

## RL78/I1A

### PMBus（スレーブ送受信）

---

#### 要旨

本アプリケーションノートでは、RL78/I1A に搭載されているシリアル・インタフェース IICA を使用して PMBus（スレーブ送受信）を実現するサンプルコードについて説明します。サンプルコードは、IICA を使用して PMBus のシングルマスタ・システムでのスレーブ動作（アドレス受信、データ送受信）を行います。

#### 対象デバイス

RL78/I1A

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
2. 動作確認条件.....	6
3. 関連アプリケーションノート .....	6
4. ハードウェア説明.....	7
4.1 使用端子一覧 .....	7
5. ソフトウェア説明.....	8
5.1 動作概要 .....	8
5.2 オプション・バイトの設定一覧 .....	10
5.3 関数一覧 .....	11
5.4 関数仕様 .....	12
5.5 フローチャート.....	14
5.5.1 全体フロー.....	14
5.5.2 初期設定関数.....	14
5.5.3 周辺機能の初期設定関数.....	15
5.5.4 main処理.....	16
5.5.5 タイマ制御処理 .....	17
5.5.6 IICA0受信データ取得処理.....	17
5.5.7 IICA0送信データ取得処理.....	18
5.5.8 IICA0割り込み処理.....	19
5.5.9 PMBus制御処理.....	21
5.5.10 通信ステータス取得処理.....	22
5.5.11 タイムアウト動作制御処理 .....	22
5.5.12 タイムアウト時間生成処理 .....	23
5.5.13 CRC8算出処理 .....	24
5.5.14 受信データ解析処理 .....	25
5.5.15 コマンド解析処理.....	26
5.5.16 送信データ作成処理 .....	27
5.5.17 受信データコピー処理 .....	28
5.5.18 通信ステータスエラー保存処理.....	29
5.5.19 PMBus User記述関数呼び出し処理 .....	29
5.6 ROM/RAMサイズ .....	30
6. サンプルコード .....	30
7. 参考ドキュメント.....	30
付録A 定数一覧.....	31
付録B 変数一覧.....	34
付録C 関数仕様.....	35
付録D PMBusコマンド.....	39
付録E 対応コマンドの追加、削除方法 .....	45
改訂記録.....	50

## 1. 仕様

本アプリケーションノートでは、PMBus（Power Management Bus）インタフェースによるスレーブ送受信（アドレス受信、データ送受信）を行う方法を説明します。

PMBus はオープン規格の電源管理用シリアル・インタフェースです。必要なコマンド言語を定義することで、各電源間での通信を容易にしています。バスラインで複数の電源を接続でき、制御することが出来ます。本サンプルコードは PMBus の通信部分のみ実装されており、電源に関する具体的な処理はユーザー関数として記載して頂く必要があります。

表 1.1 に使用する周辺機能と用途を、図 1.1 に PMBus の概要を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
シリアル・インタフェース IICA	PMBus におけるスレーブ送受信に使用します (SCLA0 端子と SDAA0 端子を使用)
タイマ・アレイ・ユニット チャンネル 0	インターバル・タイマ動作 (PMBus のタイムアウト時間の生成を行います)
ウォッチドッグ・タイマ	プログラム暴走時、リセットで復帰する為に使用します
P200-P206	PMBus のスレーブアドレスを設定するための ADDRESS 端子として使用します
P02	PMBus の SMBALERT#端子として使用します
P20	PMBus の CONTROL 端子として使用します
P75	PMBus の WRITE_PROTECT 端子として使用します

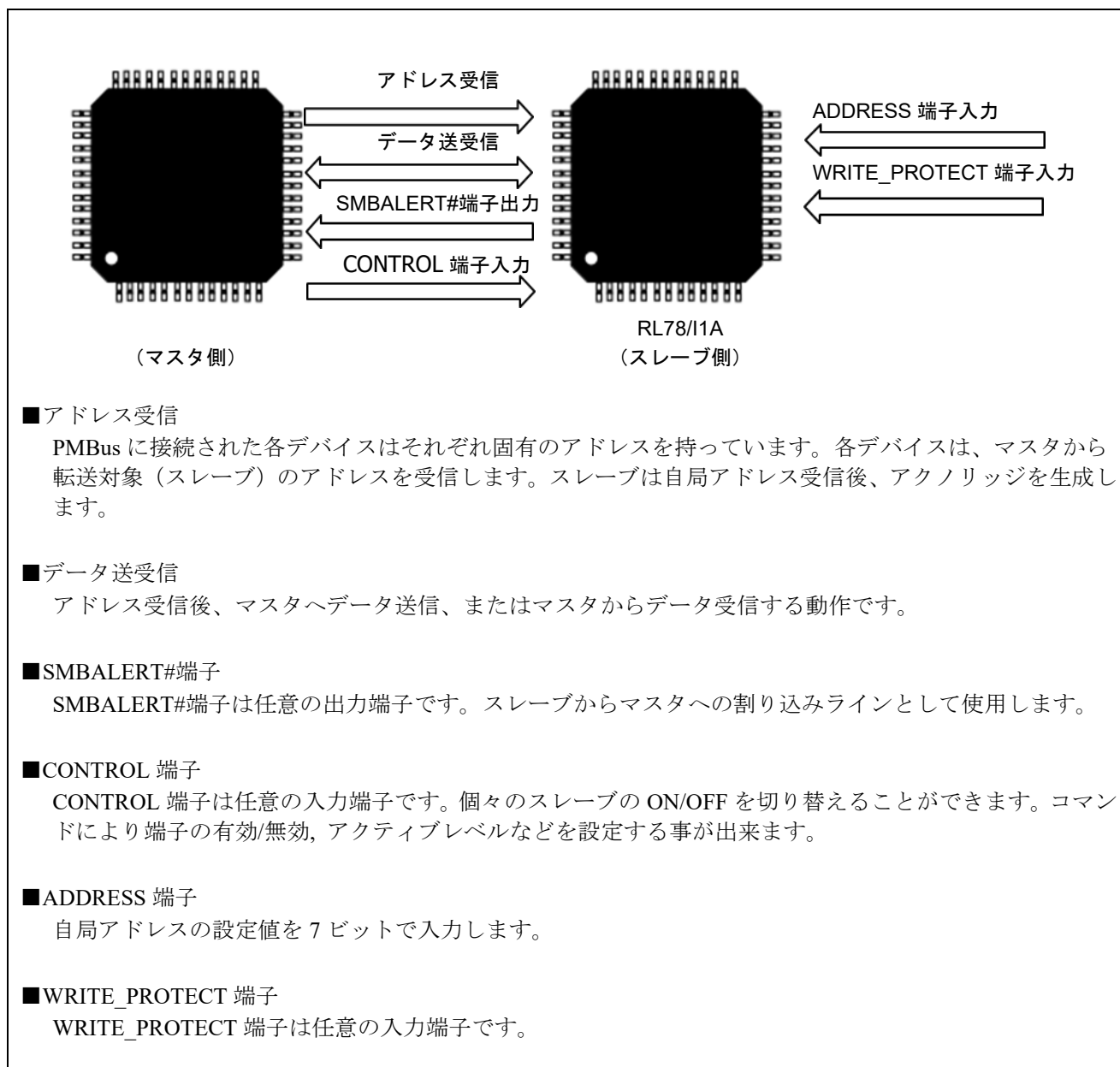


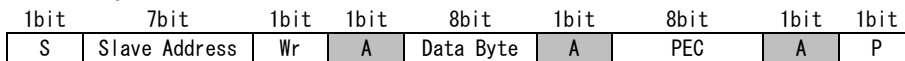
図 1.1 PMBus 通信概要

■ PMBus の通信フォーマット

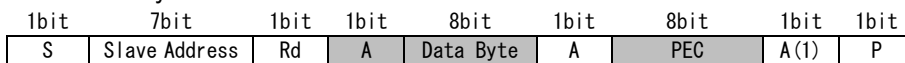
PMBus では通信方向やバイト数に応じてフォーマットが定められています。以下に主な種類を示します。

- S : スタート・コンディション
- Sr : リピートスタートコンディション
- Rd : リード(1)
- Wr : ライト(0)
- A : アクノリッジ(ACK : 0、NACK : 1)
- P : ストップ・コンディション
- PEC : Packet Error Code
- |  |   |           |
|--|---|-----------|
|  | : | マスター→スレーブ |
|  | : | スレーブ→マスタ  |

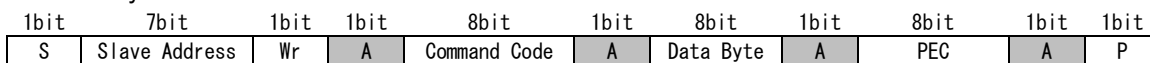
・ Send byte



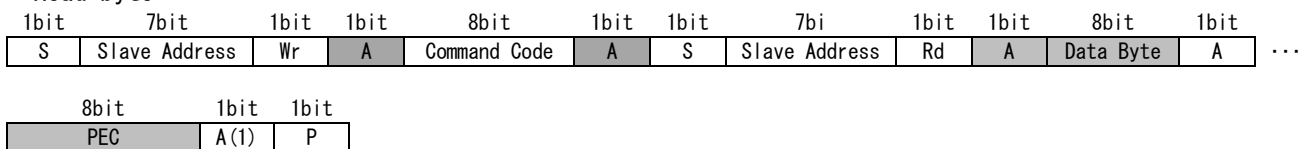
・ Receive byte



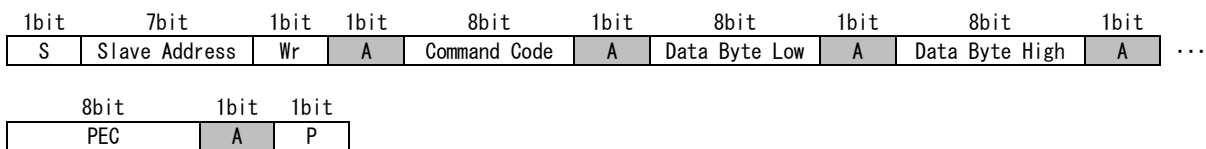
・ Write byte



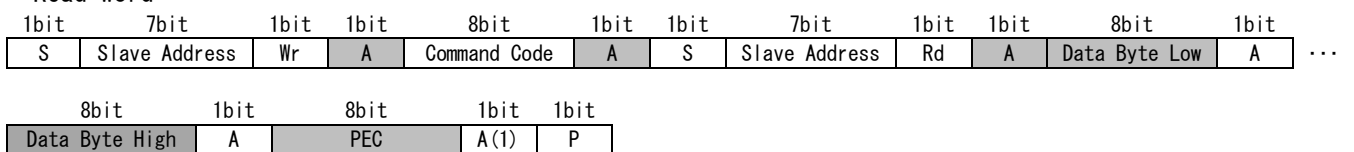
・ Read byte



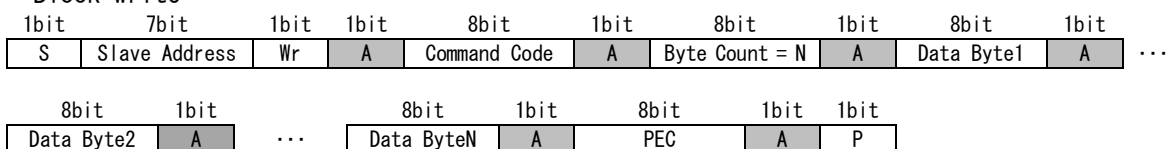
・ Write word



・ Read word



・ Block write



・ Block read

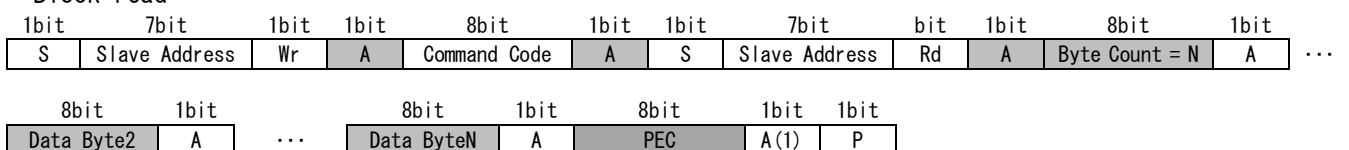


図 1.1 PMBus 通信概要(2/2)

## 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	RL78/I1A (R5F107DE)
動作周波数	●高速オンチップオシレータ (HOCO) クロック : 32MHz ●CPU/周辺ハードウェア・クロック : 32MHz
動作電圧	5.0V (4.0V~5.5V で動作可能)
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ V1.02.01
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CA78K0R V1.41
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ V8.04.00
C コンパイラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.09.00
統合開発環境 (e <sup>2</sup> studio)	ルネサス エレクトロニクス製 e <sup>2</sup> studio V2020-10
C コンパイラ (e <sup>2</sup> studio)	ルネサス エレクトロニクス製 CC-RL V1.09.00
統合開発環境 (IAR)	IAR Embedded Workbench for Renesas RL78 V4.20.1
C コンパイラ (IAR)	IAR C/C++ Compiler for Renesas RL78 V4.20.1.2260

## 3. 関連アプリケーションノート

本アプリケーションノートに関連するユーザーズマニュアルを以下に示します。併せて参照してください。

RL78/I1A ユーザーズマニュアル ハードウェア編 (R01UH0169)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015)

## 4. ハードウェア説明

### 4.1 使用端子一覧

表 4.1 に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
P10/SCLA0	入出力	IICA0 (PMBus) のシリアル・クロック入出力端子
P11/SDAA0	入出力	IICA0 (PMBus) のシリアル・データ送受信端子
P200-P206	入力	ADDRESS 端子
P02	出力	SMBALERT#端子 <sup>注1</sup>
P20	入力	CONTROL 端子 <sup>注2</sup>
P75	入力	WRITE_PROTECT 端子 <sup>注2</sup>

注意 PMBus の仕様では、SMBALERT#、CONTROL、WRITE\_PROTECT は、任意の信号ラインとなります。

- 注
- 1 N-ch オープンドレイン出力（VDD 耐圧）モードに設定し、外部でプルアップ抵抗に接続する必要があります。
  - 2 外部でプルアップ抵抗に接続する必要があります。

## 5. ソフトウェア説明

### 5.1 動作概要

サンプルコードは、シリアル・インタフェース IICA を利用して PMBus のスレーブ送受信 (アドレス受信、データ送受信) 動作を行います。サンプルコードではまず初期設定をおこない、マスタからの送信を待機します。マスタから自局宛のアドレスを受信した後、データを受信バッファへ格納します。その後、コマンド解析を行い、各コマンドに対応した処理関数(ユーザー関数)を呼び出します。電源に関する具体的な処理はユーザー関数内に追記して頂く必要があります。

- (1) 周辺 I/O ポートの初期設定を行います。

<設定条件>

- SMBALERT#端子用ポート (1 ビット) を出力モード、N-ch オープンドレイン出力 (VDD 耐圧) モードに設定します。初期値は High に設定します。
- ADDRESS 端子用ポート (7 ビット) を入力モードに設定します。
- CONTROL 端子用ポート (1 ビット) を入力モードに設定します。
- WRITE PROTECT 端子用ポート (1 ビット) を入力モードに設定します。
- P10/SCLA0、P11/SDAA0 は、TTL 入力バッファ、N-ch オープンドレイン出力 (VDD 耐圧) モードに設定します。

- (2) タイマ・アレイ・ユニットの初期設定を行います。

<設定条件>

- 動作クロックを fCLK に設定します。
- 動作モードをインターバル・タイマ・モードに設定します。
- コンペア値を 0x7CFF に設定します。(1ms 基準タイマ生成用)

- (3) シリアル・インタフェース IICA の初期設定を行います。

<設定条件>

- 動作モードを標準モードに設定します。
- 転送クロックを 100kHz に設定します。
- 自局アドレスを設定します。
- 9 クロック目で割り込みが発生するように設定します。
- ストップ・コンディション割り込み禁止に設定します。
- P10/SCLA0 端子をシリアル・クロックに、P11/SDAA0 端子をデータ送受信に設定します。

上記(1)~(3)実施後、マスタからのアドレス送信を待機します。

- (4) マスタから自局のアドレスまたは拡張コードを受信すると IICA0 の割り込みが発生します。続いて送られる PMBus のコマンドを IICA0 の割り込み毎に受信バッファに格納します。
- (5) ストップ・コンディションを検出後、コマンド解析を行います。

