t	g_RTC_constperiod	定周期割り込み発生を示す フラグ	main、 r_Config_RTC_Create_UserInit、 r_Config_RTC_callback_constperiod、 r_RTC_is_constperiod_flag_on、 r_RTC_clear_constperiod_flag
uint8_t	g_LCM_is_sendend	LCD モジュールとの I2C 通 信完了を示すフラグ	r_LCM_init、 r_LCM_turn_sendend_on、 r_LCM_wait_sendend

### 4.4 関数一覧

表 4-5 に関数を示します。

表 4-5 関数

関数名	概要
R_Config_RTC_Create_UserInit()	ユーザ指定の RTC2 初期化処理
r_Config_RTC_callback_alarm()	アラーム割り込み時の処理
r_Config_RTC_callback_constperiod()	定周期割り込み時の処理
r_RTC_init_current_time()	現在時刻の初期設定
r_RTC_init_alarm_time()	アラーム時刻の初期設定
r_RTC_is_constperiod_flag_on()	定周期割り込みフラグの確認
r_RTC_clear_constperiod_flag()	定周期割り込みフラグをクリア
r_RTC_display_current_time()	現在時刻をLCDに表示
convert_BCD_to_2chars()	2桁の BCD を文字 2 つに変換
convert_week_to_3chars()	曜日コードを文字 3 つに変換
r_Config_IICA0_callback_master_sendend()	IICA0 送信完了時のコールバック処理
r_Config_IICA0_callback_master_error()	IICA0 エラー発生時のコールバック処理
r_LCM_init()	LCD モジュール初期化 処理
r_LCM_clear()	LCD モジュール 表示消去処理
r_LCM_send_string()	LCD モジュール 文字列送信処理
r_LCM_send_command()	LCD モジュール コマンド送信処理
r_LCM_send_data()	LCD モジュール データ送信処理
r_LCM_turn_sendend_on()	LCD モジュール 通信終了フラグ設定
r_LCM_wait_sendend()	LCD モジュール 通信終了待ち処理

## 4.5 関数仕様

サンプルコードの関数仕様を示します。

### R\_Config\_RTC\_Create\_UserInit()

概 要	ユーザー指定のRTC2初期化処理
ヘッダ	r_cg_userdefine.h
宣 言	void R_Config_RTC_Create_UserInit(void);
説 明	RTC2 開始前に必要な初期化のうち、ユーザ指定の処理を行います。
引 数	なし
リターン値	なし

### r\_Config\_RTC\_callback\_alarm()

概 要	アラーム割込み時の処理
ヘッダ	r_cg_makrodriver.h
宣 言	static void r_Config_RTC_callback_alarm(void);
説 明	RTC2 のアラーム割込み発生時に呼ばれるコールバック関数です。
引 数	なし
リターン値	なし

### r\_Config\_RTC\_callback\_constperiod()

概 要	定周期割込み時の処理
ヘッダ	r_cg_makrodriver.h、r_cg_userdefine.h
宣 言	static void r_Config_RTC_callback_constperiod(void);
説 明	RTC2 の定周期割込み発生時に呼ばれるコールバック関数です。
引 数	なし
リターン値	なし

### r\_RTC\_init\_current\_time()

概 要	現在時刻の初期設定
ヘッダ	r_cg_makrodriver.h、r_cg_userdefine.h
宣 言	void r_RTC_init_current_time(void);
説 明	RTC2 のレジスタに現在時刻を設定します。
引 数	なし
リターン値	なし

### r\_RTC\_init\_alarm\_time()

概 要	アラーム時刻の初期設定
ヘッダ	r_cg_makrodriver.h、r_cg_userdefine.h、Config_RTC.h
宣 言	void r_RTC_init_alarm_time(void);
説 明	RTC2 のレジスタにアラーム時刻を設定します。
引 数	なし
リターン値	なし

---

**r\_RTC\_is\_constperiod\_flag\_on()**

---

概 要 定周期割り込みフラグの確認  
 ヘッダ r\_cg\_userdefine.h  
 宣 言 uint8\_t r\_RTC\_is\_constperiod\_flag\_on(void);  
 説 明 g\_RTC\_constperiod を確認し、結果を返します。  
 引 数 なし  
 リターン値 1: g\_RTC\_constperiod が R\_INTERRUPT\_ON  
           0: g\_RTC\_constperiod が R\_INTERRUPT\_OFF

---



---

**r\_RTC\_clear\_constperiod\_flag()**

---

概 要 定周期割り込みフラグをクリア  
 ヘッダ r\_cg\_userdefine.h  
 宣 言 void r\_RTC\_clear\_constperiod\_flag(void);  
 説 明 g\_RTC\_constperiod をクリアします (R\_INTERRUPT\_OFF に設定)。  
 引 数 なし  
 リターン値 なし

---



---

**r\_RTC\_display\_current\_time()**

---

概 要 現在時刻をLCDに表示  
 ヘッダ Config\_RTC.h  
 宣 言 void r\_RTC\_display\_current\_time(void);  
 説 明 現在時刻を LCD に表示します。  
 引 数 なし  
 リターン値 なし

---



---

**convert\_BCD\_to\_2chars()**

---

概 要 BCD2桁の数値を文字2つに変換  
 ヘッダ r\_cg\_macrodriver.h  
 宣 言 static void convert\_BCD\_to\_2chars(uint8\_t bcd, uint8\_t \* const str);  
 説 明 BCD2桁の数値を文字2つに変換します。  
 引 数 uint8\_t bcd : 変換元となる2桁のBCD値。0x00~0x99でなければならない。  
       uint8\_t \*str : 変換した文字が格納される領域。末尾に'¥0'は付加されない。  
 リターン値 なし

---



---

**convert\_week\_to\_3chars()**

---

概 要 曜日を示す数値を文字3つに変換  
 ヘッダ r\_cg\_macrodriver.h  
 宣 言 static void convert\_week\_to\_3chars(uint8\_t week, uint8\_t \* const str);  
 説 明 曜日を示す数値を文字3つに変換します。以下のような結果となります。

元の数値	0	1	2	3	4	5	6
変換結果	Sun	Mon	Tue	Wed	Thu	Fri	Sat

引 数 uint8\_t week : 変換元となる曜日を示す値。0~6でなければならない。  
       uint8\_t \*str : 変換した文字が格納される領域。末尾に'¥0'は付加されない。  
 リターン値 なし

---

---

**r\_Config\_IICA0\_callback\_master\_sendend()**

---

概 要	IICA0送信完了時のコールバック処理
ヘッダ	r_cg_macrodriver.h、Config_IICA0.h、LCM_driver.h
宣 言	static void r_Config_IICA0_callback_master_receiveend(void);
説 明	IICA0 の送信完了時に呼ばれるコールバック関数です。 ストップ・コンディションを生成後、LCD モジュール 通信終了フラグ設定関数を呼びます。
引 数	なし
リターン値	なし

---



---

**r\_Config\_IICA0\_callback\_master\_error()**

---

概 要	IICA0送信エラー発生時のコールバック処理
ヘッダ	r_cg_mmacrodriver.h、Config_IICA0.h、LCM_driver.h
宣 言	static void r_Config_IICA0_callback_master_error(MD_STATUS flag);
説 明	IICA0 エラー発生時に呼ばれるコールバック関数です。 LCD モジュール 通信終了フラグ設定関数を呼びます。
引 数	MD_STATUS flag : エラータイプ
リターン値	なし

---



---

**r\_LCM\_init()**

---

概 要	LCDモジュール初期化
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	void r_LCM_init(void);
説 明	LCD モジュールを初期化します。
引 数	なし
リターン値	なし

---



---

**r\_LCM\_clear()**

---

概 要	LCDモジュール 表示消去処理
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	void r_LCM_clear(void);
説 明	LCD モジュールに表示消去処理のコマンドを送信します。
引 数	なし
リターン値	なし

---



---

**r\_LCM\_send\_string()**

---

概 要	LCDモジュール 文字列送信処理
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	void r_LCM_send_string(uint8_t * const str, lcm_position_t pos);
説 明	LCD モジュールに str で渡された文字列を表示します。 表示させるラインは pos で指定します。 uint8_t * const str : 表示させる文字列
引 数	lcm_position_t pos : LCM_POSITION_TOP で上段に表示 LCM_POSITION_BOTTOM で下段に表示
リターン値	なし

---

---

**r\_LCM\_send\_command()**

---

概 要	LCDモジュール コマンド送信処理
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	void r_LCM_send_command(uint8_t command);
説 明	LCD モジュールに command で渡されたコマンドを送信します。
引 数	uint8_t command : LCD モジュールへ送信するコマンド
リターン値	なし

---

**r\_LCM\_send\_data()**

---

概 要	LCDモジュール データ送信処理
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	void r_LCM_send_data(uint8_t data);
説 明	LCD モジュールに data で渡されたデータを送信します。
引 数	uint8_t data : LCD モジュールへ送信するデータ
リターン値	なし

---

**r\_LCM\_turn\_sendend\_on()**

---

概 要	LCDモジュール 通信終了フラグ設定
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	void r_LCM_turn_sendend_on(void);
説 明	g_LCM_is_sendend に LCD モジュールとの IIC 通信終了フラグを設定します。
引 数	なし
リターン値	なし

---

**r\_LCM\_wait\_sendend()**

---

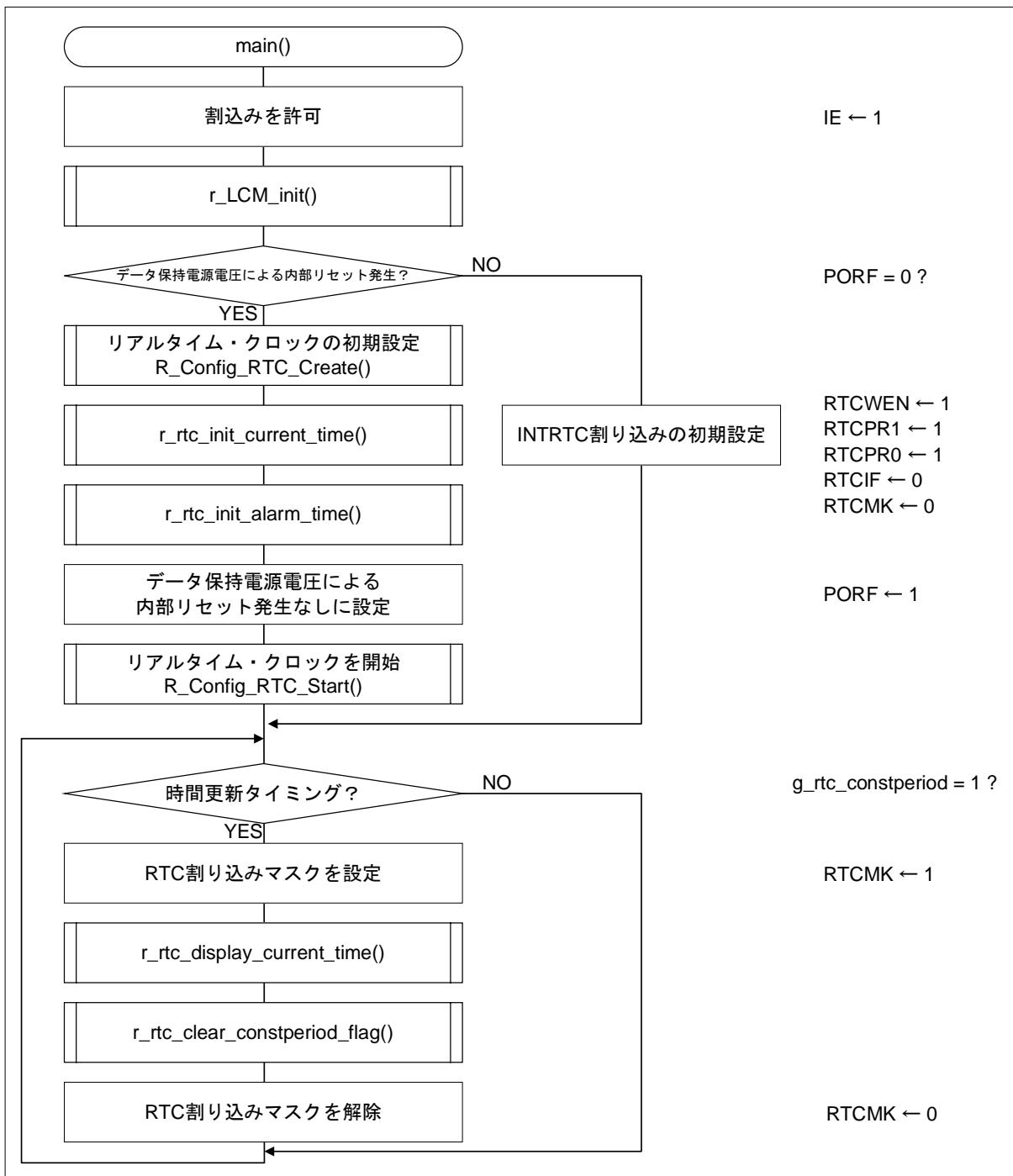
概 要	LCDモジュール 通信終了待ち処理
ヘッダ	LCM_driver.h、Config_IICA0.h
宣 言	static void r_LCM_wait_sendend(void);
説 明	LCD モジュールとの IIC 通信が終了するまで待ち、コマンド実行ウェイト時間 (5ms) だけウェイトを実行します。
引 数	なし
リターン値	なし

## 4.6 フローチャート

### 4.6.1 メイン処理

図 4-1にメイン処理のフローチャートを示します。

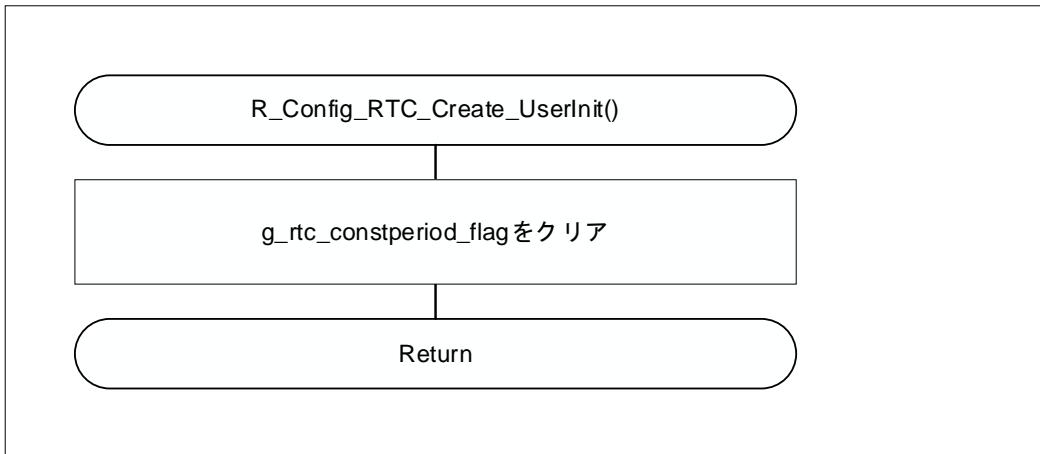
図 4-1 メイン処理



#### 4.6.2 RTC2 初期化処理 (ユーザ定義箇所)

図 4-2 にユーザ定義の RTC2 初期化処理のフローチャートを示します。

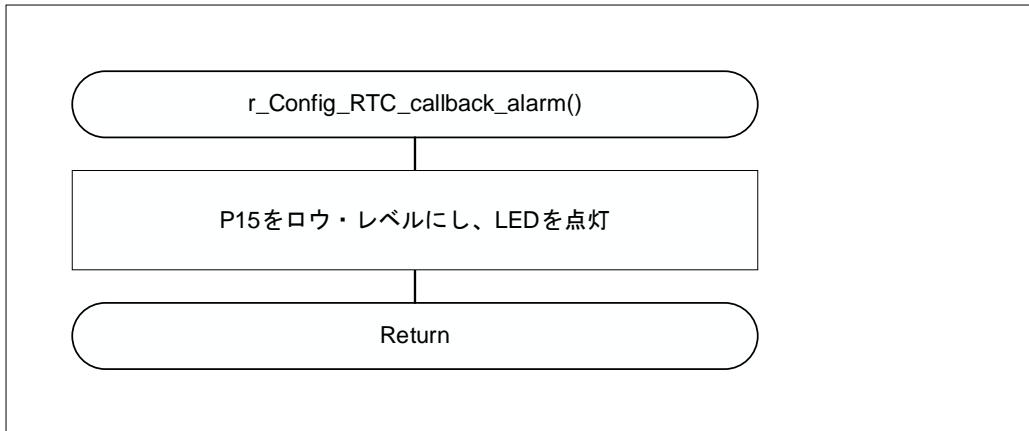
図 4-2 RTC2 初期化処理 (ユーザ定義箇所)