#### 【受信系コマンド受信時の動作】

受信データ(コマンド、PEC)を解析し、正常であれば、保存用の RAM に格納します。

その後、ユーザー関数を呼び出します。



**【送信系コマンド受信時の動作】**

受信データ(コマンド)を解析し、正常であれば、ユーザー関数を呼び出します。

その後、送信データを作成し送信開始します。

- (6) データ通信時、1 パケット通信中に 25ms 経過した場合、通信エラー (タイムアウト) と判断し、シリアル・インタフェース IICA の停止、初期設定を行います。

注意 1. 動作モードは、標準モードまたは、ファースト・モードに設定可能です。

また、転送クロックも同様に 100kHz または、400kHz に設定可能です。

※ 転送クロックの切り替えは、`r_iica0_driver.c` (76 行目) のコメントアウトを切り替えることで設定変更が可能です。

2. PMBus で使用しない周辺 I/O ポートは自由に使用することが出来ます。

## 5.2 オプション・バイトの設定一覧

表 5.1 にオプション・バイト設定を示します。

表 5.1 オプション・バイト設定

アドレス	設定値	内容
000C0H/010C0H	01111110B	ウォッチドッグ・タイマのカウンタ動作許可 (リセット解除後, カウント開始)
000C1H/010C1H	11111111B	LVD オフ(外部リセット使用)
000C2H/010C2H	11101000B	HS (高速メイン) モード 高速内蔵発振回路の周波数 : 32 MHz
000C3H/010C3H	10000100B	オンチップ・デバッグ許可

### 5.3 関数一覧

表 5.2 に関数を示します。

表 5.2 関数

関数名	概要
R_TAU0_Create	タイマ・アレイ・ユニット初期化処理
R_TAU0_Channel0_Start	タイマ・アレイ・ユニット動作開始処理
R_TAU0_Channel0_Stop	タイマ・アレイ・ユニット動作停止処理
R_TIMER_Control	1ms 基準タイマ生成処理
R_PORT_Create	ポート初期化処理
R_PORT_Smbalert	SMBALERT#ポート出力設定処理
R_PORT_Get_Address	ADDRESS ポート入力値取得処理
R_PORT_Get_Control	CONTROL ポート入力値取得処理
R_PORT_Get_Write_Protect	WRITE_PROTECT ポート入力値取得処理
R_IICA0_Create	IICA0 初期化処理
R_IICA0_Stop	IICA0 停止処理
R_IICA0_Init	IICA0 Driver Module 初期化処理
R_IICA0_Get_Receivedata	受信データ取得処理
R_IICA0_Set_Senddata	送信データ設定処理
R_PMBUS_Init	PMBus Control Module 初期化処理
R_PMBUS_Set_Analysis_Sts	通信ステータス設定処理
R_PMBUS_Timeout_Enable	Timeout タイマ動作制御処理
R_PMBUS_Control	PMBus Control Module 制御処理
R_PMBUS_Timeout_Make	Timeout 時間生成処理
R_PMBUS_Fault_Handler	エラー情報設定処理
R_PMBUS_User_App	コマンド対象関数呼び出し処理

## 5.4 関数仕様

サンプルコード中の主な関数の仕様を示します。

### R\_IICA0\_Get\_Receivedata

---

概要	受信データ取得処理	
ヘッダ	r_iica0_driver.h	
宣言	uint8_t *R_IICA0_Get_Receivedata(void)	
説明	受信データ格納バッファの先頭アドレスを取得します。	
引数	なし	
リターン値	buffer_address	受信バッファの先頭アドレス
備考	なし	

### R\_IICA0\_Set\_Senddata

---

概要	送信データ設定処理	
ヘッダ	r_iica0_driver.h	
宣言	void R_IICA0_Set_Senddata( uint8_t* )	
説明	送信データをバッファへコピーし、1byte 目のデータを送信します。	
引数	send_buffer	送信データ格納バッファの先頭アドレス
リターン値	なし	
備考	なし	

### R\_PMBUS\_Set\_Analysis\_Sts

---

概要	通信ステータス設定処理	
ヘッダ	r_pmbus_control.h	
宣言	void R_PMBUS_Set_Analysis_Sts( uint8_t )	
説明	通信ステータスをセットします。	
引数	analysis_sts	通信ステータス
リターン値	なし	
備考	なし	

### R\_PMBUS\_Timeout\_Enable

---

概要	Timeout タイマ動作制御処理	
ヘッダ	r_pmbus_control.h	
宣言	void R_PMBUS_Timeout_Enable( uint8_t )	
説明	Timeout タイマの動作を許可します。	
引数	timer_enable	タイマ動作許可/禁止フラグ
リターン値	なし	
備考	なし	

---

**R\_PMBUS\_Timeout\_Make**

---

概要	Timeout 時間生成処理
ヘッダ	r_pmbus_control.h
宣言	void R_PMBUS_Timeout_Make(void)
説明	タイムアウト(25ms)時間をカウントして、タイムアウト発生を判断します。
引数	なし
リターン値	なし
備考	なし

---

**R\_PMBUS\_Fault\_Handler**

---

概要	エラー情報設定処理
ヘッダ	r_pmbus_control.h
宣言	void R_PMBUS_Fault_Handler( uint8_t )
説明	エラー情報を内部ステータスにセットします。
引数	error_result                      エラー情報
リターン値	なし
備考	なし

---

**R\_PMBUS\_Control**

---

概要	PMBus Control Module 制御処理
ヘッダ	r_pmbus_control.h
宣言	void R_PMBUS_Control(void)
説明	PMBus 通信全体の動作制御を行う関数です。
引数	なし
リターン値	なし
備考	なし

---

**R\_PMBUS\_User\_App**

---

概要	コマンド対象関数呼び出し
ヘッダ	r_pmbus_userapp.h
宣言	void R_PMBUS_User_App( uint8_t )
説明	コマンド対象関数を呼び出します。
引数	array_index                      対象コマンド
リターン値	なし
備考	なし

## 5.5 フローチャート

### 5.5.1 全体フロー

図 5.1 に本サンプルコードの全体フローを示します。

#### 【概要】

本サンプルコードは起動後 `hdwinit()`にて各使用機能の初期設定を行い、`main()`を呼び出します。

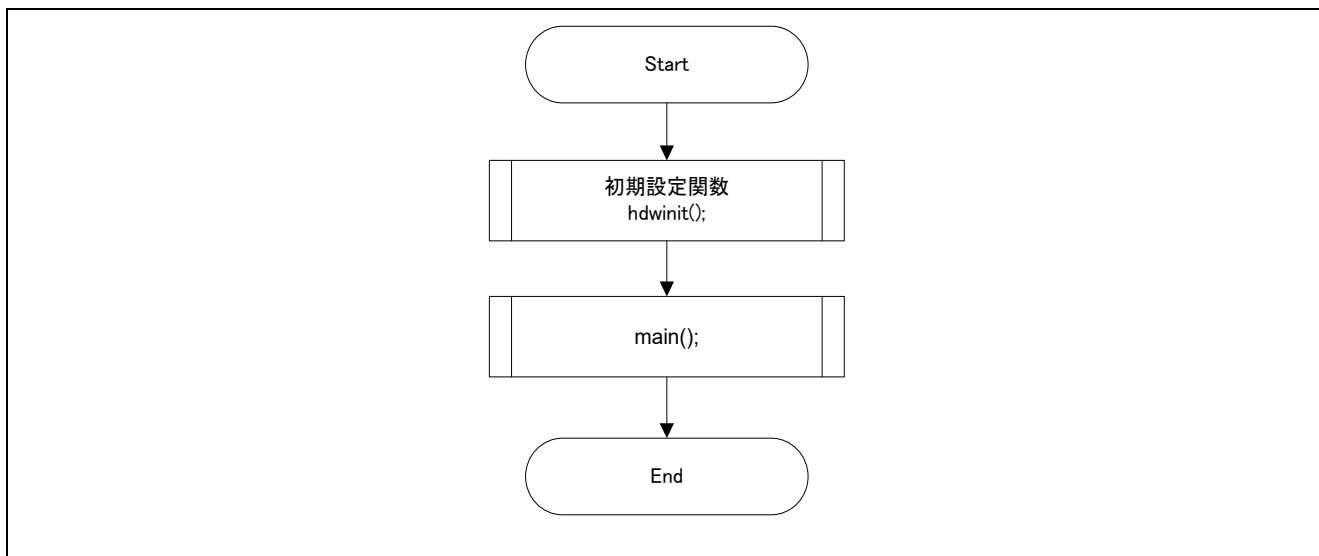


図 5.1 全体フロー

### 5.5.2 初期設定関数

図 5.2 に初期設定関数のフローチャートを示します。

#### 【概要】

`hdwinit()`は割り込み禁止状態中に、`r_systeminit()`で各使用機能の初期設定を行います。

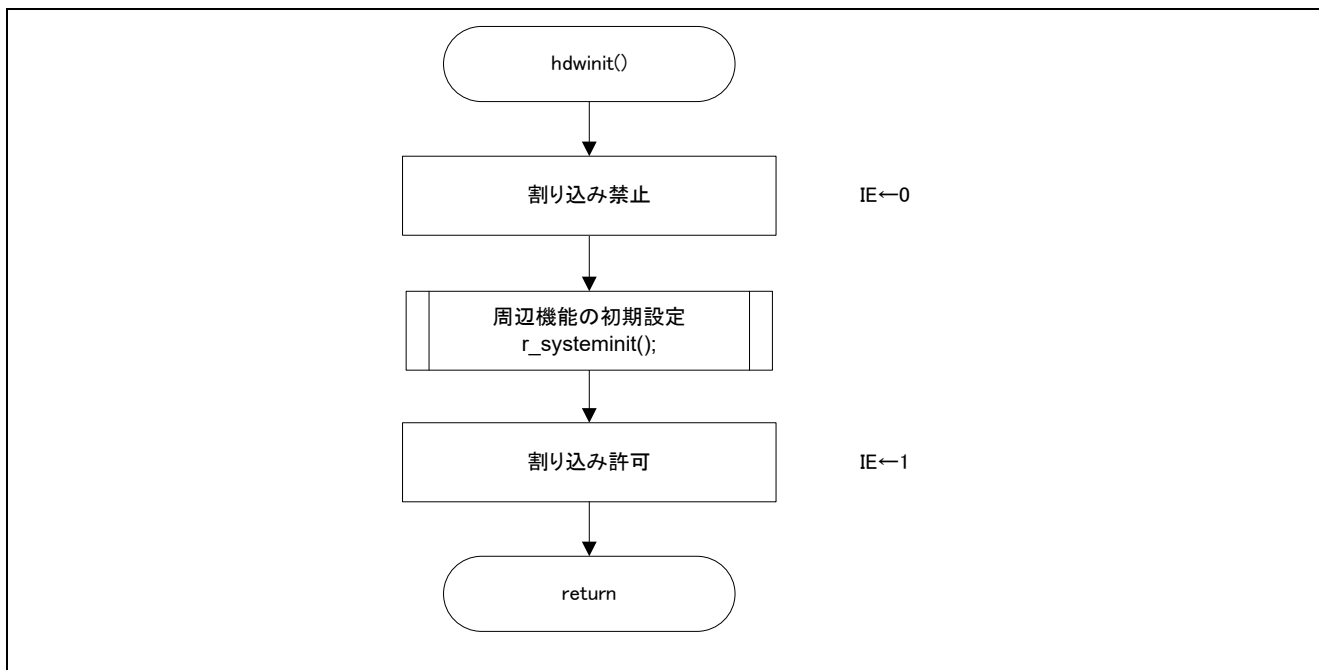


図 5.2 初期設定関数

### 5.5.3 周辺機能の初期設定関数

図 5.3 にシステム関数のフローチャートを示します。

#### 【概要】

r\_systeminit()は各使用機能の初期設定を行います。R\_PORT\_Create()は使用端子の初期設定を行います。r\_cgc\_create()は動作クロックの初期設定を行います。R\_IICA0\_Create()はIICA0の初期設定を行います。R\_TAU0\_Create()はTAU0を使用した1msを生成する設定を行います。この1msはPMBusの通信処理で時間管理の基準となっています。

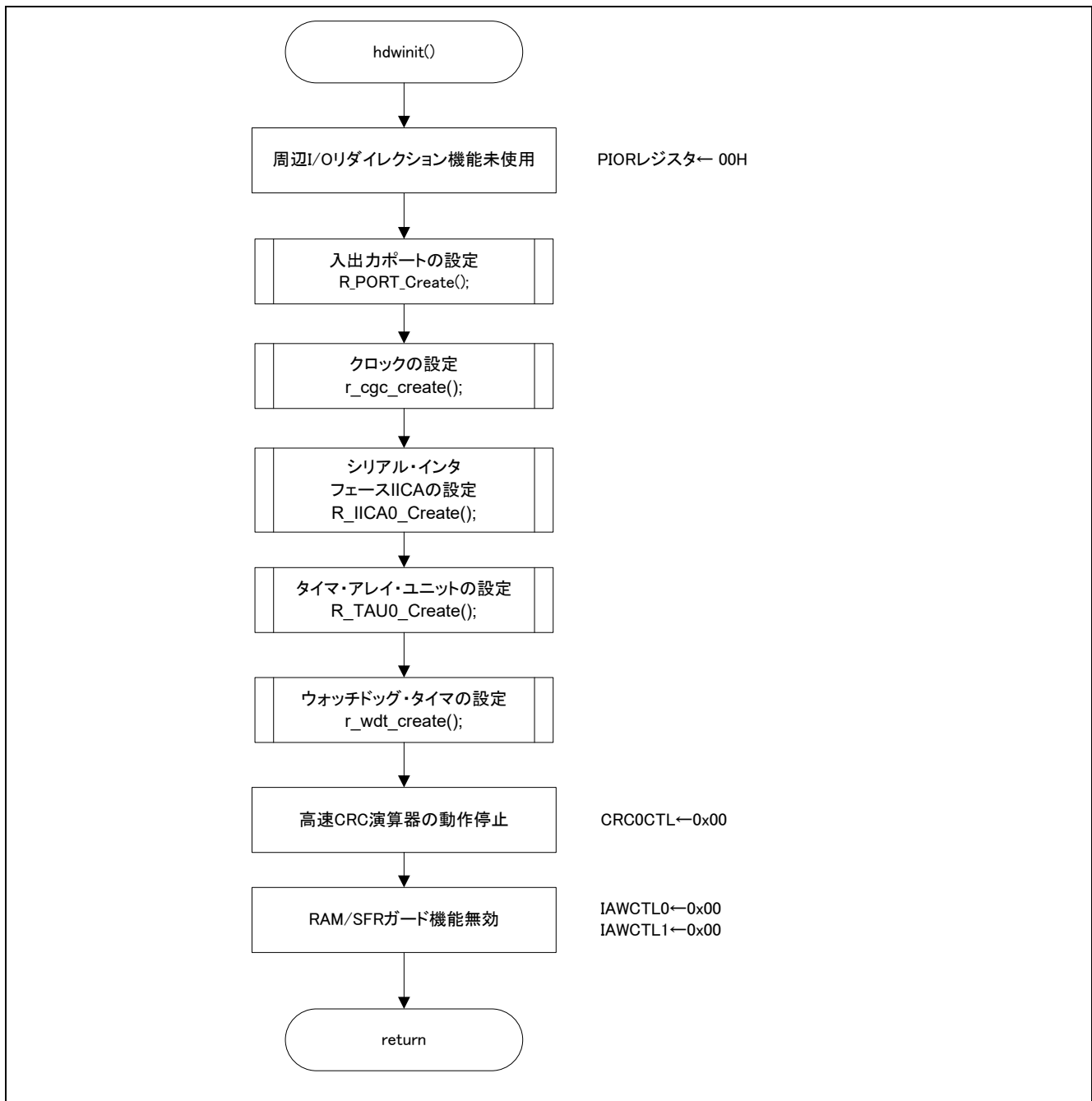


図 5.3 システム関数

### 5.5.4 main 処理

図 5.4 に main 処理のフローチャートを示します。

#### 【概要】

main()は本サンプルコードの main 処理を行います。moduleinit()で各モジュールの変数を初期化します。R\_TAU0\_Channel0\_Start()は TAU0 を動作開始します。r\_wdt\_restart()は周期毎にウォッチドッグ・タイマをリスタートします。R\_TIMER\_Control()は PMBus の時間管理の基準である 1ms の経過をポーリングします。R\_PMBUS\_Control()は PMBus の動作制御を行います。