#### 4.6.3 アラーム割込み時の処理

図 4-3 にアラーム割込み時の処理のフローチャートを示します。

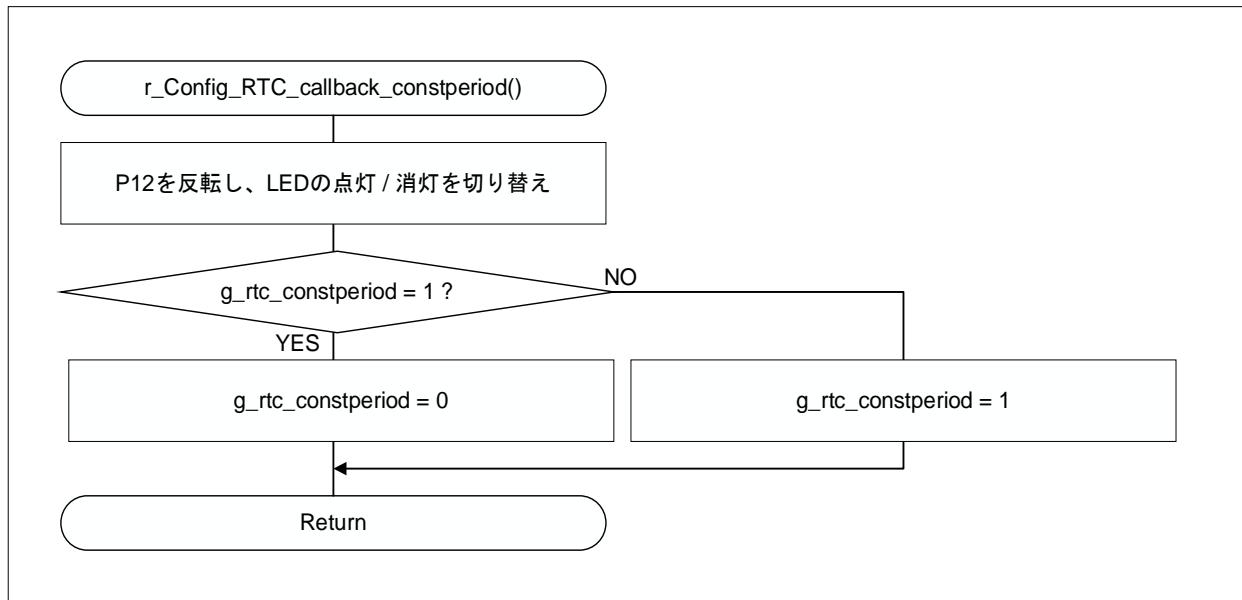
図 4-3 アラーム割込み時の処理



#### 4.6.4 定周期割込み時の処理

図 4-4 に定周期割込み時の処理のフローチャートを示します。

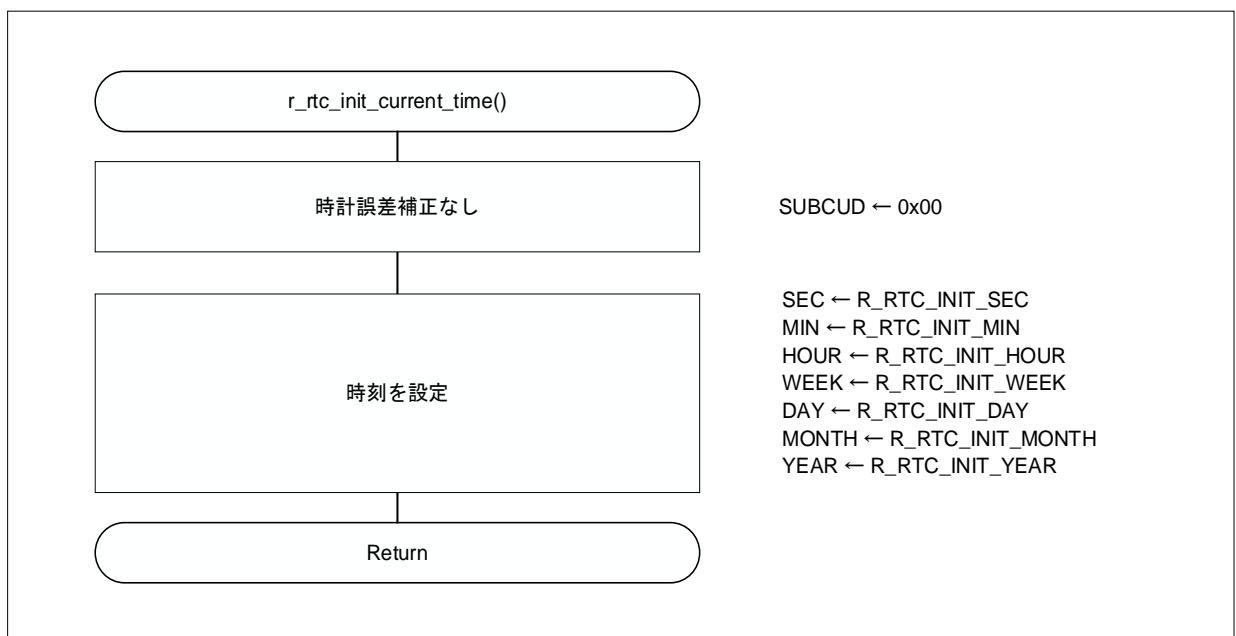
図 4-4 定周期割込み時の処理



#### 4.6.5 現在時刻初期設定

図 4-5 に現在時刻初期設定のフローチャートを示します。

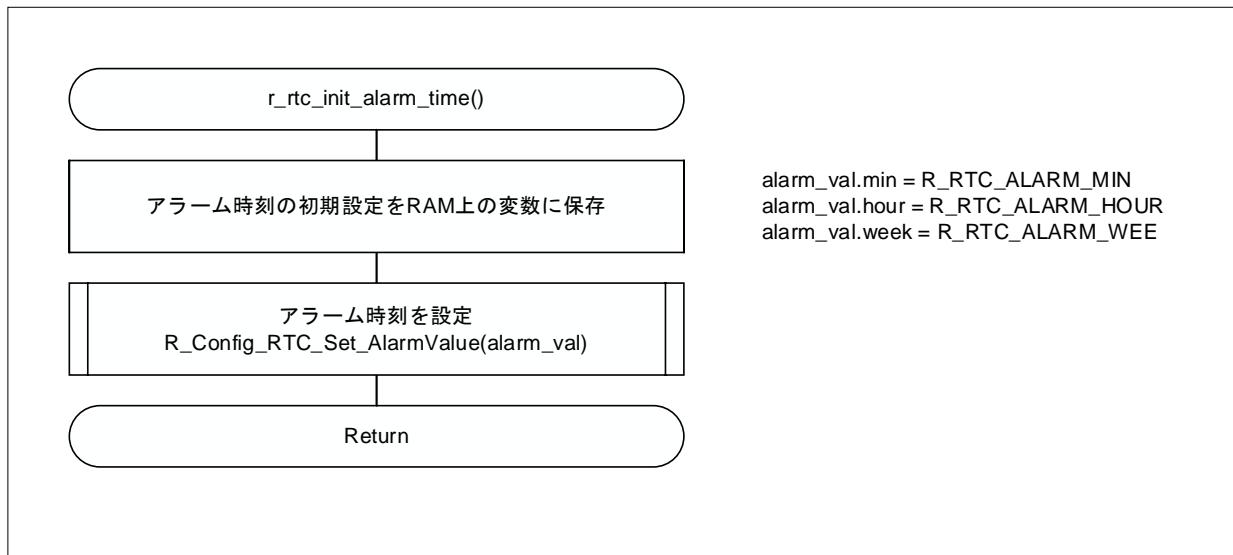
図 4-5 現在時刻初期設定



#### 4.6.6 アラーム時刻初期設定

図 4-6 にアラーム時刻初期設定のフローチャートを示します。

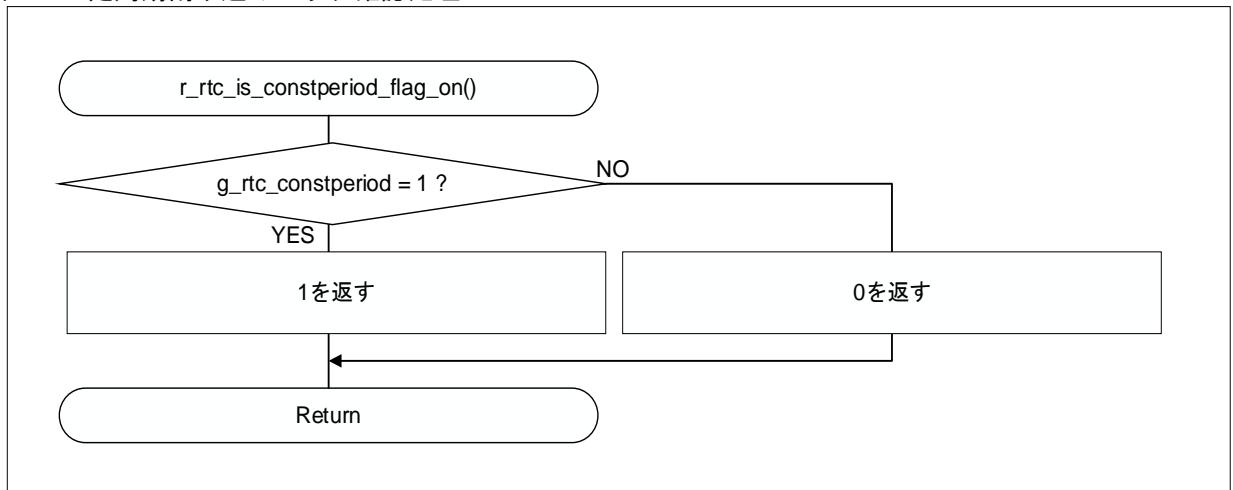
図 4-6 アラーム時刻初期設定



#### 4.6.7 定周期割込みフラグ確認処理

図 4-7 に定周期割込みフラグ確認処理のフローチャートを示します。

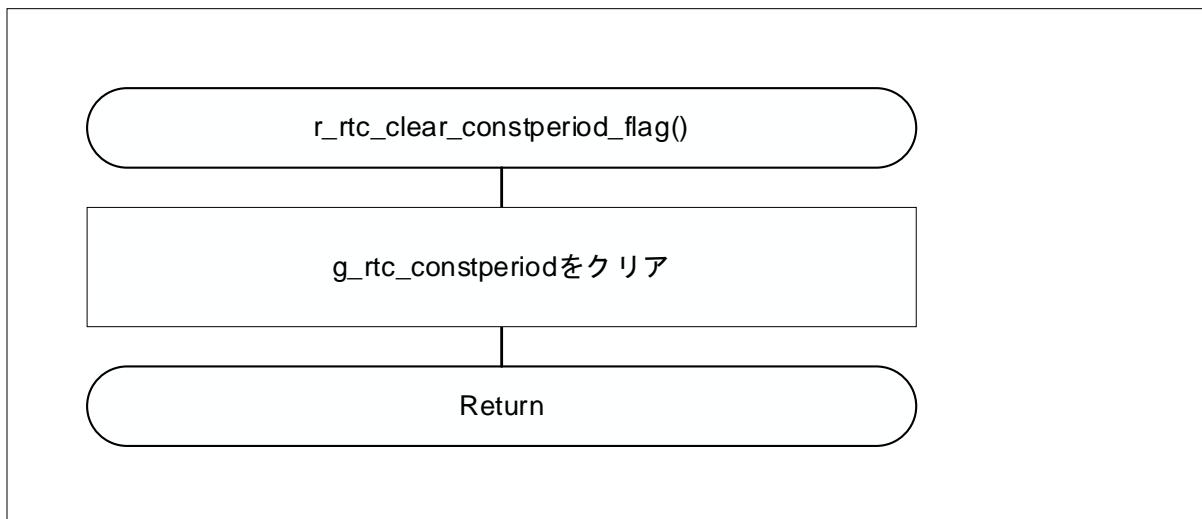
図 4-7 定周期割り込みフラグ確認処理



#### 4.6.8 定周期割込みフラグクリア処理

図 4-8 に定周期割込みフラグクリア処理のフローチャートを示します。

図 4-8 定周期割り込みフラグクリア処理



#### 4.6.9 現在時刻表示処理

図 4-9、図 4-10 に現在時刻表示処理のフローチャートを示します。

図 4-9 現在時刻表示処理 (1/2)