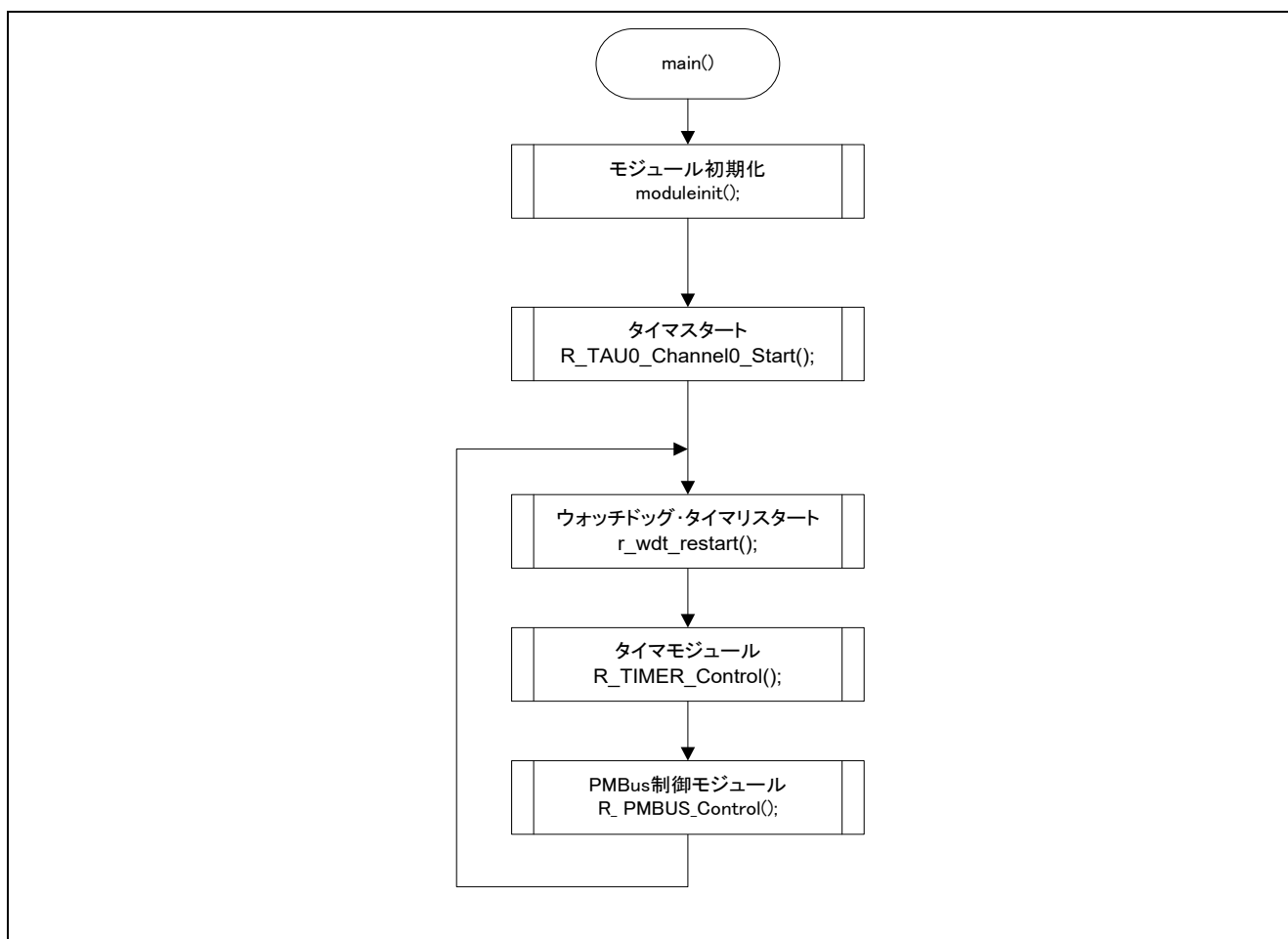


図 5.4 main 処理



### 5.5.5 タイマ制御処理

図 5.5 にタイマ制御処理のフローチャートを示します。

#### 【概要】

R\_TIMER\_Control()は、PMBus の時間管理の基準である 1ms の経過をポーリングします。

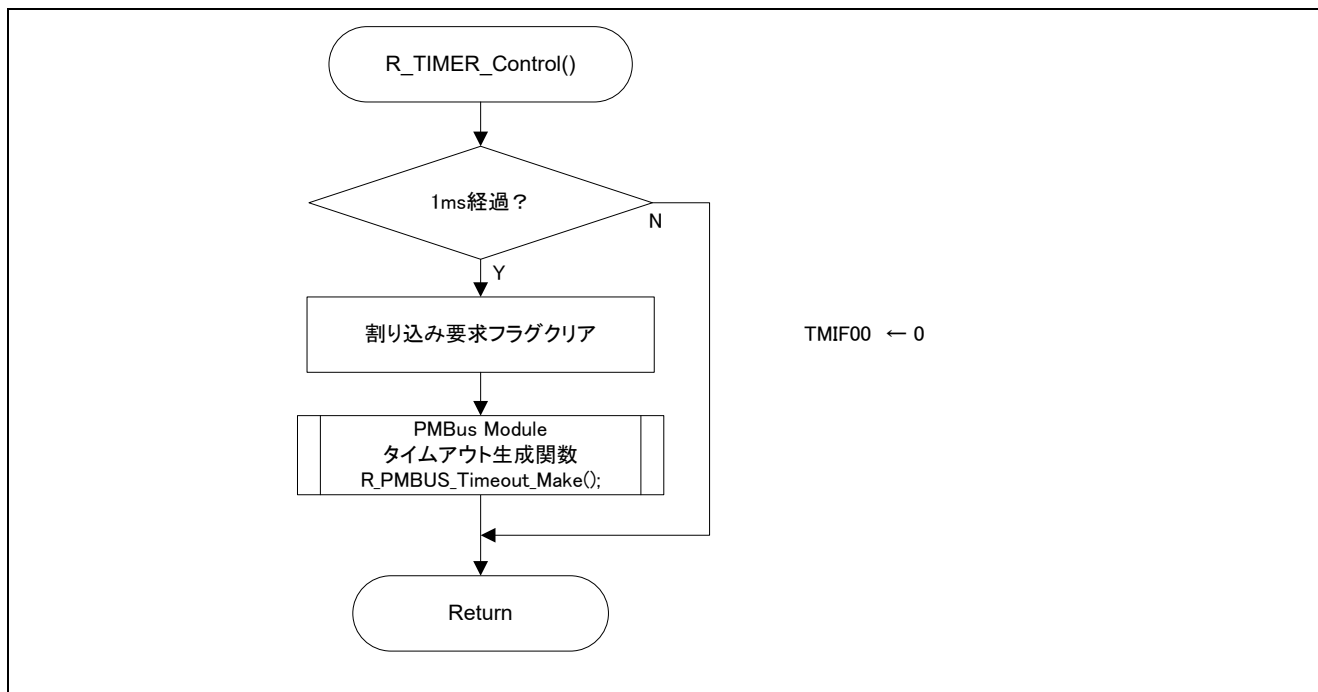


図 5.5 タイマ制御処理

### 5.5.6 IICA0 受信データ取得処理

図 5.6 に IICA0 受信データ取得処理のフローチャートを示します。

#### 【概要】

R\_IICA0\_Get\_Receivedata()は受信用バッファのアドレスを呼び出し元へ返します。

r\_pmbus\_command\_check(),r\_pmbus\_receive\_check()関数内で呼び出されています。

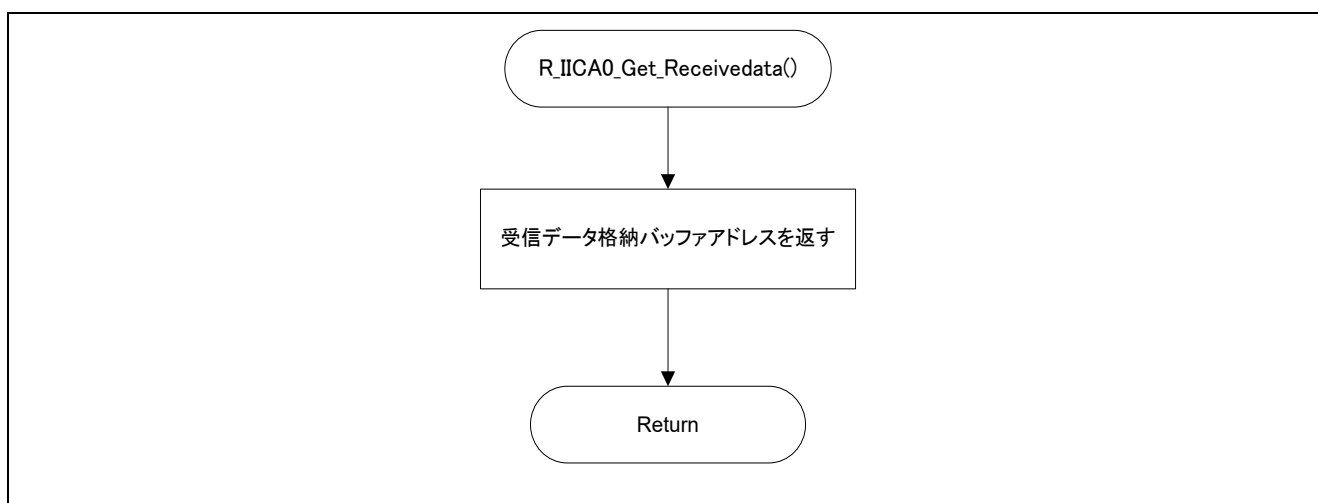


図 5.6 IICA0 受信データ取得処理

### 5.5.7 IICA0 送信データ取得処理

図 5.7 に IICA0 送信データ取得処理のフローチャートを示します。

#### 【概要】

R\_IICA0\_Set\_Senddata()は、送信データを送信用バッファへコピーし、1byte 目のデータを送信します。

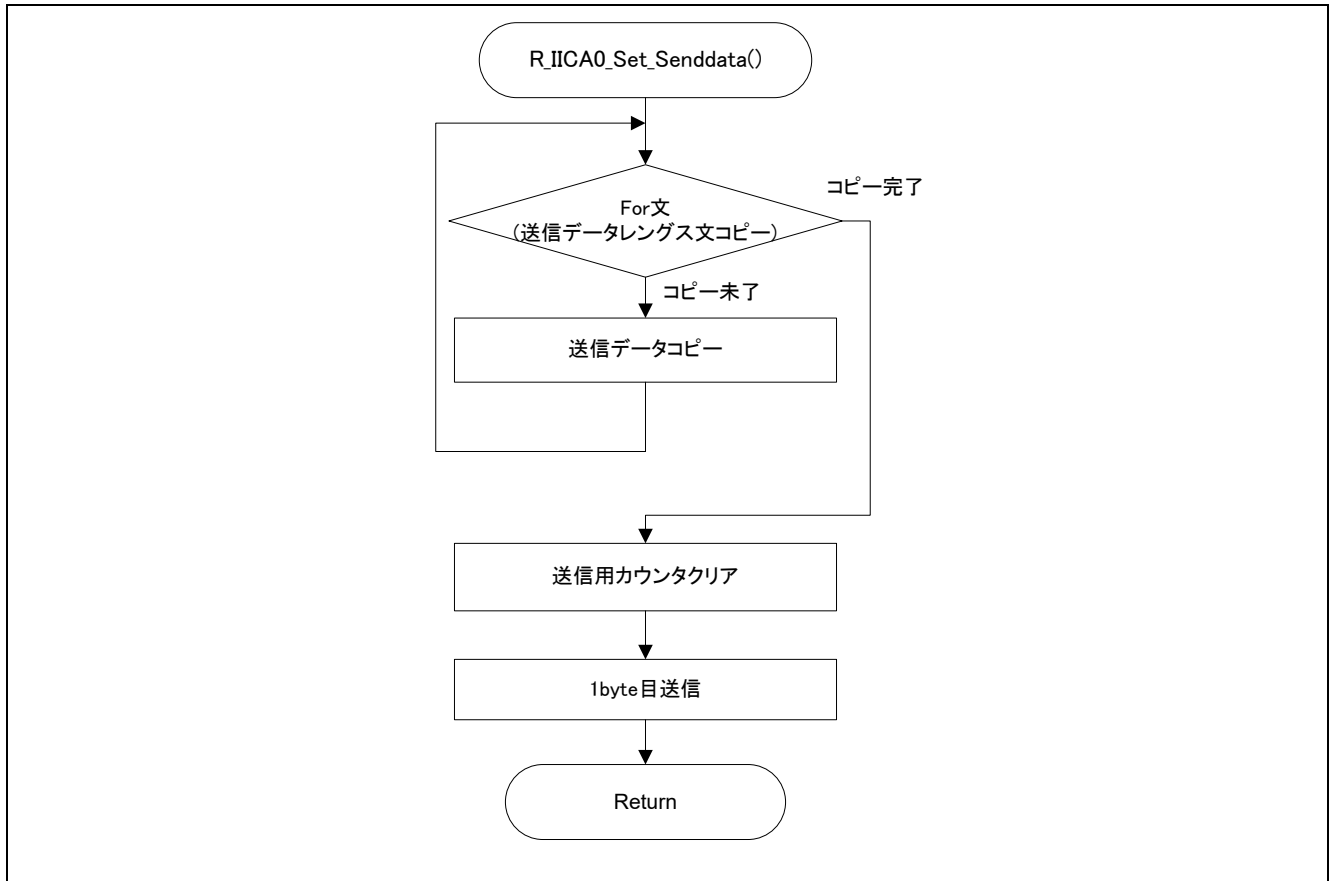


図 5.7 IICA0 送信データ取得処理

5.5.8 IICA0 割り込み処理

図 5.8 に IICA0 割り込み処理のフローチャートを示します。

【概要】

r\_iica0\_interrupt()は、IICA0 通信完了割り込み処理を行います。割り込み発生に応じてタイムアウト用タイマの起動・停止処理、受信データの取得、ウェイトの解除などを行います。

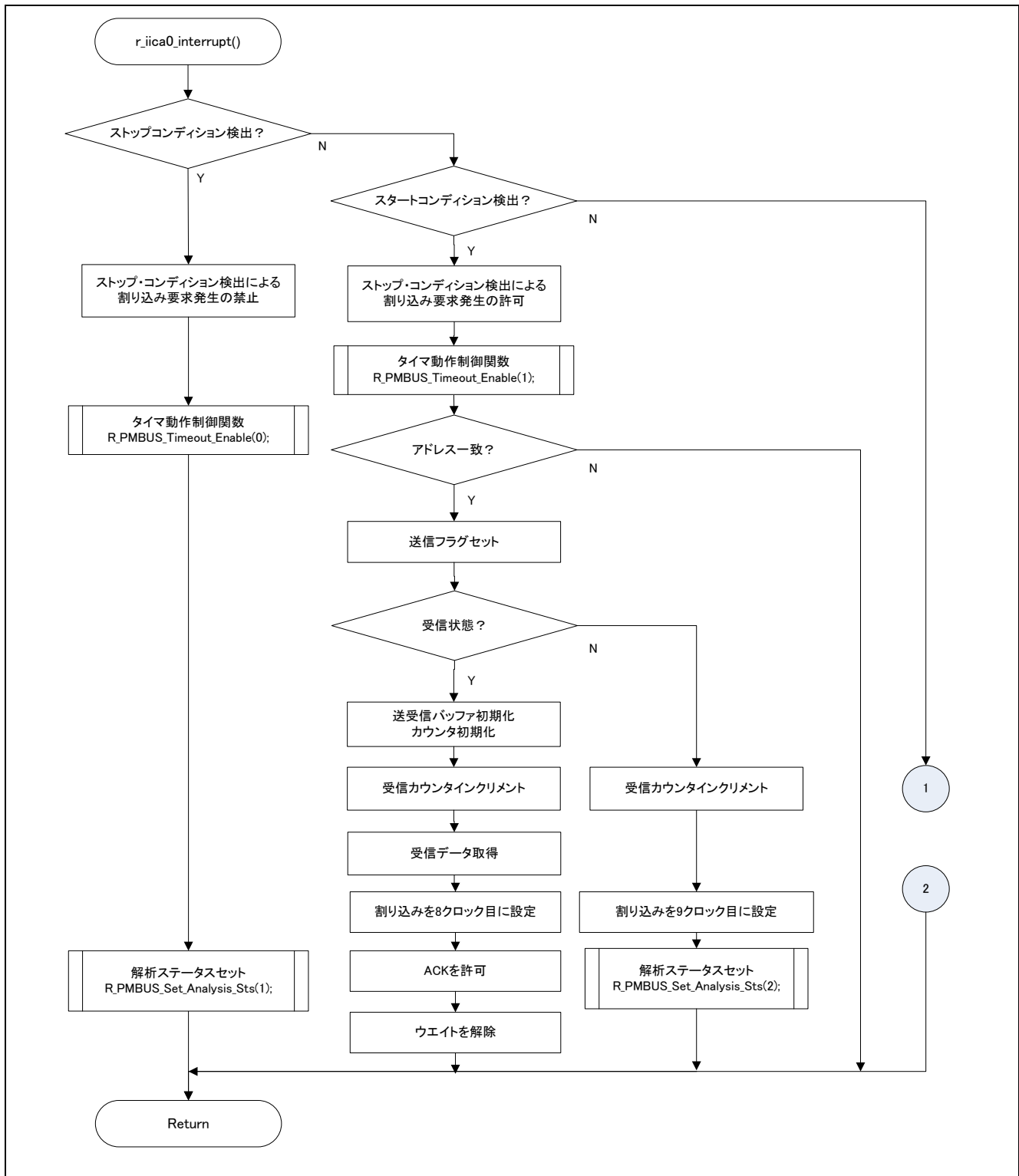


図 5.8 IICA0 割り込み処理 (1/2)