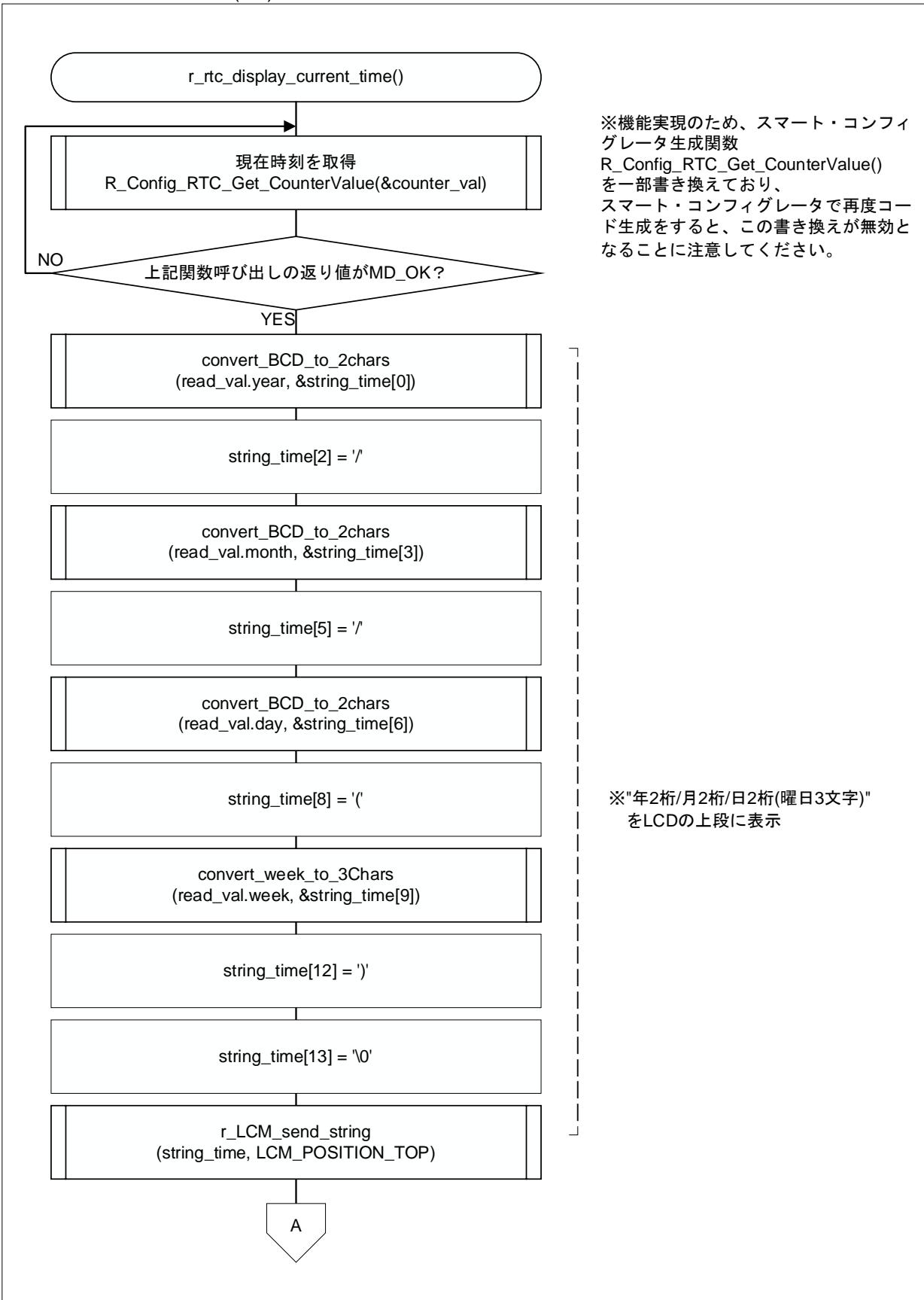
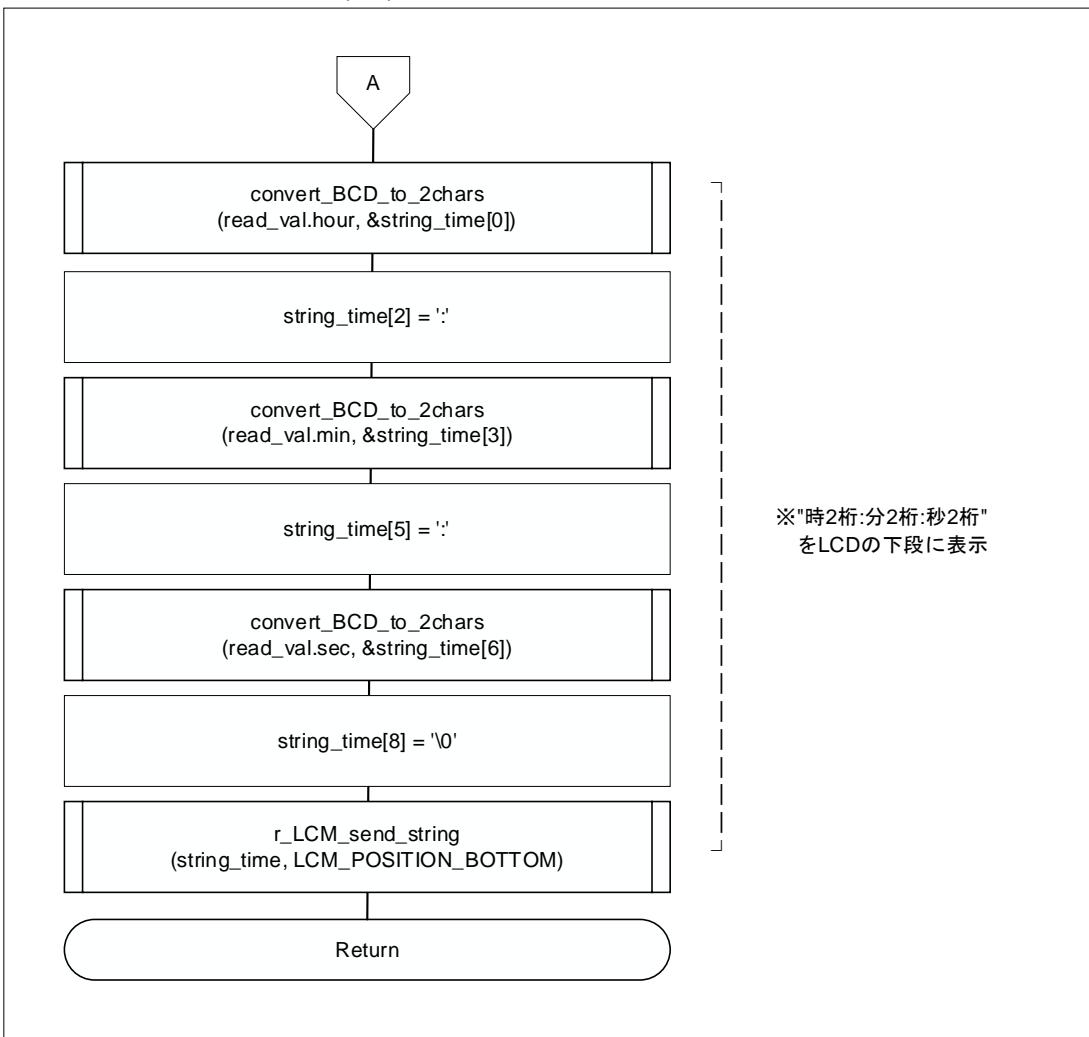


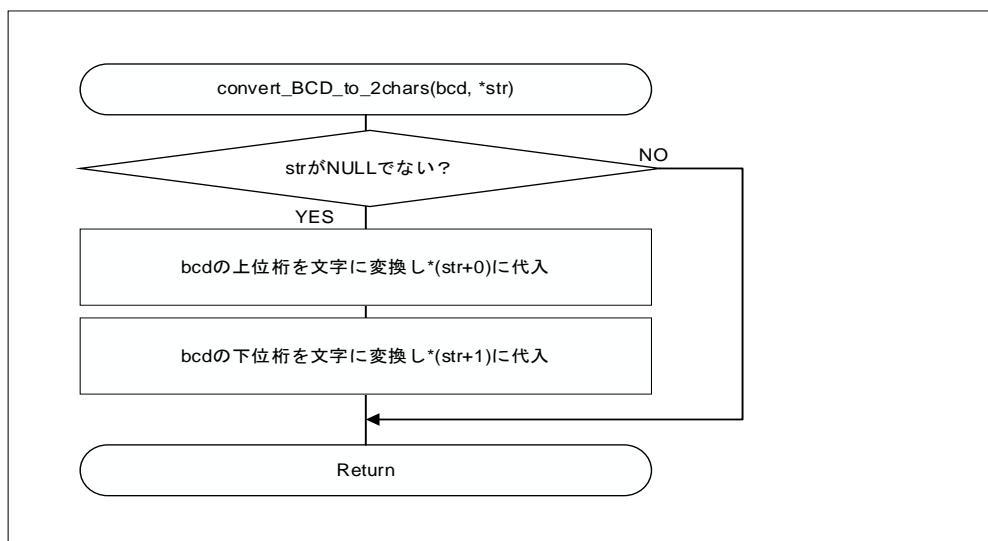
図 4-10 現在時刻表示処理 (2/2)