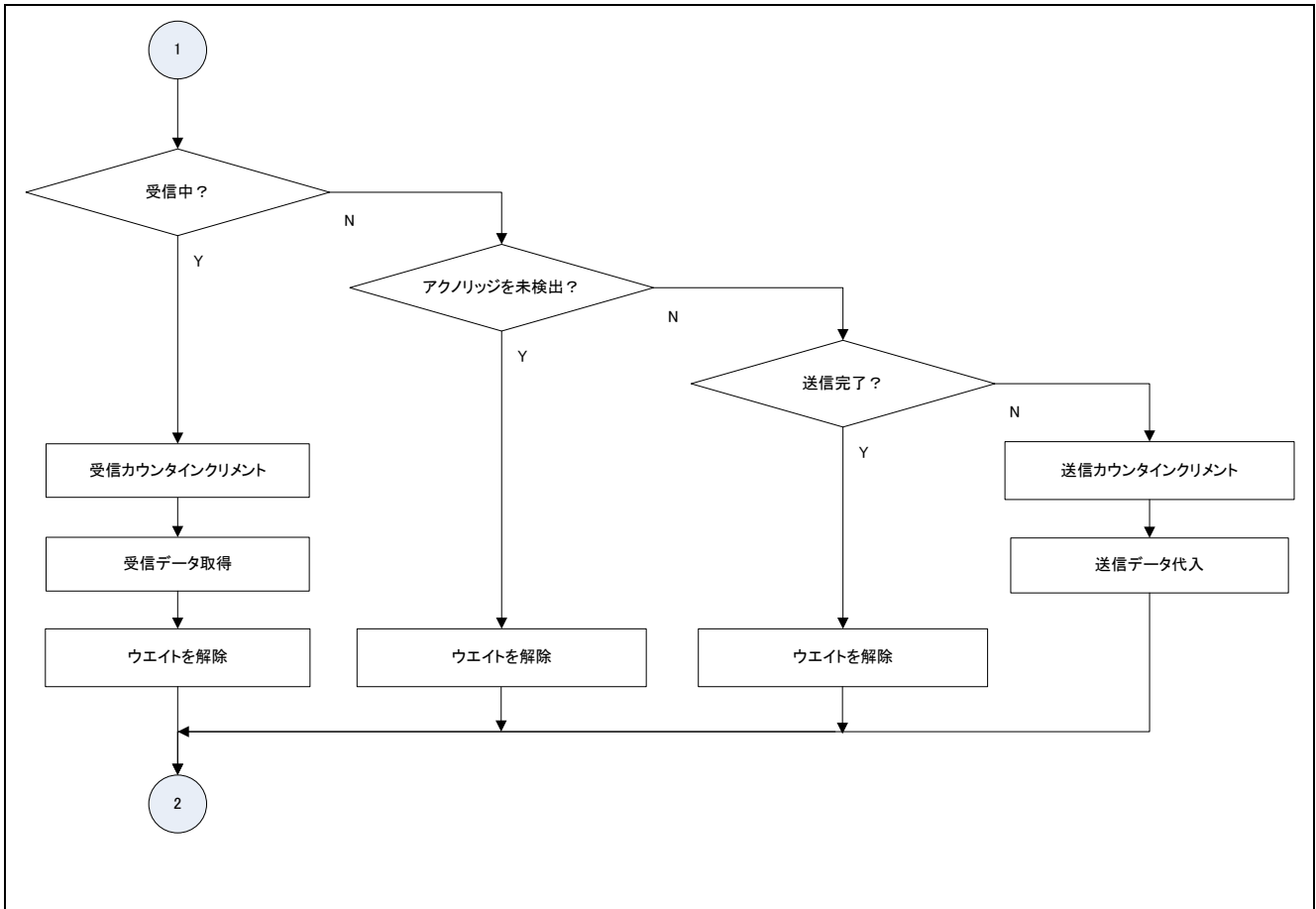


図 5.8 IICA0 割り込み処理 (2/2)

5.5.9 PMBus 制御処理

図 5.9 に PMBus 制御処理のフローチャートを示します。

【概要】

R\_PMBUS\_Control()は、PMBus 通信全体の制御を行っています。主に受信データの解析関数、ユーザー関数、送信データ作成関数の呼び出しを通信のステータスに応じて行います。

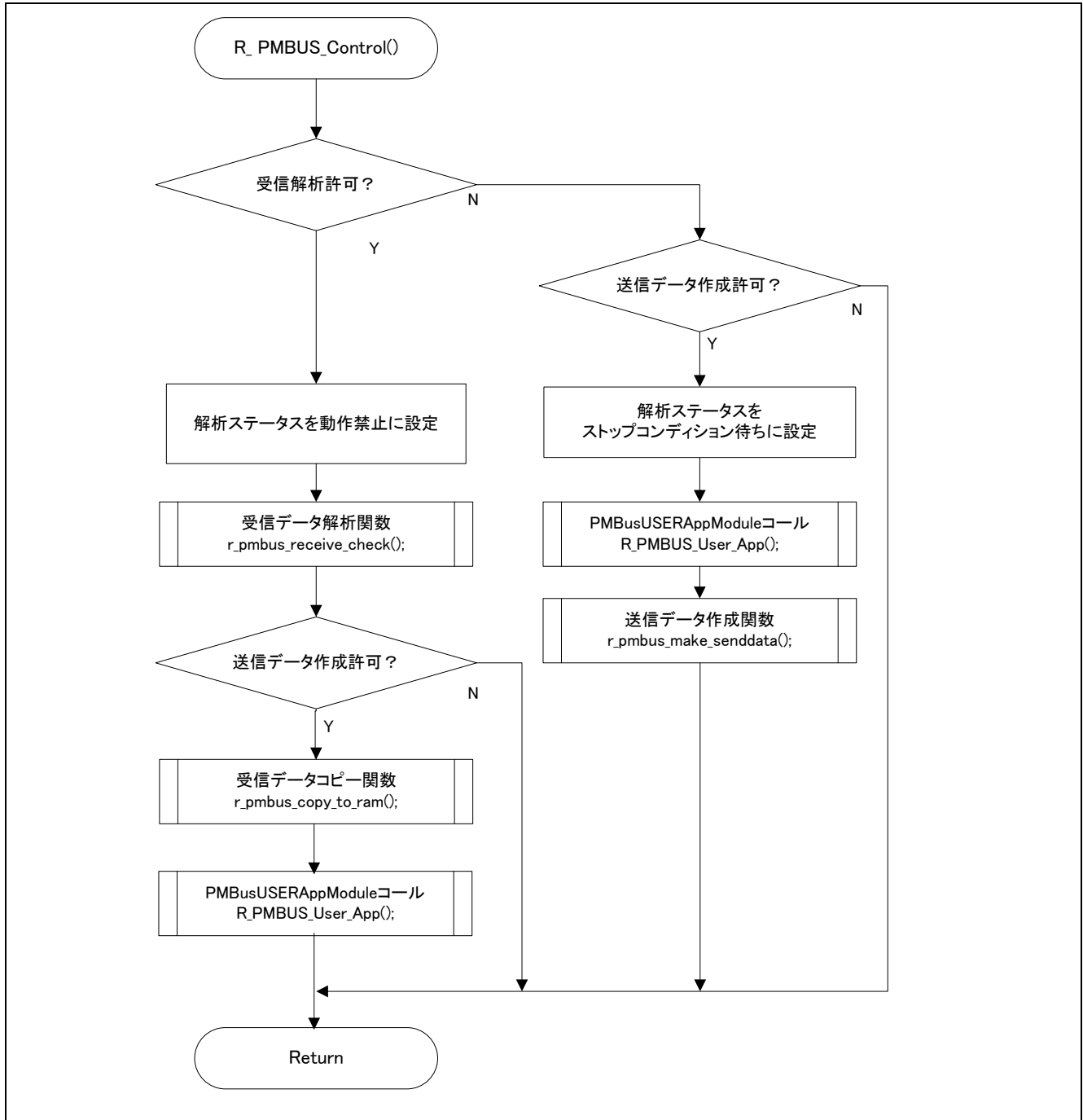


図 5.9 PMBus 制御処理

### 5.5.10 通信ステータス取得処理

図 5.10 に通信ステータス取得処理のフローチャートを示します。

#### 【概要】

R\_PMBUS\_Set\_Analysis\_Sts()は IICA0 通信完了割り込み処理内で呼び出され、引数で送受信の状態を取得し、戻り値として上位関数に返します。

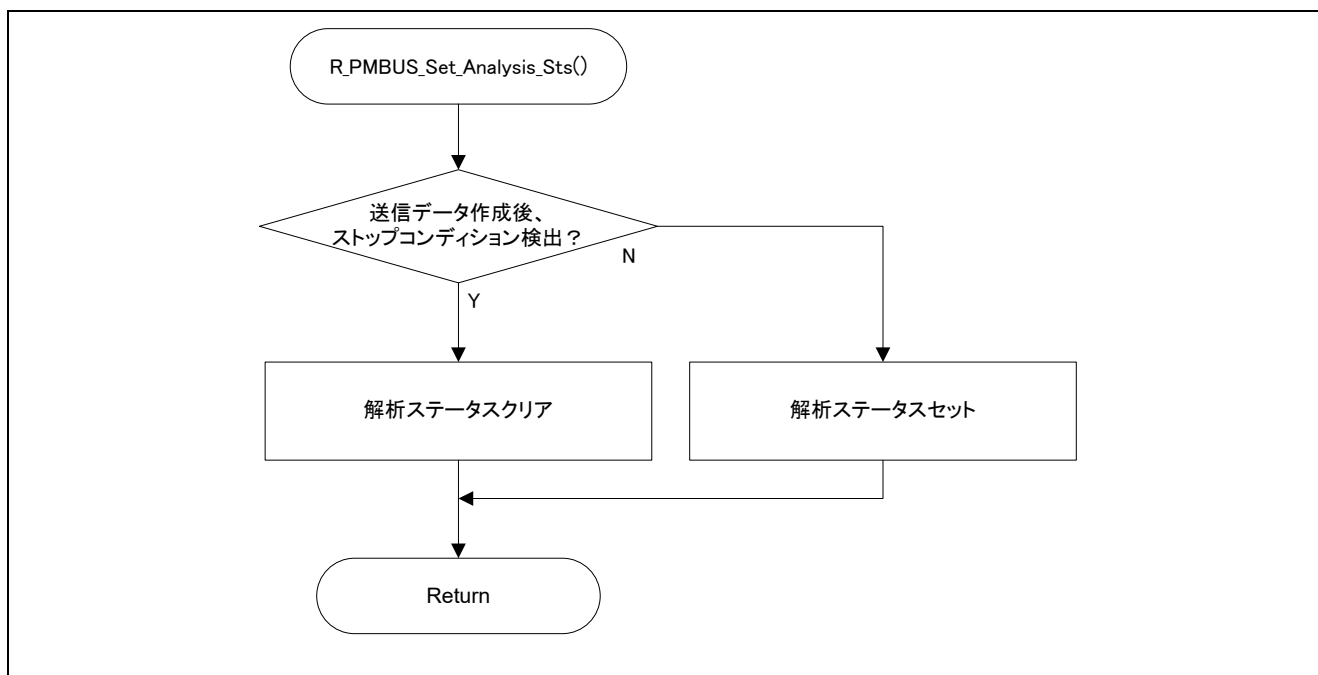


図 5.10 通信ステータス取得処理

### 5.5.11 タイムアウト動作制御処理

図 5.11 にタイムアウト動作制御処理のフローチャートを示します。

#### 【概要】

R\_PMBUS\_Timeout\_Enable()は引数により、PMBus の時間管理を行う処理の動作制御を行います。

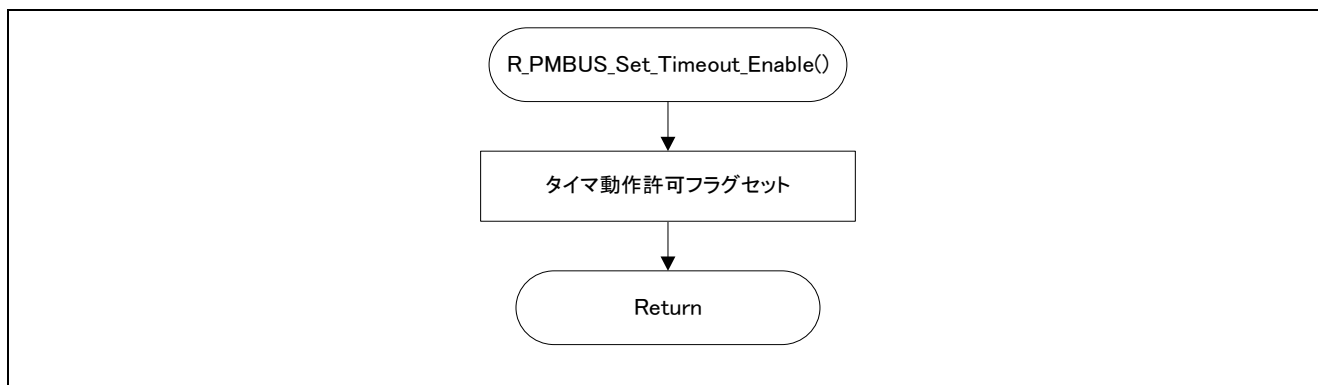


図 5.11 タイムアウト動作制御処理

## 5.5.12 タイムアウト時間生成処理

図 5.12 にタイムアウト時間生成処理のフローチャートを示します。

## 【概要】

R\_PMBUS\_Timeout\_Make()はPMBusのタイムアウト時間を生成しています。タイムアウト時間である25msを経過した場合はR\_IICA0\_Stop()でIICAを停止させ、R\_IICA0\_Create()で動作開始します。

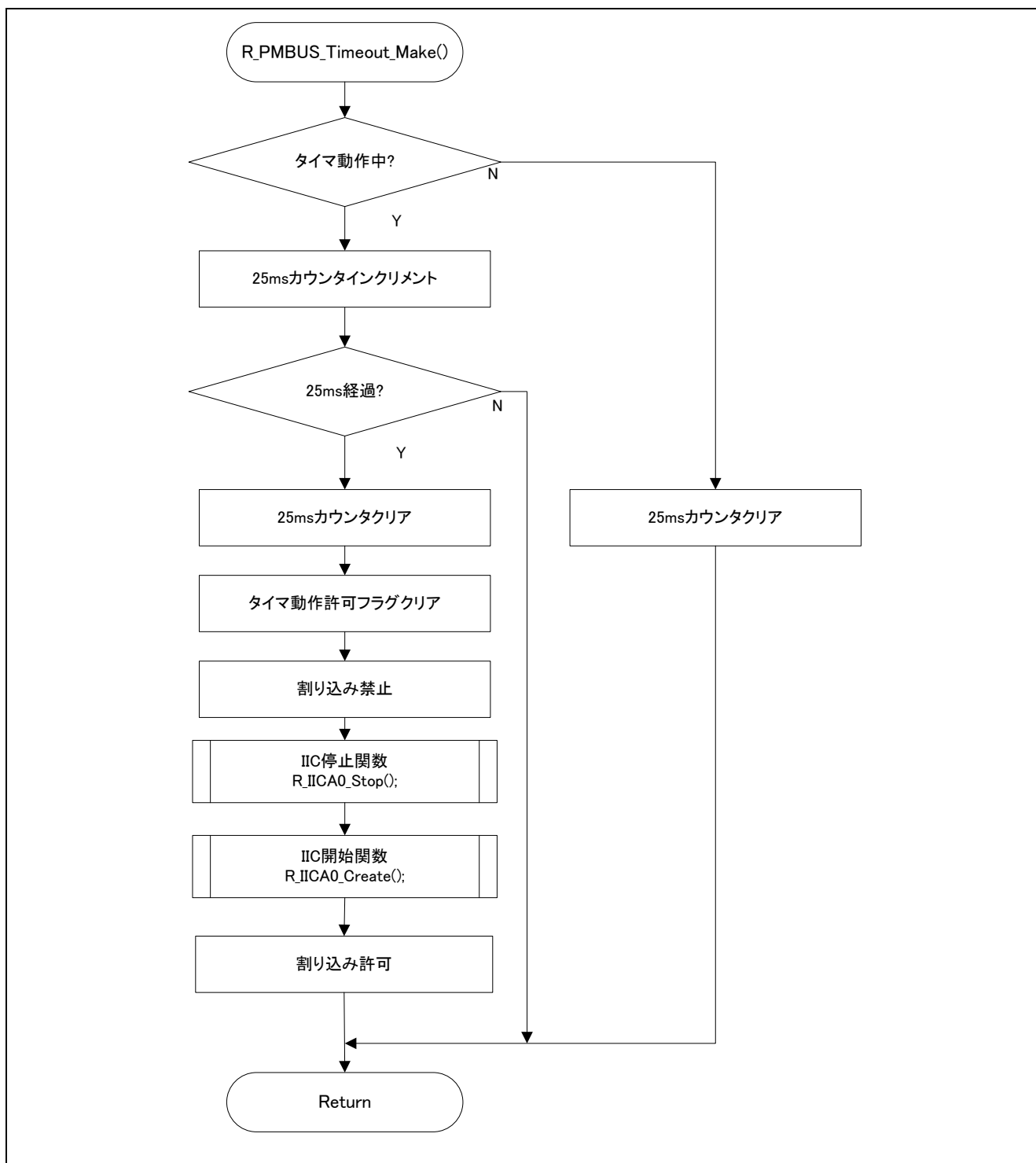


図 5.12 タイムアウト時間生成処理

## 5.5.13 CRC8 算出処理

図 5.13 に CRC8 算出処理のフローチャートを示します。

## 【概要】

r\_pmbus\_pec\_crc8()は通信データのCRC8を算出します。

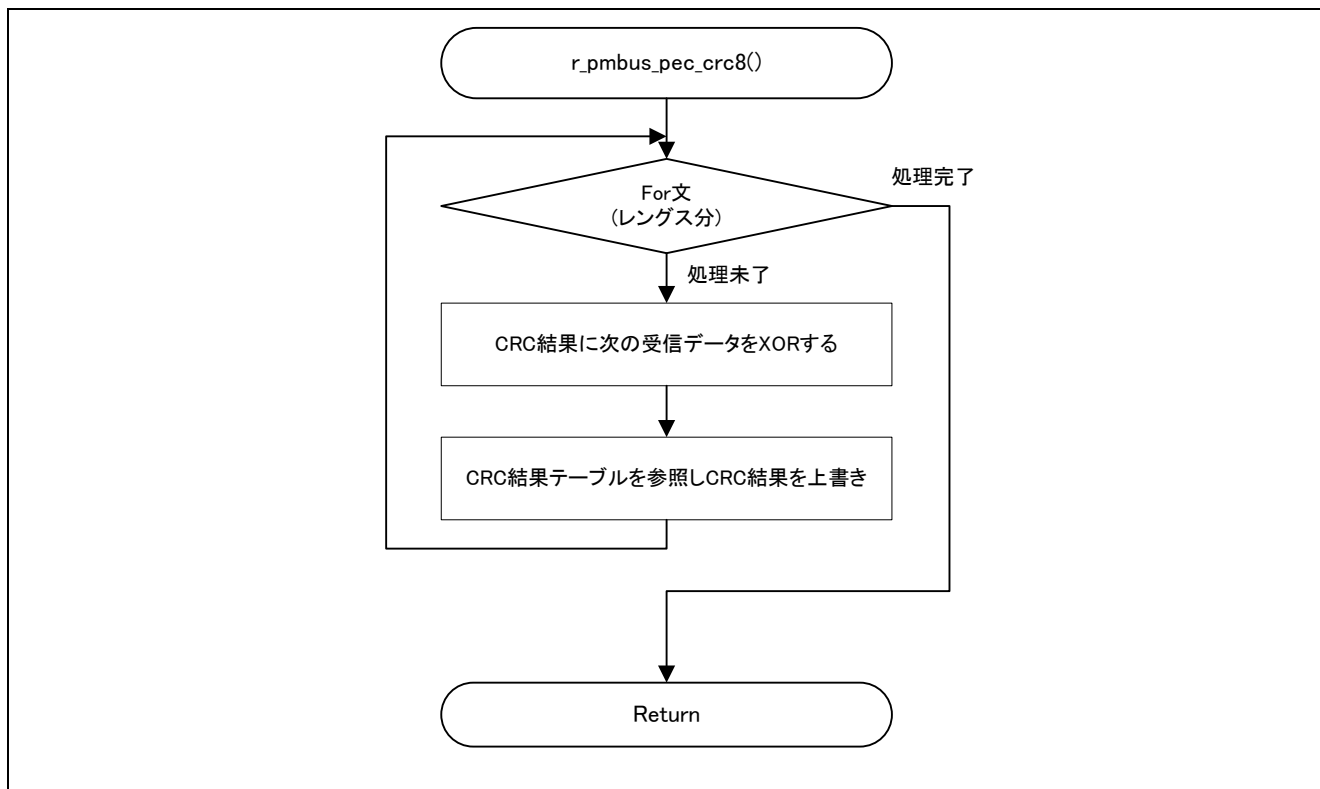


図 5.13 CRC8 算出処理



### 5.5.14 受信データ解析処理

図 5.14 に受信データ解析処理のフローチャートを示します。

**【概要】**

r\_pmbus\_receive\_check()は受信データを解析します。R\_IICA0\_Get\_Receivedata()で受信用バッファのアドレスを取得します。r\_pmbus\_pec\_crc8()でCRC8を算出します。

解析結果が異常の場合はR\_PMBUS\_Fault\_Handler()でエラー状態を設定します。

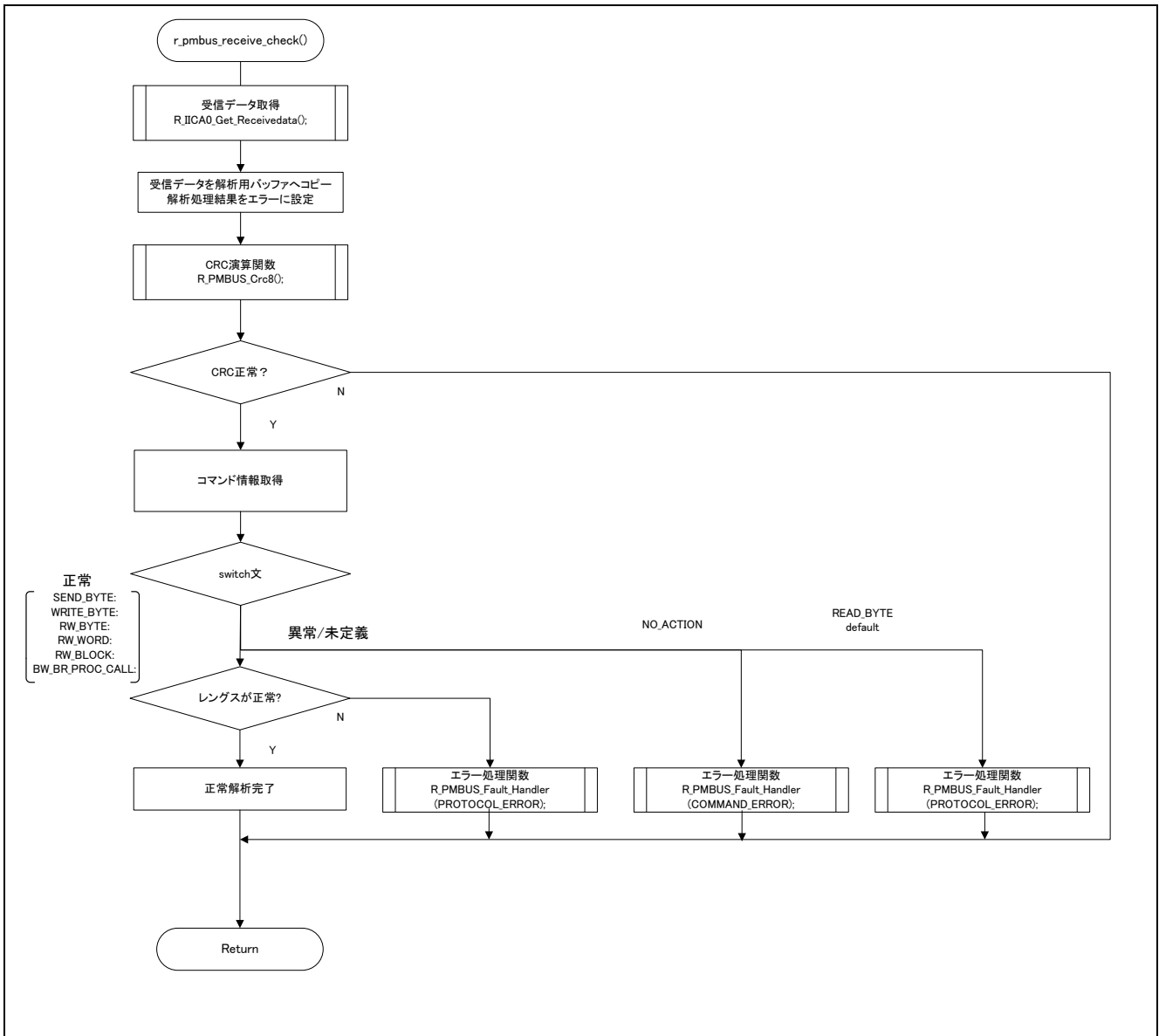


図 5.14 受信データ解析処理

### 5.5.15 コマンド解析処理

図 5.15 にコマンド解析処理のフローチャートを示します。

#### 【概要】

r\_pmbus\_command\_check()は受信コマンドを解析します。R\_IICA0\_Get\_Receivedata()で受信用バッファのアドレスを取得します。PMBusにて定められている各コマンドに対応したデータ数を取得します。その後、r\_pmbus\_pec\_crc8()でCRC8を算出し、算出結果を呼び出し元へ返します。

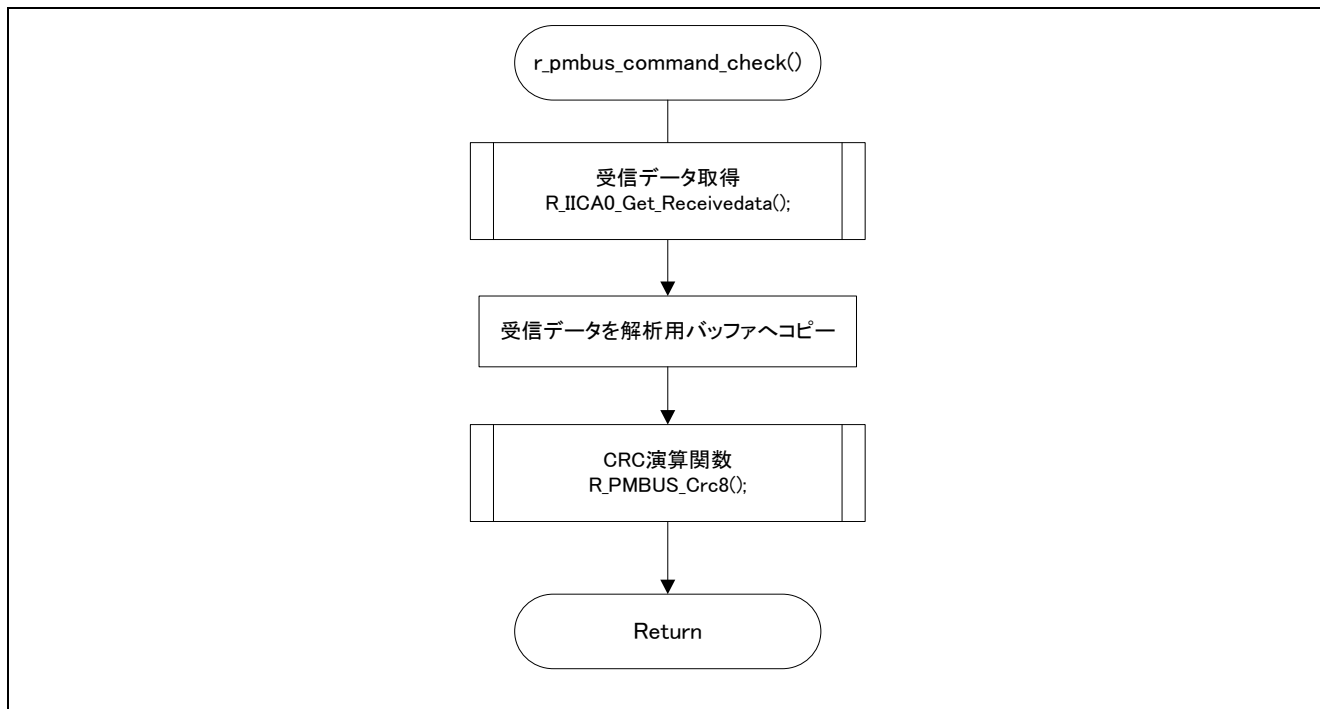


図 5.15 コマンド解析処理

5.5.16 送信データ作成処理

図 5.16 に送信データ作成処理のフローチャートを示します。

【概要】

r\_pmbus\_make\_senddata()は受信コマンドに応じた処理を行います。受信コマンドが正常の場合は送信データを取得します。その後、r\_pmbus\_pec\_crc8()でCRC8を算出し、R\_IICA0\_Set\_Senddata()で送信データを送信用バッファへコピーし送信開始します。