#### 4.6.10 BCD から文字への変換処理

図 4-11 に BCD から文字への変換処理のフローチャートを示します。

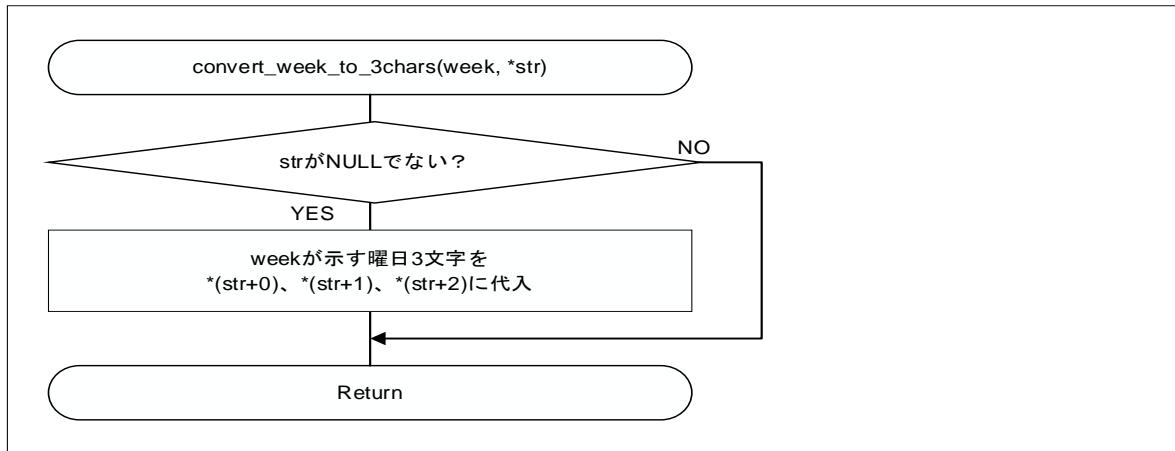
図 4-11 BCD から文字への変換処理



#### 4.6.11 曜日から文字への変換処理

図 4-12 に曜日から文字への変換処理のフローチャートを示します。

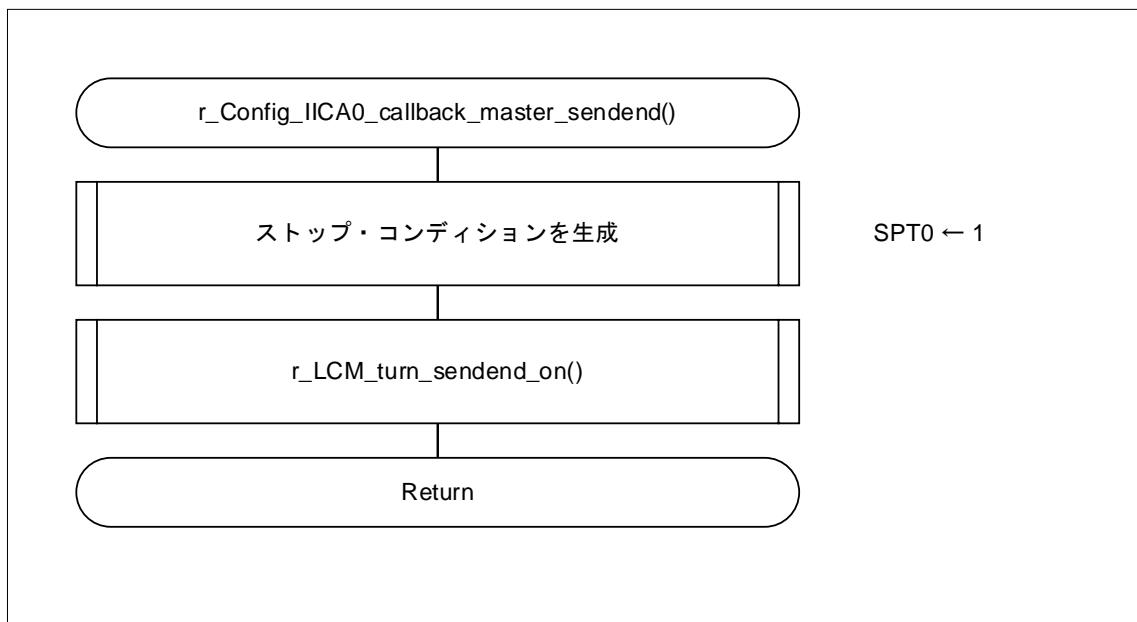
図 4-12 曜日から文字への変換処理



#### 4.6.12 IICA0 送信時のコールバック処理

図 4-13 に IICA0 送信完了時のコールバック処理のフローチャートを示します。

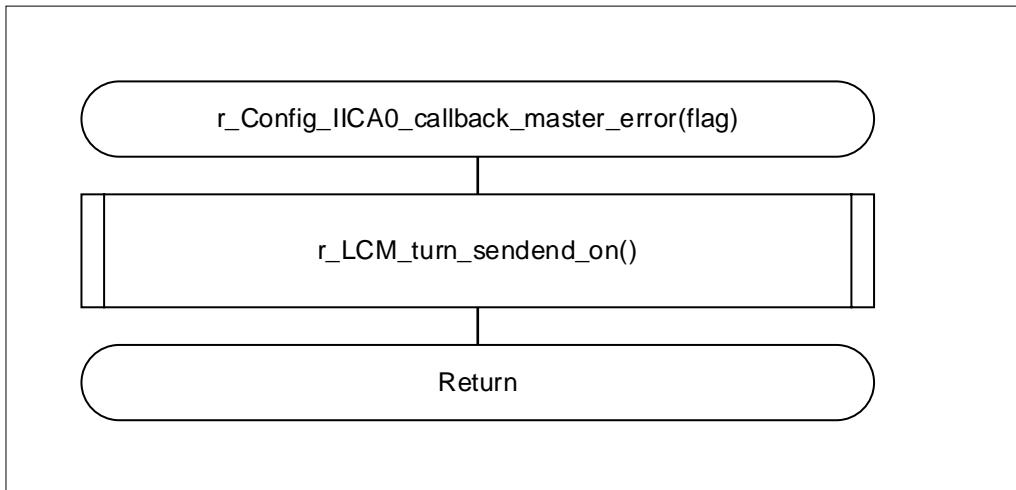
図 4-13 IICA0 送信完了時のコールバック処理



#### 4.6.13 IICA0 エラー発生時のコールバック処理

図 4-14 に IICA0 エラー発生時のコールバック処理のフローチャートを示します。

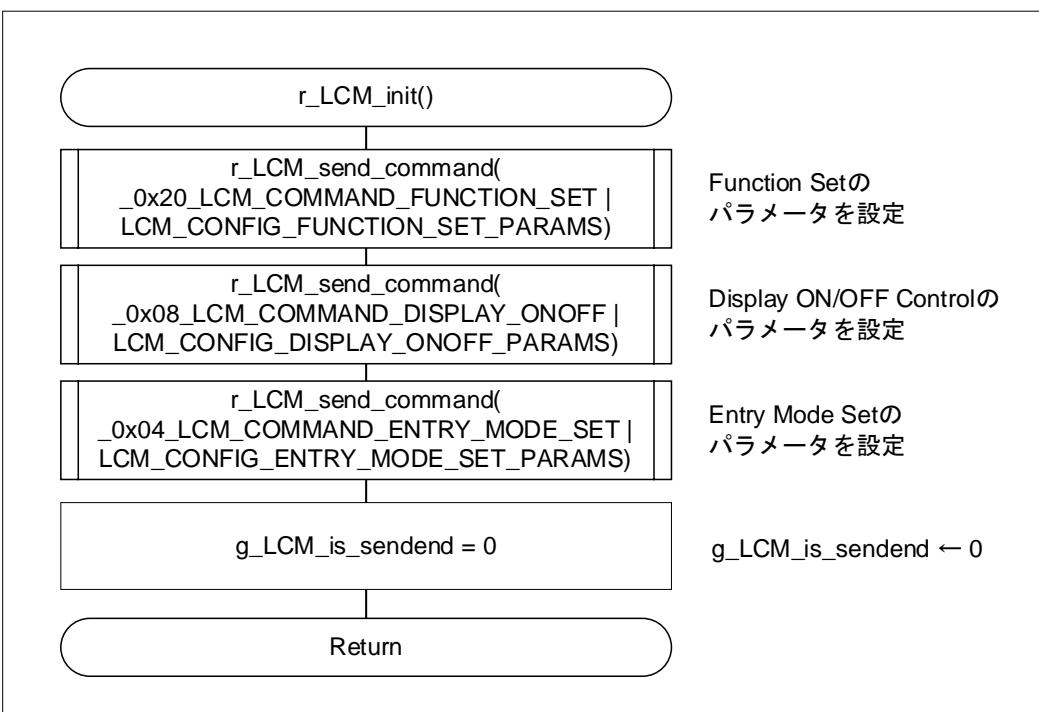
図 4-14 IICA0 エラー発生時のコールバック処理



#### 4.6.14 LCD モジュール初期化

図 4-15 に LCD モジュール初期化のフローチャートを示します。

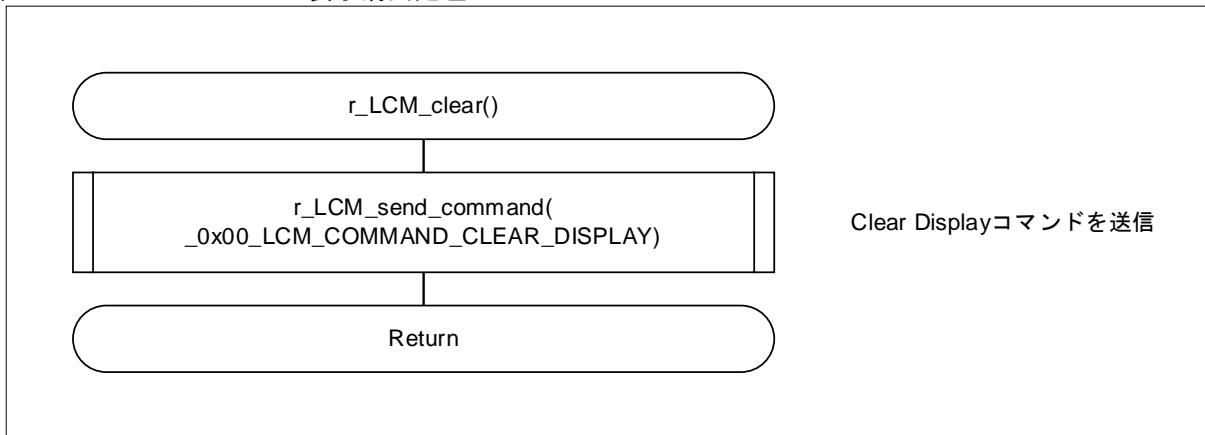
図 4-15 LCD モジュール初期化



#### 4.6.15 LCD モジュール 表示消去処理

図 4-16 に LCD モジュール 表示消去処理のフローチャートを示します。

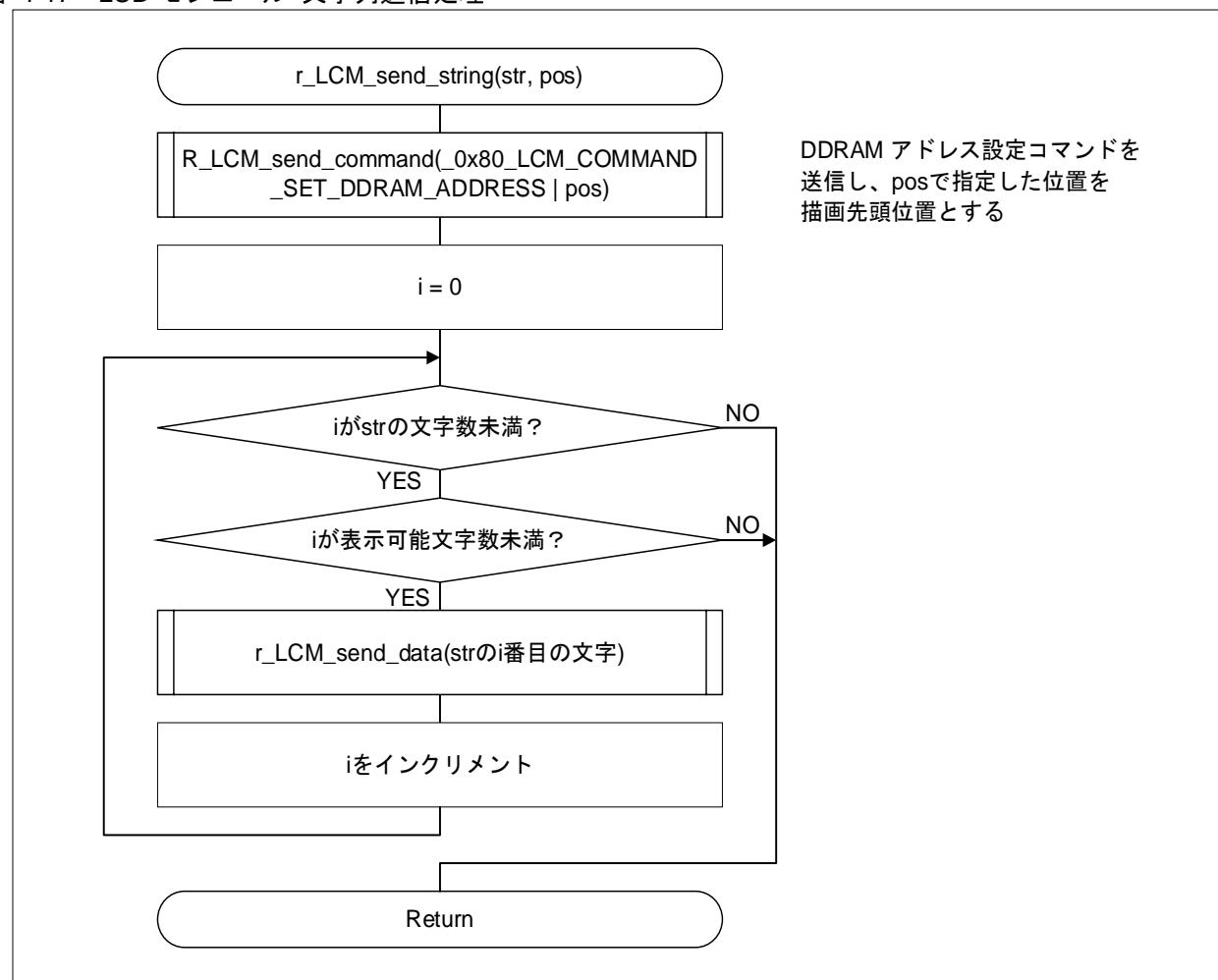