受信コマンドが未定義である場合は異常とみなしてR\_PMBUS\_Fault\_Handler()でエラー状態を設定します。

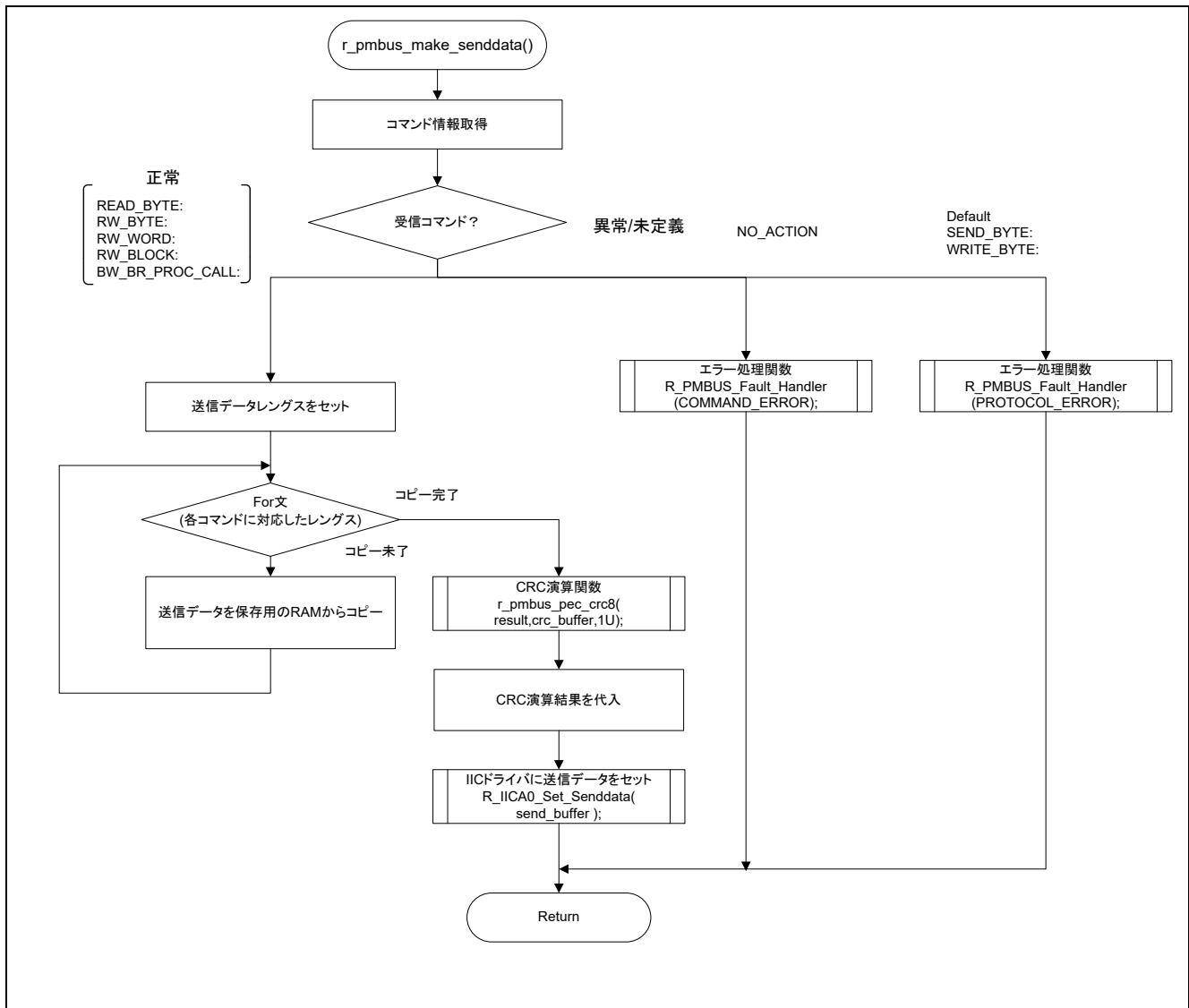


図 5.16 送信データ作成処理

## 5.5.17 受信データコピー処理

図 5.17 に受信データコピー処理のフローチャートを示します。

## 【概要】

r\_pmbus\_copy\_to\_ram()は受信データをデータ保存用の RAM へ保存します。

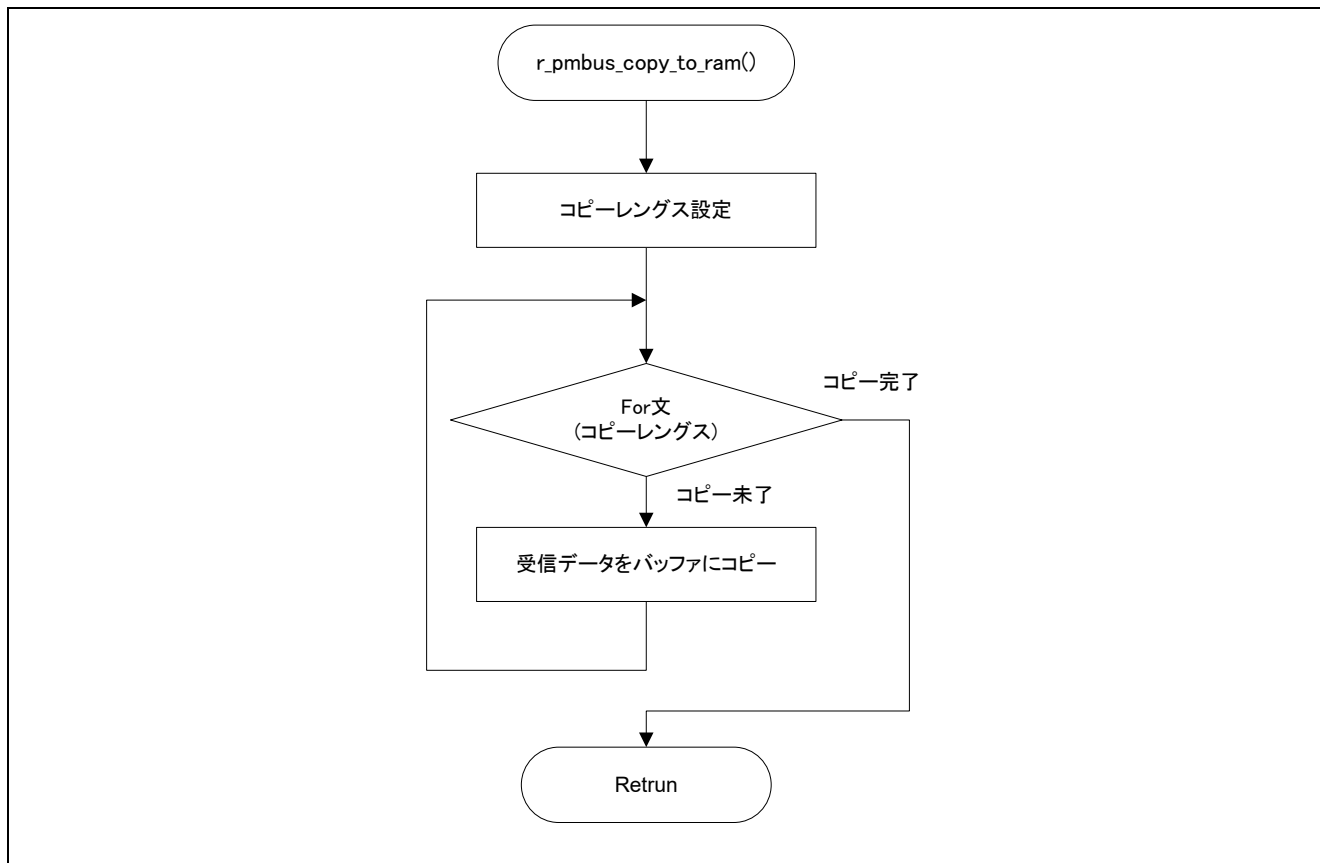


図 5.17 受信データコピー処理

### 5.5.18 通信ステータスエラー保存処理

図 5.18 に通信ステータスエラー保存処理のフローチャートを示します。

**【概要】**

R\_PMBUS\_Fault\_Handler()で各エラー状態を設定します。

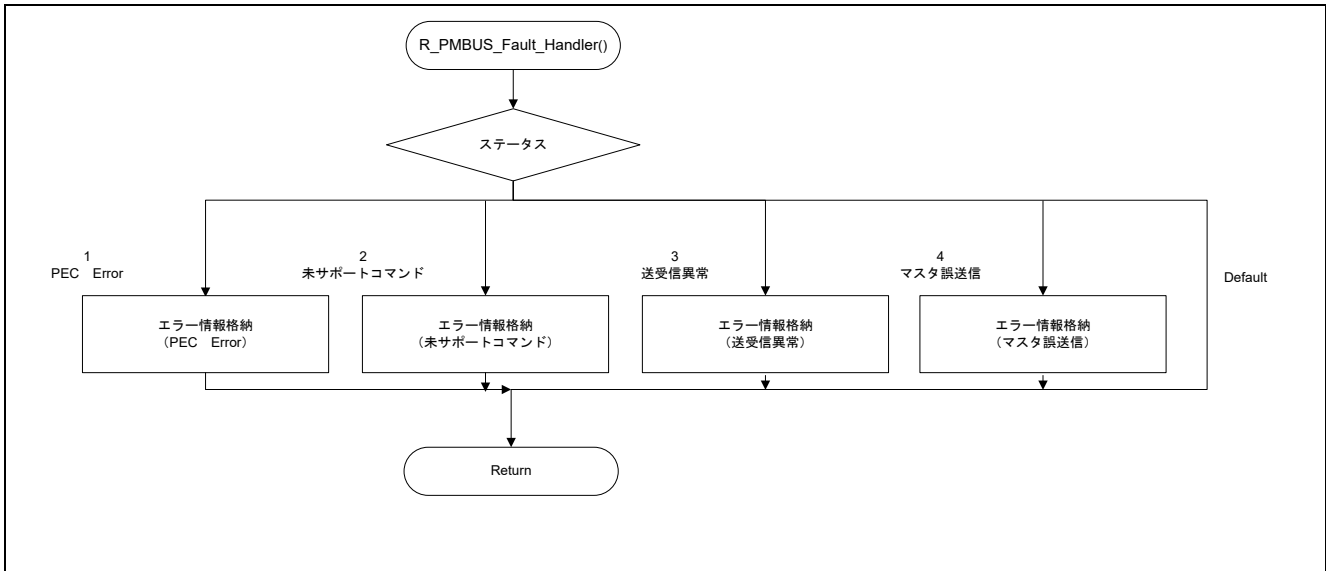


図 5.18 通信ステータスエラー保存処理

### 5.5.19 PMBus User 記述関数呼び出し処理

図 5.19 に PMBus User 記述関数呼び出し処理のフローチャートを示します。

**【概要】**

R\_PMBUS\_User\_App()は各コマンドに対応した処理を行う関数を呼び出します。

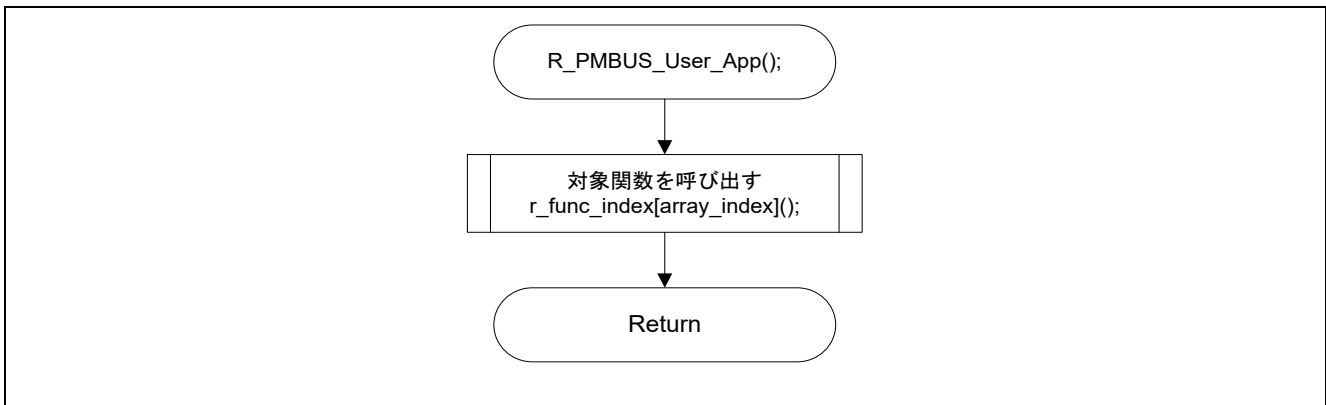


図 5.19 PMBus User 記述関数呼び出し処理

## 5.6 ROM/RAM サイズ

表 5.3 に本サンプルコードで使用する ROM/RAM サイズを記載します。スタックサイズは、見積もりサイズとなります。

表 5.3 ROM/RAM サイズ

環境	ROM サイズ	RAM サイズ	スタックサイズ
CA78K0R コンパイラ	4.726 Kbytes	0.826 Kbytes	0.063 Kbytes
CC-RL コンパイラ	4.446 Kbytes	0.438 Kbytes	0.039 Kbytes
IAR コンパイラ	3.869 Kbytes	0.696 Kbytes	0.063 Kbytes

## 6. サンプルコード

サンプルコードは、ルネサス エレクトロニクスの Web サイトから入手してください。

## 7. 参考ドキュメント

RL78/I1A ユーザーズマニュアル ハードウェア編 (R01UH0169)

RL78 ファミリー ユーザーズマニュアル ソフトウェア編 (R01US0015)

## 付録 A 定数一覧

表 A.1 にサンプルコードで使用する定数を示します。

表 A.1 サンプルコードで使用する定数（1/3）

定数名	設定値	内容
PEC_ERROR	0x01	Packet Error Check エラー（CRC エラー）
COMMAND_ERROR	0x02	未対応コマンド受信
PROTOCOL_ERROR	0x03	コマンドプロトコルエラー
MASTER_ERROR	0x04	マスタプロトコルエラー
SMBALERT_HIGH	0x01	SMBALERT#ポートの出力"High"設定
SMBALERT_LOW	0x00	SMBALERT#ポートの出力"Low"設定
UINT8_T_CLR	0x00	uint8_t 型の変数クリア用
UINT16_T_CLR	0x0000	uint16_t 型の変数クリア用
PEC_BYTE	0x01	Packet Error Check（1byte）領域用
PMBUS_RX_ADDRESS	0x00	アドレス格納場所（受信バッファ要素数）
PMBUS_RX_COMMAND	0x01	コマンド格納場所（受信バッファ要素数）
PMBUS_RX_DATA	0x02	データ格納場所（受信バッファ要素数）
PMBUS_TX_LENGTH	0x00	レングス格納場所（送信バッファ要素数）
PMBUS_TX_ADDRESS	0x01	アドレス格納場所（送信バッファ要素数）
PMBUS_TX_COMMAND	0x02	コマンド格納場所（送信バッファ要素数）
IICA0_LENGTH	0x00	レングス格納場所（受信バッファ要素数）
IICA0_ADDRESS	0x01	アドレス格納場所（受信バッファ要素数）
IICA0_COMMAND	0x02	コマンド格納場所（受信バッファ要素数）
DEFAULT_STS	0x00	通信ステータス（通信待機状態）
RECEIVE_STS	0x01	通信ステータス（受信状態）
SEND_STS	0x02	通信ステータス（送信状態）
SEND_ALERT_STS	0x03	通信ステータス（SMBALERT#異常状態）
SEND_COMP_STS	0x04	通信ステータス（送信完了状態）
TIMER_OFF	0x00	タイマ動作禁止状態
TIMER_ON	0x01	タイマ動作許可状態
OUTPUT_LOW	0x00	ポート出力"Low"設定
P0_NCH_ON	0x04	P0 N-ch オープンドレイン設定
P1_NCH_ON	0x03	P1 N-ch オープンドレイン設定
PMC02_CLR	0x00	PMC02 設定用
ADPC_DI_ON	0x01	ADPC 設定用
BIT_OUTPUT_MODE	0x00	ポート（1ビット）出力モード設定
BIT_INPUT_MODE	0x01	ポート（1ビット）入力モード設定
SMBALERT_PORT_P	P0.2	SMBALERT#ポートの設定
SMBALERT_PORT_PM	PM0.2	SMBALERT#ポートのポートモード設定
SMBALERT_PORT_PMC	PMC0.2	SMBALERT#ポートのポートモードコントロール設定
CONTROL_PORT_P	P2.0	CONTROL ポートの設定
CONTROL_PORT_PM	PM2.0	CONTROL ポートのポートモード設定
WRITE_PROTECT_PORT_P	P7.5	WRITE_PROTECT ポートの設定

注意 ポートを設定している定数は、ユーザーによって変更することが出来ます。

※PMBus 仕様書で定められている任意のハードウェア端子を示します。

表 A.1 サンプルコードで使用する定数 (2/3)

定数名	設定値	内容
WRITE_PROTECT_PORT_PM	PM7.5	WRITE_PROTECT ポートのポートモード設定
ADDRESS_PORT_P	全 ADDRESS ポート入力値 <sup>注1</sup>	全 ADDRESS ポート (1~7 ビット) の入力値取得用
ADDRESS_PORT_1_P	P20.0	ADDRESS ポートの設定 (1 ビット目)
ADDRESS_PORT_1_PM	PM20.0	ADDRESS ポートのポートモード設定 (1 ビット目)
ADDRESS_PORT_2_P	P20.1	ADDRESS ポートの設定 (2 ビット目)
ADDRESS_PORT_2_PM	PM20.1	ADDRESS ポートのポートモード設定 (2 ビット目)
ADDRESS_PORT_3_P	P20.2	ADDRESS ポートの設定 (3 ビット目)
ADDRESS_PORT_3_PM	PM20.2	ADDRESS ポートのポートモード設定 (3 ビット目)
ADDRESS_PORT_4_P	P20.3	ADDRESS ポートの設定 (4 ビット目)
ADDRESS_PORT_4_PM	PM20.3	ADDRESS ポートのポートモード設定 (4 ビット目)
ADDRESS_PORT_5_P	P20.4	ADDRESS ポートの設定 (5 ビット目)
ADDRESS_PORT_5_PM	PM20.4	ADDRESS ポートのポートモード設定 (5 ビット目)
ADDRESS_PORT_6_P	P20.5	ADDRESS ポートの設定 (6 ビット目)
ADDRESS_PORT_6_PM	PM20.5	ADDRESS ポートのポートモード設定 (6 ビット目)
ADDRESS_PORT_7_P	P20.6	ADDRESS ポートの設定 (7 ビット目)
ADDRESS_PORT_7_PM	PM20.6	ADDRESS ポートのポートモード設定 (7 ビット目)
TAU_TDR00_VALUE	0x7CFF	1ms 生成用コンペアレジスタ設定値
TAU_TDR00_START	0x0001	タイマ・アレイ・ユニット動作許可設定
TAU_TDR00_STOP	0x0001	タイマ・アレイ・ユニット動作停止設定
TMIF00_REQUEST	0x01	1ms 経過要求比較用
TMIF00_CLR	0x00	TMIF00 フラグクリア用
SEND_BYTE	0x00	通信プロトコル (SEND_BYTE)
READ_BYTE	0x01	通信プロトコル (READ_BYTE)
WRITE_BYTE	0x02	通信プロトコル (WRITE_BYTE)
RW_BYTE	0x03	通信プロトコル (RW_BYTE)
READ_WORD	0x04	通信プロトコル (READ_WORD)
RW_WORD	0x05	通信プロトコル (RW_WORD)
RW_BLOCK	0x06	通信プロトコル (RW_BLOCK)
BW_BR_PROC_CALL	0x07	通信プロトコル (BW_BR_PROC_CALL)
NO_ACTION	0xFF	通信プロトコル (NO_ACTION)
VARIABLE	0x02	コマンドデータ長 (User 依存のコマンド)
VARIABLE_2BYTES_DATA	0x03	コマンドデータ長 (User 依存のコマンド)
VARIABLE_3BYTES_DATA	0x04	コマンドデータ長 (User 依存のコマンド)
VARIABLE_4BYTES_DATA	0x05	コマンドデータ長 (User 依存のコマンド)
VARIABLE_5BYTES_DATA	0x06	コマンドデータ長 (User 依存のコマンド)
VARIABLE_6BYTES_DATA	0x07	コマンドデータ長 (User 依存のコマンド)
VARIABLE_7BYTES_DATA	0x08	コマンドデータ長 (User 依存のコマンド)
VARIABLE_8BYTES_DATA	0x09	コマンドデータ長 (User 依存のコマンド)
MFR_DEFINED	0x01	コマンドデータ長 (User 依存のコマンド)
PEC_ERROR_SET_CML	0x20	CML コマンドデータセット用 (PEC_ERROR 情報)

注 ADDRESS\_PORT\_P は、ADDRESS\_PORT\_1\_P~ADDRESS\_PORT\_7\_P の入力値を 1byte のデータに変換し、自局アドレスとして設定出来る値となります。

注意 ポートを設定している定数は、ユーザーによって変更することが出来ます。  
※PMBus 仕様書で定められている任意のハードウェア端子を示します。



表 A.1 サンプルコードで使用する定数 (3/3)

定数名	設定値	内容
COMMAND_ERROR_SET_CML	0x80	CML コマンドデータセット用 (COMMAND_ERROR 情報)
PROTOCOL_ERROR_SET_CML	0x40	CML コマンドデータセット用 (PROTOCOL_ERROR 情報)
MASTER_ERROR_SET_CML	0x02	CML コマンドデータセット用 (PEC_ERROR 情報)
CML_ERROR_SET_STATUS_WORD	0x02	STATUS_WORD コマンドデータセット用 (CML コマンド情報)
PMBUS_RX_BUFF_MAX	0x24	PMBus Control Module 用受信バッファの要素数
PMBUS_TX_BUFF_MAX	0x24	PMBus Control Module 用送信バッファの要素数
COPY_TO_RAM_ENABLE	0x01	RAM へコピー許可
COPY_TO_RAM_DISABLE	0x00	RAM へコピー禁止
TIMER_COUNT_ENABLE	0x01	タイマ動作許可
TIMER_COUNT_DISABLE	0x00	タイマ動作禁止
TIMEOUT_COMPARE	0x19	タイムアウト時間経過 (25ms)
ALL_COMMAND	0x0100	コマンドデータ初期化用 (256 コマンド)
EXCEPT_DATA_BYTE	0x03	データバイト除外用
ADD_COUNT	0x01	バッファの要素数加算用
CHECK_NG	0x00	データ確認結果が異常
CHECK_OK	0x01	データ確認結果が正常
DATA_BYTE_LOW	0x00	異常情報セット位置格納用 (1ワードの Low バイト)
DATA_BYTE_HIGH	0x01	異常情報セット位置格納用 (1ワードの High バイト)
DATA_BYTE	0x00	異常情報セット位置格納用 (1バイト)

## 付録 B 変数一覧

表 B.1 にグローバル変数を示します。

表 B.1 グローバル変数

Type	Variable Name	Contents	Function Used
無し	-	-	-

## 付録 C 関数仕様

サンプルコードの関数仕様を示します。

## R\_TAU0\_Create

---

概要	タイマ・アレイ・ユニット初期化処理
ヘッダ	r_timer_module.h
宣言	void R_TAU0_Create(void)
説明	タイマ・アレイ・ユニットのチャンネル0をインターバルタイマ（1ms）と使用する為の初期設定を行います。
引数	なし
リターン値	なし
備考	なし

## R\_TAU0\_Channel0\_Start

---

概要	タイマ・アレイ・ユニット動作開始処理
ヘッダ	r_timer_module.h
宣言	void R_TAU0_Channel0_Start(void)
説明	タイマ・アレイ・ユニットのチャンネル0の動作を開始します。
引数	なし
リターン値	なし
備考	なし

## R\_TAU0\_Channel0\_Stop

---

概要	タイマ・アレイ・ユニット動作停止処理
ヘッダ	r_timer_module.h
宣言	void R_TAU0_Channel0_Stop(void)
説明	タイマ・アレイ・ユニットのチャンネル0の動作を停止します。
引数	なし
リターン値	なし
備考	なし

## R\_TIMER\_Control

---

概要	1ms 基準タイマ生成処理
ヘッダ	r_timer_module.h
宣言	void R_TIMER_Control(void)
説明	1ms の基準時間を生成し、TMIF00 フラグをクリアします。
引数	なし
リターン値	なし
備考	なし

---

**R\_PORT\_Create**

---

概要	ポート初期化処理
ヘッダ	r_port_module.h
宣言	void R_PORT_Create(void)
説明	使用ポートの初期設定を行います。
引数	なし
リターン値	なし
備考	なし

---

**R\_PORT\_Smbalert**

---

概要	SMBALERT#ポート出力設定処理
ヘッダ	r_port_module.h
宣言	void R_PORT_Smbalert(uint8_t)
説明	SMBALERT#ポートの出力設定を行います。
引数	level_set                      SMBALERT#ポート出力レベル
リターン値	なし
備考	本サンプルコードでは、PMBus の仕様で定められている任意の周辺 I/O 用に関数を用意しています。使用するポートは、r_port_module.h 内の定数に任意のポートを設定することで使用することが出来ます。

---

**R\_PORT\_Get\_Address**

---

概要	ADDRESS ポート入力値取得処理
ヘッダ	r_port_module.h
宣言	uint8_t R_PORT_Get_Address(void)
説明	自局アドレスに設定する値を ADDRESS ポートの入力から取得します。
引数	なし
リターン値	address_result                  入力値（自局アドレス設定値）
備考	本サンプルコードでは、PMBus の仕様で定められている任意の周辺 I/O 用に関数を用意しています。使用するポートは、r_port_module.h 内の定数に任意のポートを設定することで使用することが出来ます。

---

**R\_PORT\_Get\_Control**

---

概要	CONTROL ポート入力値取得処理
ヘッダ	r_port_module.h
宣言	uint8_t R_PORT_Get_Control(void)
説明	CONTROL ポートの入力値を取得します。
引数	なし
リターン値	control_result                  入力値（CONTROL ポート設定値）
備考	本サンプルコードでは、PMBus の仕様で定められている任意の周辺 I/O 用に関数を用意しています。使用するポートは、r_port_module.h 内の定数に任意のポートを設定することで使用することが出来ます。

---

**R\_PORT\_Get\_Write\_Protect**

---

概要	WRITE_PROTECT ポート入力値取得処理
ヘッダ	r_port_module.h
宣言	uint8_t R_PORT_Get_Write_Protect(void)
説明	WRITE_PROTECT ポートの入力値を取得します。
引数	なし
リターン値	write_result                      入力値（WRITE_PROTECT ポート設定値）
備考	本サンプルコードでは、PMBus の仕様で定められている任意の周辺 I/O 用に関数を用意しています。使用するポートは、r_port_module.h 内の定数に任意のポートを設定することで使用することが出来ます。

---

**R\_IICA0\_Create**

---

概要	シリアル・インタフェース IICA 初期化処理
ヘッダ	r_iica0_driver.h
宣言	void R_IICA0_Create(void)
説明	シリアル・インタフェース IICA の初期化処理を行います。
引数	なし
リターン値	なし
備考	本サンプルコードでは、PMBus の仕様で定められている任意の周辺 I/O 用に関数を用意しています。使用するポートは、R_IICA0_Create 関数内のコンパイルスイッチを切り替えることで、自局アドレスの設定を周辺 I/O かソフトに変更することが出来ます。また、転送レートも同様に 100kbps、400kbps に切り替えることが出来ます。

---

**R\_IICA0\_Stop**

---

概要	シリアル・インタフェース IICA 停止処理
ヘッダ	r_iica0_driver.h
宣言	void R_IICA0_Stop(void)
説明	シリアル・インタフェース IICA の停止処理を行います。
引数	なし
リターン値	なし
備考	なし

---

**R\_IICA0\_Init**

---

概要	IICA0 Driver Module 初期化処理
ヘッダ	r_iica0_driver.h
宣言	void R_IICA0_Init(void)
説明	IICA0 Driver Module の初期化処理を行います。（変数初期化）
引数	なし
リターン値	なし
備考	なし

---

**R\_PMBUS\_Init**

---

概要	PMBus Control Module 初期化処理
ヘッダ	r_pmbus_control.h
宣言	void R_PMBUS_Init(void)
説明	PMBus Control Module の初期化を行います。(変数初期化)
引数	なし
リターン値	なし
備考	なし

## 付録 D PMBus コマンド

本サンプルコードは、PMBus™（Power Management Bus）の仕様で定められている全コマンドに対応しています。表 D.1 にコマンド一覧を示します。

表 D.1 コマンド一覧（1/6）

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes
00h	PAGE	R/W Byte	1
01h	OPERATION	R/W Byte	1
02h	ON_OFF_CONFIG	R/W Byte	1
03h	CLEAR_FAULTS	Send Byte	0
04h	Reserved	-	-
05h	Reserved	-	-
06h	Reserved	-	-
07h	Reserved	-	-
08h	Reserved	-	-
09h	Reserved	-	-
0Ah	Reserved	-	-
0Bh	Reserved	-	-
0Ch	Reserved	-	-
0Dh	Reserved	-	-
0Eh	Reserved	-	-
0Fh	Reserved	-	-
10h	WRITE_PROTECT	R/W Byte	1
11h	STORE_DEFAULT_ALL	Send Byte	0
12h	RESTORE_DEFAULT_ALL	Write Byte	1
13h	STORE_DEFAULT_CODE	Send Byte	0
14h	RESTORE_DEFAULT_CODE	Write Byte	1
15h	STORE_USER_ALL	Send Byte	0
16h	RESTORE_USER_ALL	Write Byte	1
17h	STORE_USER_CODE	Send Byte	0
18h	RESTORE_USER_CODE	Write Byte	1
19h	Reserved	-	-
1Ah	Reserved	-	-
1Bh	Reserved	-	-
1Ch	Reserved	-	-
1Dh	Reserved	-	-
1Eh	Reserved	-	-
1Fh	Reserved	-	-
20h	VOUT_MODE	R/W Byte	1
21h	VOUT_COMMAND	R/W Word	2
22h	VOUT_TRIM	R/W Word	2
23h	VOUT_CAL	R/W Word	2
24h	VOUT_MAX	R/W Word	2
25h	VOUT_MARGIN_HIGH	R/W Word	2
26h	VOUT_MARGIN_LOW	R/W Word	2
27h	VOUT_TRANSITION_RATE	R/W Word	2

表 D.1 コマンド一覧 (2/6)

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes
28h	VOUT_DROOP	R/W Word	2
29h	VOLTAGE_SCALE_LOOP	R/W Word	2
2Ah	VOLTAGE_SCALE_MONITOR	R/W Word	2
2Bh	Reserved	-	-
2Ch	Reserved	-	-
2Dh	Reserved	-	-
2Eh	Reserved	-	-
2Fh	Reserved	-	-
30h	COEFFICIENTS	Block R/W Process Call	6 (Includes Byte Count)
31h	POUT_MAX	R/W Word	2
32h	MAX_DUTY	R/W Word	2
33h	FREQUENCY_SWITCH	R/W Word	2
34h	Reserved	-	-
35h	VIN_ON	R/W Word	2
36h	VIN_OFF	R/W Word	2
37h	INTERLEAVE	R/W Word	2
38h	IOUT_SCALE	R/W Word	2
39h	IOUT_CAL_OFFSET	R/W Word	2
3Ah	VFAN_1	R/W Word	2
3Bh	VFAN_2	R/W Word	2
3Ch	Reserved	-	-
3Dh	Reserved	-	-
3Eh	Reserved	-	-
3Fh	Reserved	-	-
40h	VOUT_OV_FAULT_LIMIT	R/W Word	2
41h	VOUT_OV_FAULT_RESPONSE	R/W Byte	1
42h	VOUT_OV_WARN_LIMIT	R/W Word	2
43h	VOUT_UV_WARN_LIMIT	R/W Word	2
44h	VOUT_UV_FAULT_LIMIT	R/W Word	2
45h	VOUT_UV_FAULT_RESPONSE	R/W Byte	1
46h	IOUT_OC_FAULT_LIMIT	R/W Word	2
47h	IOUT_OC_FAULT_RESPONSE	R/W Byte	1
48h	IOUT_OC_LV_FAULT_LIMIT	R/W Word	2
49h	IOUT_OC_LV_FAULT_RESPONSE	R/W Byte	1
4Ah	IOUT_OC_WARN_LIMIT	R/W Word	2
4Bh	IOUT_UC_FAULT_LIMIT	R/W Word	2
4Ch	IOUT_UC_FAULT_RESPONSE	R/W Byte	1
4Dh	Reserved For POUT_FAULT_LIMIT	R/W Word	2
4Eh	Reserved For: POUT_MAX_FAULT_RESPONSE	R/W Byte	1
4Fh	OT_FAULT_LIMIT	R/W Word	2
50h	OT_FAULT_RESPONSE	R/W Byte	1
51h	OT_WARN_LIMIT	R/W Word	2
52h	UT_WARN_LIMIT	R/W Word	2
53h	UT_FAULT_LIMIT	R/W Word	2



表 D.1 コマンド一覧 (3/6)