図 4-16 LCD モジュール 表示消去処理



#### 4.6.16 LCD モジュール 文字列送信処理

図 4-17 に LCD モジュール 文字列送信処理のフローチャートを示します。

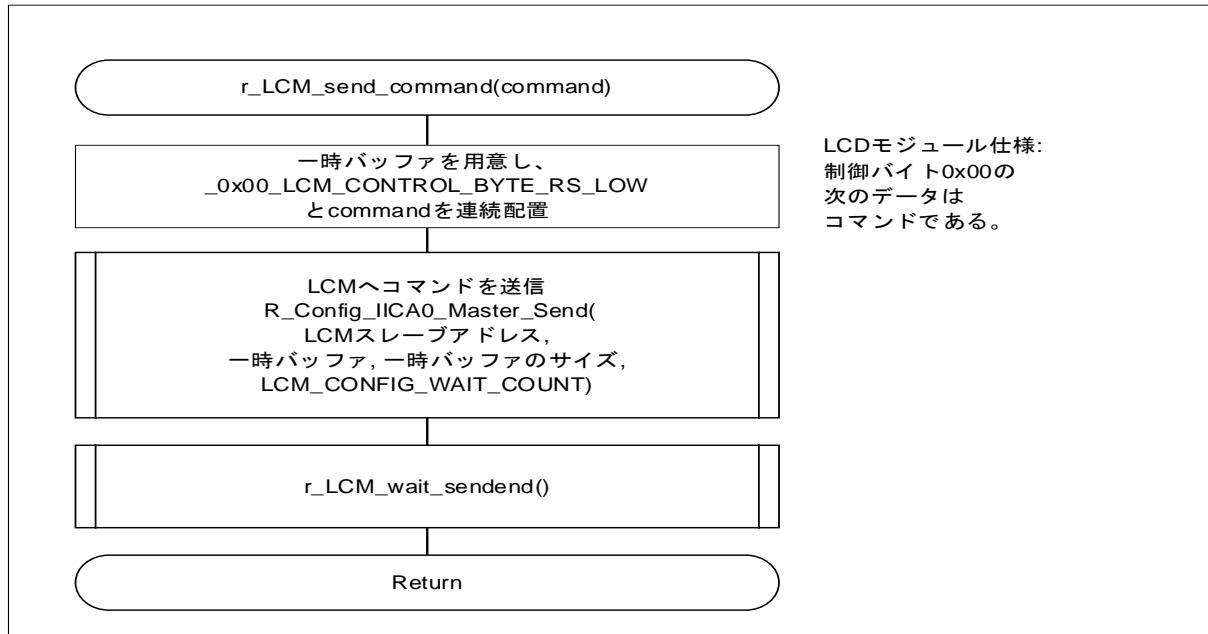
図 4-17 LCD モジュール 文字列送信処理



#### 4.6.17 LCD モジュール コマンド送信処理

図 4-18 に LCD モジュール コマンド送信処理のフローチャートを示します。

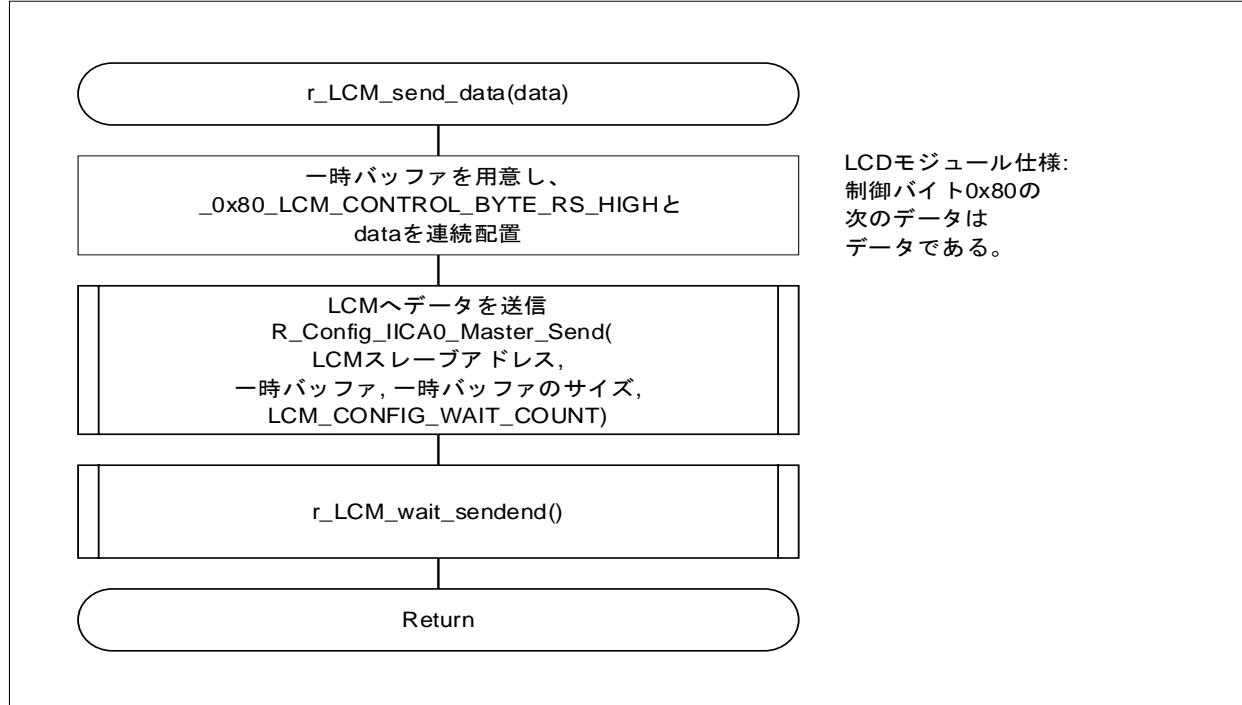
図 4-18 LCD モジュール コマンド送信処理



#### 4.6.18 LCD モジュール データ送信処理

図 4-19 に LCD モジュール データ送信処理のフローチャートを示します。

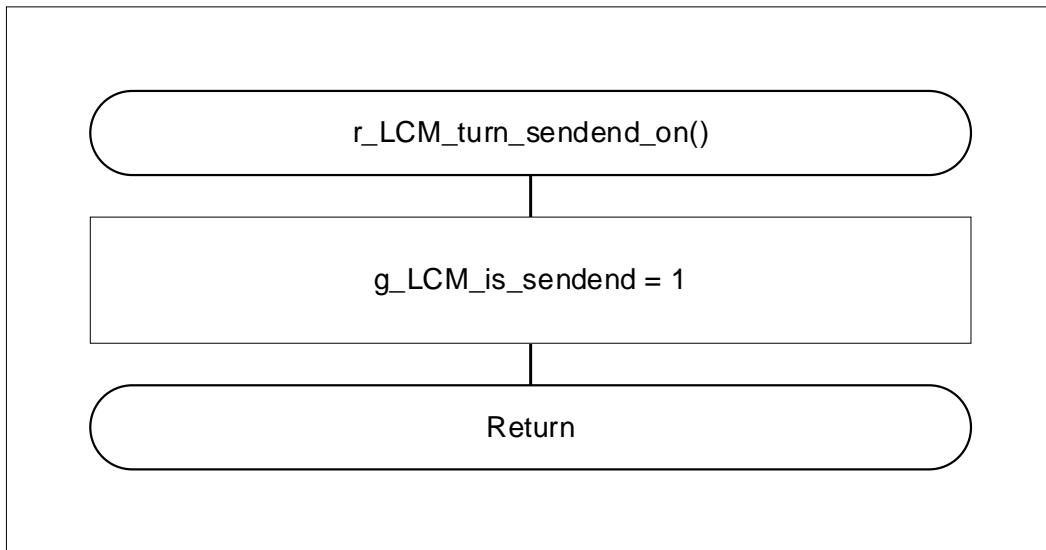
図 4-19 LCD モジュール データ送信処理



#### 4.6.19 LCD モジュール 通信終了フラグ設定

図 4-20 に LCD モジュール 通信終了フラグ設定のフローチャートを示します。

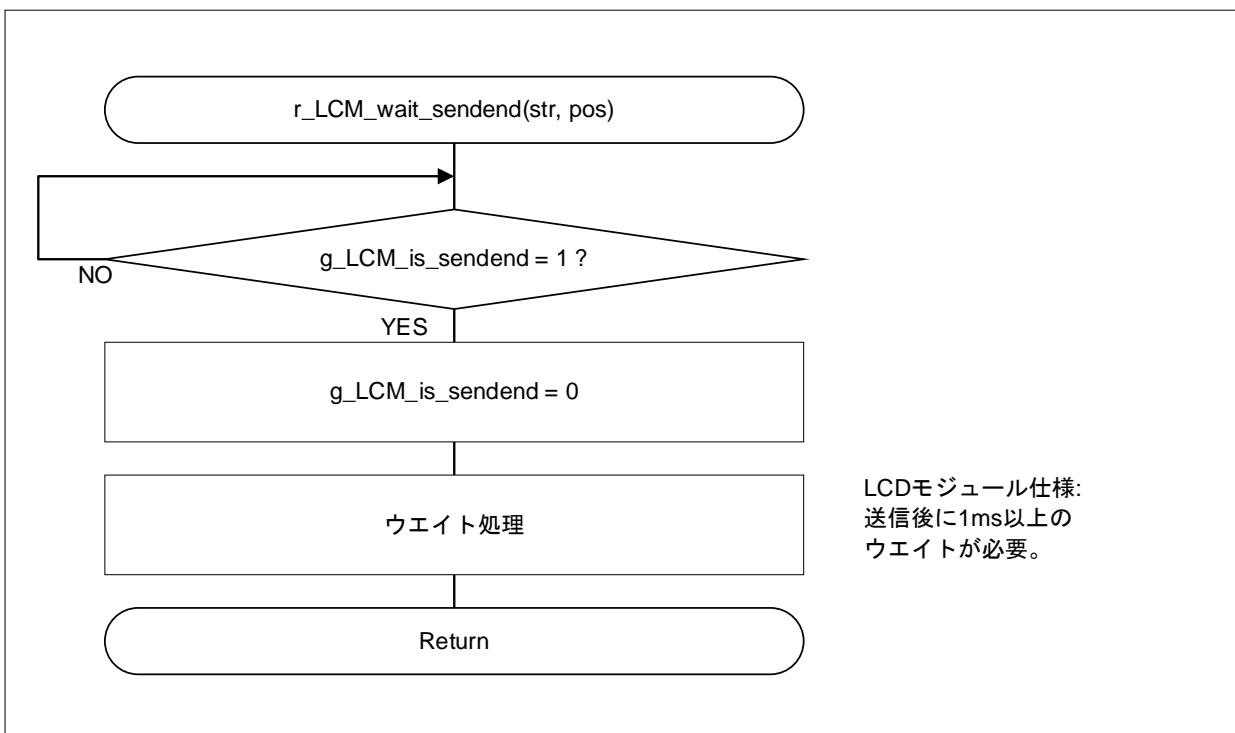
図 4-20 LCD モジュール 通信終了フラグ設定



#### 4.6.20 LCD モジュール 通信終了待ち処理

図 4-21 に LCD モジュール 通信終了待ち処理のフローチャートを示します。

図 4-21 LCD モジュール 通信終了待ち処理



## 5. サンプルコード

サンプルコードは、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。

注意. Smart configurator (SC) で生成されたソースコードを一部書き換えております。

SC でコードを再生成すると、下記ソースファイルの書き換えは無効となりますのでご注意ください。

- Config\_RTC.c
- Config\_RTC\_user.c
- r\_cg\_systeminit.c

## 6. 参考ドキュメント

RL78/G16 ユーザーズマニュアルハードウェア編 (R01UH0980J)

RL78 ファミリユーザーズマニュアルソフトウェア編 (R01US0015J)

(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

(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。)

LCD モジュールデータシート

(ACM1602NI-FLW-FBW-M01 (ZETTLER DISPLAYS) CHARACTER MODULE VER1.4)

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2023.10.11	—	初版発行
1.01	2023.10.16	全ページ	ヘッダー誤記修正
1.02	2023.11.20	p5	ツールバージョンを一部更新
		P26	SC でコード生成を行う際の注意事項を追記

## 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](http://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/](http://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。