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes
54h	UT_FAULT_RESPONSE	R/W Byte	1
55h	VIN_OV_FAULT_LIMIT	R/W Word	2
56h	VIN_OV_FAULT_RESPONSE	R/W Byte	1
57h	VIN_OV_WARN_LIMIT	R/W Word	2
58h	VIN_UV_WARN_LIMIT	R/W Word	2
59h	VIN_UV_FAULT_LIMIT	R/W Word	2
5Ah	VIN_UV_FAULT_RESPONSE	R/W Byte	1
5Bh	IIN_OC_FAULT_LIMIT	R/W Word	2
5Ch	IIN_OC_FAULT_RESPONSE	R/W Byte	1
5Dh	IIN_OC_WARN_LIMIT	R/W Word	2
5Eh	POWER_GOOD_ON	R/W Word	2
5Fh	POWER_GOOD_OFF	R/W Word	2
60h	TON_DELAY	R/W Word	2
61h	TON_RISE	R/W Word	2
62h	TON_MAX_FAULT_LIMIT	R/W Word	2
63h	TON_MAX_FAULT_RESPONSE	R/W Byte	1
64h	TOFF_DELAY	R/W Word	2
65h	TOFF_FALL	R/W Word	2
66h	TOFF_MAX_FAULT_LIMIT	R/W Word	2
67h	TOFF_MAX_FAULT_RESPONSE	R/W Byte	1
68h	Reserved	-	-
69h	Reserved	-	-
6Ah	Reserved	-	-
6Bh	Reserved	-	-
6Ch	Reserved	-	-
6Dh	Reserved	-	-
6Eh	Reserved	-	-
6Fh	Reserved	-	-
70h	Reserved (Test Input Fuse A)	-	-
71h	Reserved (Test Input Fuse B)	-	-
72h	Reserved (Test Input OR-ing A)	-	-
73h	Reserved (Test Input OR-ing B)	-	-
74h	Reserved (Test Output OR-ing)	-	-
75h	Reserved	-	-
76h	Reserved	-	-
77h	Reserved	-	-
78h	STATUS_BYTE	Read Byte	1
79h	STATUS_WORD	Read Word	2
7Ah	STATUS_VOUT	Read Byte	1
7Bh	STATUS_IOUT	Read Byte	1
7Ch	STATUS_INPUT	Read Byte	1
7Dh	STATUS_TEMPERATURE	Read Byte	1
7Eh	STATUS_CML	Read Byte	1
7Fh	STATUS_OTHER	Read Byte	1
80h	STATUS_MFR_SPECIFIC	Read Byte	1
81h	Reserved	-	-
82h	Reserved	-	-

表 D.1 コマンド一覧 (4/6)

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes
83h	Reserved	-	-
84h	Reserved	-	-
85h	Reserved	-	-
86h	Reserved	-	-
87h	Reserved	-	-
88h	READ_VIN	Read Word	2
89h	READ_IIN	Read Word	2
8Ah	READ_VCAP	Read Word	2
8Bh	READ_VOUT	Read Word	2
8Ch	READ_IOUT	Read Word	2
8Dh	READ_TEMPERATURE_1	Read Word	2
8Eh	READ_TEMPERATURE_2	Read Word	2
8Fh	READ_TEMPERATURE_3	Read Word	2
90h	READ_FAN SPEED_1	Read Word	2
91h	READ_FAN SPEED_2	Read Word	2
92h	READ_VFAN_1	Read Word	2
93h	READ_VFAN_2	Read Word	2
94h	READ_DUTY_CYCLE	Read Word	2
95h	READ_FREQUENCY	Read Word	2
96h	Reserved	-	-
97h	Reserved	-	-
98h	PMBUS_REVISION	Read Byte	1
99h	MFR_ID	R/W Block	Variable
9Ah	MFR_MODEL	R/W Block	Variable
9Bh	MFR_REVISION	R/W Block	Variable
9Ch	MFR_LOCATION	R/W Block	Variable
9Dh	MFR_DATE	R/W Block	Variable
9Eh	MFR_SERIAL	R/W Block	Variable
9Fh	Reserved	-	-
A0h	MFR_VIN_MIN	Read Word	2
A1h	MFR_VIN_MAX	Read Word	2
A2h	MFR_IIN_MAX	Read Word	2
A3h	MFR_PIN_MAX	Read Word	2
A4h	MFR_VOUT_MIN	Read Word	2
A5h	MFR_VOUT_MAX	Read Word	2
A6h	MFR_IOUT_MAX	Read Word	2
A7h	MFR_POUT_MAX	Read Word	2
A8h	MFR_TAMBIENT_MAX	Read Word	2
A9h	MFR_TAMBIENT_MIN	Read Word	2
AAh	Reserved	-	-
ABh	Reserved	-	-
ACH	Reserved	-	-
ADh	Reserved	-	-
A Eh	Reserved	-	-
AFh	Reserved	-	-
B0h	USER_DATA_00	Block R/W	Variable
B1h	USER_DATA_01	Block R/W	Variable

表 D.1 コマンド一覧 (5/6)

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes
B2h	USER_DATA_02	Block R/W	Variable
B3h	USER_DATA_03	Block R/W	Variable
B4h	USER_DATA_04	Block R/W	Variable
B5h	USER_DATA_05	Block R/W	Variable
B6h	USER_DATA_06	Block R/W	Variable
B7h	USER_DATA_07	Block R/W	Variable
B8h	USER_DATA_08	Block R/W	Variable
B9h	USER_DATA_09	Block R/W	Variable
BAh	USER_DATA_10	Block R/W	Variable
BBh	USER_DATA_11	Block R/W	Variable
BCh	USER_DATA_12	Block R/W	Variable
BDh	USER_DATA_13	Block R/W	Variable
BEh	USER_DATA_14	Block R/W	Variable
BFh	USER_DATA_15	Block R/W	Variable
C0h	Reserved	-	-
C1h	Reserved	-	-
C2h	Reserved	-	-
C3h	Reserved	-	-
C4h	Reserved	-	-
C5h	Reserved	-	-
C6h	Reserved	-	-
C7h	Reserved	-	-
C8h	Reserved	-	-
C9h	Reserved	-	-
CAh	Reserved	-	-
CBh	Reserved	-	-
CCh	Reserved	-	-
CDh	Reserved	-	-
CEh	Reserved	-	-
CFh	Reserved	-	-
D0h	MFR_SPECIFIC_00	Mfr. Defined	Mfr. Defined
D1h	MFR_SPECIFIC_01	Mfr. Defined	Mfr. Defined
D2h	MFR_SPECIFIC_02	Mfr. Defined	Mfr. Defined
D3h	MFR_SPECIFIC_03	Mfr. Defined	Mfr. Defined
D4h	MFR_SPECIFIC_04	Mfr. Defined	Mfr. Defined
D5h	MFR_SPECIFIC_05	Mfr. Defined	Mfr. Defined
D6h	MFR_SPECIFIC_06	Mfr. Defined	Mfr. Defined
D7h	MFR_SPECIFIC_07	Mfr. Defined	Mfr. Defined
D8h	MFR_SPECIFIC_08	Mfr. Defined	Mfr. Defined
D9h	MFR_SPECIFIC_09	Mfr. Defined	Mfr. Defined
DAh	MFR_SPECIFIC_10	Mfr. Defined	Mfr. Defined
DBh	MFR_SPECIFIC_11	Mfr. Defined	Mfr. Defined
DCh	MFR_SPECIFIC_12	Mfr. Defined	Mfr. Defined
DDh	MFR_SPECIFIC_13	Mfr. Defined	Mfr. Defined
DEh	MFR_SPECIFIC_14	Mfr. Defined	Mfr. Defined
DFh	MFR_SPECIFIC_15	Mfr. Defined	Mfr. Defined
E0h	MFR_SPECIFIC_16	Mfr. Defined	Mfr. Defined

表 D.1 コマンド一覧 (6/6)

Command Code	Command Name	SMBus Transaction Type	Number Of Data Bytes
E1h	MFR_SPECIFIC_17	Mfr.Defined	Mfr.Defined
E2h	MFR_SPECIFIC_18	Mfr.Defined	Mfr.Defined
E3h	MFR_SPECIFIC_19	Mfr.Defined	Mfr.Defined
E4h	MFR_SPECIFIC_20	Mfr.Defined	Mfr.Defined
E5h	MFR_SPECIFIC_21	Mfr.Defined	Mfr.Defined
E6h	MFR_SPECIFIC_22	Mfr.Defined	Mfr.Defined
E7h	MFR_SPECIFIC_23	Mfr.Defined	Mfr.Defined
E8h	MFR_SPECIFIC_24	Mfr.Defined	Mfr.Defined
E9h	MFR_SPECIFIC_25	Mfr.Defined	Mfr.Defined
EAh	MFR_SPECIFIC_26	Mfr.Defined	Mfr.Defined
EBh	MFR_SPECIFIC_27	Mfr.Defined	Mfr.Defined
ECh	MFR_SPECIFIC_28	Mfr.Defined	Mfr.Defined
EDh	MFR_SPECIFIC_29	Mfr.Defined	Mfr.Defined
EEh	MFR_SPECIFIC_30	Mfr.Defined	Mfr.Defined
EFh	MFR_SPECIFIC_31	Mfr.Defined	Mfr.Defined
F0h	MFR_SPECIFIC_32	Mfr.Defined	Mfr.Defined
F1h	MFR_SPECIFIC_33	Mfr.Defined	Mfr.Defined
F2h	MFR_SPECIFIC_34	Mfr.Defined	Mfr.Defined
F3h	MFR_SPECIFIC_35	Mfr.Defined	Mfr.Defined
F4h	MFR_SPECIFIC_36	Mfr.Defined	Mfr.Defined
F5h	MFR_SPECIFIC_37	Mfr.Defined	Mfr.Defined
F6h	MFR_SPECIFIC_38	Mfr.Defined	Mfr.Defined
F7h	MFR_SPECIFIC_39	Mfr.Defined	Mfr.Defined
F8h	MFR_SPECIFIC_40	Mfr.Defined	Mfr.Defined
F9h	MFR_SPECIFIC_41	Mfr.Defined	Mfr.Defined
FAh	MFR_SPECIFIC_42	Mfr.Defined	Mfr.Defined
FBh	MFR_SPECIFIC_43	Mfr.Defined	Mfr.Defined
FCh	MFR_SPECIFIC_44	Mfr.Defined	Mfr.Defined
FDh	MFR_SPECIFIC_45	Mfr.Defined	Mfr.Defined
FEh	MFR_SPECIFIC_COMMAND_EXT	Extended Command	1 Or 2
FFh	PMBUS_COMMAND_EXT	Extended Command	Or 2

## 付録 E 対応コマンドの追加、削除方法

本サンプルコードは、ユーザーによって対応するコマンドの追加、削除が可能です。

以下に追加、削除方法を示します。

### ■追加、削除方法

手順 1 : r\_pmbus\_control.c ファイルの r\_matrix[256]の変更。

r\_matrix[256]には、各コマンドの情報を格納しています。

<コマンド情報>

- 通信プロトコル
- コマンドデータ長
- コマンドデータ格納領域のアドレス

対応コマンドの追加、削除を行う場合は、各コマンドの情報を変更する必要があります。

また、対応コマンドを追加する場合は、コマンドデータ格納用のバッファを確保する必要があります。

例) 0x04 にコマンド追加する場合

#### 【変更前】

```

/*****
PMBus Protocol Table (0x00-0xFF)
*****/

static const command r_matrix[256] = {
/* Protocol-----read/write bytes---Command_ram_address-----Command No */
{ RW_BYTE, 0x1, r_pmbus_page }, /* 0x0 */
{ RW_BYTE, 0x1, r_pmbus_opration }, /* 0x1 */
{ RW_BYTE, 0x1, r_pmbus_on_off_config }, /* 0x2 */
{ SEND_BYTE, 0x0, 0x0 }, /* 0x3 */
{ NO_ACTION, 0x0, r_pmbus_dummy }, /* 0x4 */
.
};
    
```

#### 【変更後】

```

static uint8_t r_pmbus_user[1]; /* 0x4 */
/*****
PMBus Protocol Table (0x00-0xFF)
*****/

static const command r_matrix[256] = {
/* Protocol-----read/write bytes---Command_ram_address-----Command No */
{ RW_BYTE, 0x1, r_pmbus_page }, /* 0x0 */
{ RW_BYTE, 0x1, r_pmbus_opration }, /* 0x1 */
{ RW_BYTE, 0x1, r_pmbus_on_off_config }, /* 0x2 */
{ SEND_BYTE, 0x0, 0x0 }, /* 0x3 */
{ RW_BYTE, 0x1, r_pmbus_user }, /* 0x4 */
.
};
    
```

追加するコマンドのデータを格納するバッファを宣言します。

追加するコマンドのデータを格納するバッファのアドレスを記載  
※バッファは User が宣言します。

追加するコマンドの通信プロトコル追加

追加するコマンドのデータ長を追加

例) 0x01 のコマンドを削除する場合

【変更前】

```

/*****
PMBus Protocol Table (0x00-0xFF)
*****/

static const command r_matrix[256] = {
/* Protocol-----read/write bytes---Command_ram_address-----Command No */
{ RW_BYTE, 0x1, r_pmbus_page }, /* 0x0 */
{ RW_BYTE, 0x1, r_pmbus_opration }, /* 0x1 */
{ RW_BYTE, 0x1, r_pmbus_on_off_config }, /* 0x2 */
{ SEND_BYTE, 0x0, 0x0 }, /* 0x3 */
{ NO_ACTION, 0x0, r_pmbus_dummy }, /* 0x4 */
.
.
.
};
    
```

【変更後】

```

/*****
PMBus Protocol Table (0x00-0xFF)
*****/

static const command r_matrix[256] = {
/* Protocol-----read/write bytes---Command_ram_ad
{ RW_BYTE, 0x1, r_pmbus_page
{ NO_ACTION, 0x0, r_pmbus_dummy
ff_config }, /* 0x2 */
}, /* 0x3 */
r_pmbus_dummy }, /* 0x4 */
.
.
.
};
    
```

削除するコマンドの通信プロトコルを NO\_ACTION に変更

削除するコマンドのデータ長を 0 に変更

削除するコマンドのデータを格納するバッファのアドレスを r\_pmbus\_dummy に変更

手順 2 : r\_pmbus\_userapp.c ファイルの r\_func\_index[256]の変更。

r\_func\_index[256]は、関数ポインタ配列です。

各コマンド応答処理関数のアドレスを格納しています。

対応コマンドの追加、削除を行う場合は、関数ポインタ配列の登録、削除を行う必要があります。

例) 0x04 にコマンド追加する場合

**【変更前】**

```

/*****
PMBus Protocol Table (0x00-0xFF)
*****/

static void (* const r_func_index[256])( void ) = {
/* User function */

    &r_pmbus_user_page,           /* 0x0 */
    &r_pmbus_user_operation,      /* 0x1 */
    &r_pmbus_user_on_off_config,  /* 0x2 */
    &r_pmbus_user_clear_faults,   /* 0x3 */
    &r_pmbus_user_dummy,         /* 0x4 */
    .
};

```

**【変更後】**

```

/*****
PMBus Protocol Table (0x00-0xFF)
*****/

static void (* const r_func_index[256])( void ) = {
/* User function */

    &r_pmbus_user_page,           /* 0x0 */
    &r_pmbus_user_operation,      /* 0x1 */
    &r_pmbus_user_on_off_config,  /* 0x2 */
    &r_pmbus_user_clear_faults,   /* 0x3 */
    &r_pmbus_user_add,           /* 0x4 */
    .
};

```

追加するコマンドの  
関数を登録する

例) 0x01 のコマンドを削除する場合

【変更前】

```
/* *****  
PMBus Protocol Table (0x00-0xFF)  
***** */  
static void (* const r_func_index[256])( void ) = {  
/* User function */  
    &r_pmbus_user_page,           /* 0x0 */  
    &r_pmbus_user_operation,      /* 0x1 */  
    &r_pmbus_user_on_off_config,  /* 0x2 */  
    &r_pmbus_user_clear_faults,  /* 0x3 */  
    &r_pmbus_user_dummy,         /* 0x4 */  
    .  
};
```

【変更後】

```
/* *****  
PMBus Protocol Table (0x00-0xFF)  
***** */  
static void (* const r_func_index[256])( void ) = {  
/* User function */  
    &r_pmbus_user_page,           /* 0x0 */  
    &r_pmbus_user_dummy,         /* 0x1 */  
    &r_pmbus_user_on_off_config, /* 0x2 */  
    &r_pmbus_user_clear_faults,  /* 0x3 */  
    &r_pmbus_user_dummy,         /* 0x4 */  
    .  
};
```

削除するコマンドの関数を  
r\_pmbus\_user\_dummyに変更



## 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/>

お問い合わせ先

<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	2013.03.25	-	初版発行
1.10	2020.12.10	6 30	動作確認条件へ CC-RL 及び IAR コンパイラ環境を追加 5.6 ROM/RAM サイズを追加

## 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しないでください。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、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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  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、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)

## 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。

## 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/)