

RL78/I1D

ソフトウェア I2C スレーブ通信制御
(複数アドレス対応) CC-RL

R01AN3289JJ0100
Rev. 1.00
2016.11.15

要旨

本アプリケーションノートでは、ソフトウェアを用いた I2C バスのスレーブ機能で複数スレーブ・アドレスの実現方法を示します。

動作確認デバイス

RL78/I1D

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.	I2C バスのスレーブとしての基本的な仕様	4
1.1	I2C バスに関する仕様	4
1.2	スレーブ機能の仕様	4
2.	動作確認条件	4
3.	関連アプリケーションノート	5
4.	ハードウェア説明	6
4.1	ハードウェア構成例	6
4.2	使用端子一覧	7
5.	ソフトウェア説明	7
5.1	動作概要	7
5.2	オプション・バイトに反映される設定一覧	8
5.3	定数一覧	9
5.4	変数一覧	10
5.5	関数一覧	12
5.6	関数仕様	13
5.7	フローチャート	18
5.7.1	初期設定関数	18
5.7.2	システム関数	19
5.7.3	CPU クロックの設定	20
5.7.4	入出力ポート設定	21
5.7.5	タイマ・アレイ・ユニットの設定	22
5.7.6	A/D コンバータの設定	28
5.7.7	外部割り込み設定	34
5.7.8	メイン処理	37
5.7.9	R_MAIN_UserInit 処理	38
5.7.10	A/D 変換の初期設定	39
5.7.11	A/D 変換の起動	39
5.7.12	A/D 変換完了割り込み処理	41
5.7.13	LED 点灯の初期化処理	42
5.7.14	TM03 の起動処理	42
5.7.15	LED の発光データの設定処理	43
5.7.16	5ms インターバル・タイマ割り込み処理	44
5.7.17	I2C 通信の初期化処理	45
5.7.18	I2C 通信状態の確認	46
5.7.19	I2C 受信データを読み出し	46
5.7.20	I2C 送信バッファへのデータ設定処理	47
5.7.21	I2C 通信完了割り込み要求確認処理	47
5.7.22	I2C 通信完了割り込み処理	48
5.7.23	I2C 初期化処理（アセンブラ部）	52
5.7.24	I2C 通信ステータス読み出し処理	54
5.7.25	ACK 応答設定	54
5.7.26	ACK 応答読み出し	55
5.7.27	SCL エッジ検出割り込みエントリ処理	55
5.7.28	SDA エッジ検出割り込み処理	56
5.7.29	SCL エッジ検出割り込み処理	59
5.7.30	次データ送信開始処理	70
5.7.31	次データ受信開始処理	70
5.7.32	データ送信中止処理	70
6.	I2C バスの基本的な内容	71
6.1	ソフトウェアでの実現	71
6.2	スレーブとしての機能	72

6.2.1	LED 表示機能	72
6.2.2	A/D 変換機能	72
6.2.3	RAM 機能	72
6.3	ライブラリとのインタフェース仕様	73
6.3.1	送受信完了フラグ	73
6.3.2	次の通信開始関数	73
6.3.3	ストップ・コンディション検出フラグ	73
6.4	スレーブ・アドレスの指定	73
6.4.1	スレーブ・アドレス・テーブル	73
6.4.2	ACK 応答フラグ	74
6.5	スレーブをアクセスするためのプロトコル	75
6.5.1	LED への表示	75
6.5.2	A/D 変換結果の読み出し	75
6.5.3	RAM のデータ読み出し	76
6.5.4	RAM へのデータ書き込み	76
7.	ソフトウェアによる I2C バスの基本的な制御	77
7.1	エッジ検出割り込み	77
7.1.1	SCL エッジ検出	77
7.1.2	SDA エッジ検出	77
7.2	制御処理	78
7.2.1	SCL エッジ検出割り込みによるシーケンス 1	78
7.2.2	SCL エッジ検出割り込みによるシーケンス 2	78
7.2.3	SCL エッジ検出割り込みによるシーケンス 3	79
7.3	I2C スレーブのファイル構成	79
7.4	コード生成での設定内容	80
8.	サンプルコード	82
9.	参考ドキュメント	82

1. I2C バスのスレーブとしての基本的な仕様

1.1 I2C バスに関する仕様

I2C バスに関する仕様は以下の通りです。

- ・ 接続する I2C バス : ファースト・モード（最大 200kbps）と標準モード^注
- ・ スレーブ・アドレス 1 : 0x60（A/D 変換機能と LED 表示機能）
- ・ スレーブ・アドレス 2 : 0x70（RAM 機能）
- ・ スレーブ・アドレス 3 : 0x80（未使用）
- ・ スレーブ・アドレス 4 : 0x90（未使用）
- ・ 拡張コード対応 : 対応しない（無視して通信から退避）
- ・ アドレッシング : スレーブ・アドレスに続く 8 ビットで RAM のアドレスを指定します。

注 CPU/周辺ハードウェア・クロックに 24MHz を選択した場合の通信速度です。

1.2 スレーブ機能の仕様

スレーブの機能として、以下の 3 つの機能を提供します。3 つの機能はスレーブ・アドレスと送受信の状態で切り替わります。

- ・ LED 表示機能 : 8 ビットのデータを LED に表示します。2 個の表示データを SW で切り替えます。
- ・ A/D 変換機能 : 4 チャンネルのアナログ入力を A/D 変換します。マスタへの送信データは 16 サンプル分の移動平均値です。
- ・ RAM 機能 : 128 バイトの RAM として使用する事が出来る機能です。マスタは任意のアドレスを指定して読み/書きができます。

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、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/I1D (R5F117GC)
動作周波数	<ul style="list-style-type: none"> ● 高速オンチップ・オシレータ (HOCO) クロック : 24MHz ● CPU/周辺ハードウェア・クロック : 24MHz
動作電圧	3.3V (2.9V~3.6V で動作可能) LVD 動作モード : リセット・モード、電圧 : 2.75V
統合開発環境	ルネサス エレクトロニクス製 CS+ V3.03.00
コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.02.00 ^注
使用ボード	RL78/I1D ターゲット・ボード (R5F117GC、LED (モジュールの 10 個中 8 個使用)、SW 等を搭載)

【注】 CA78K0R と互換モードで使用します。

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アプリケーションノートを以下に示します。併せて参照してください。

RL78/G13 初期設定 (R01AN2575J) アプリケーションノート

RL78/I1D シリアル・アレイ・ユニット(簡易 I2C)による I2C マスタ通信制御 (R01AN3288J) アプリケーションノート

4. ハードウェア説明

4.1 ハードウェア構成例

図 4.1 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。

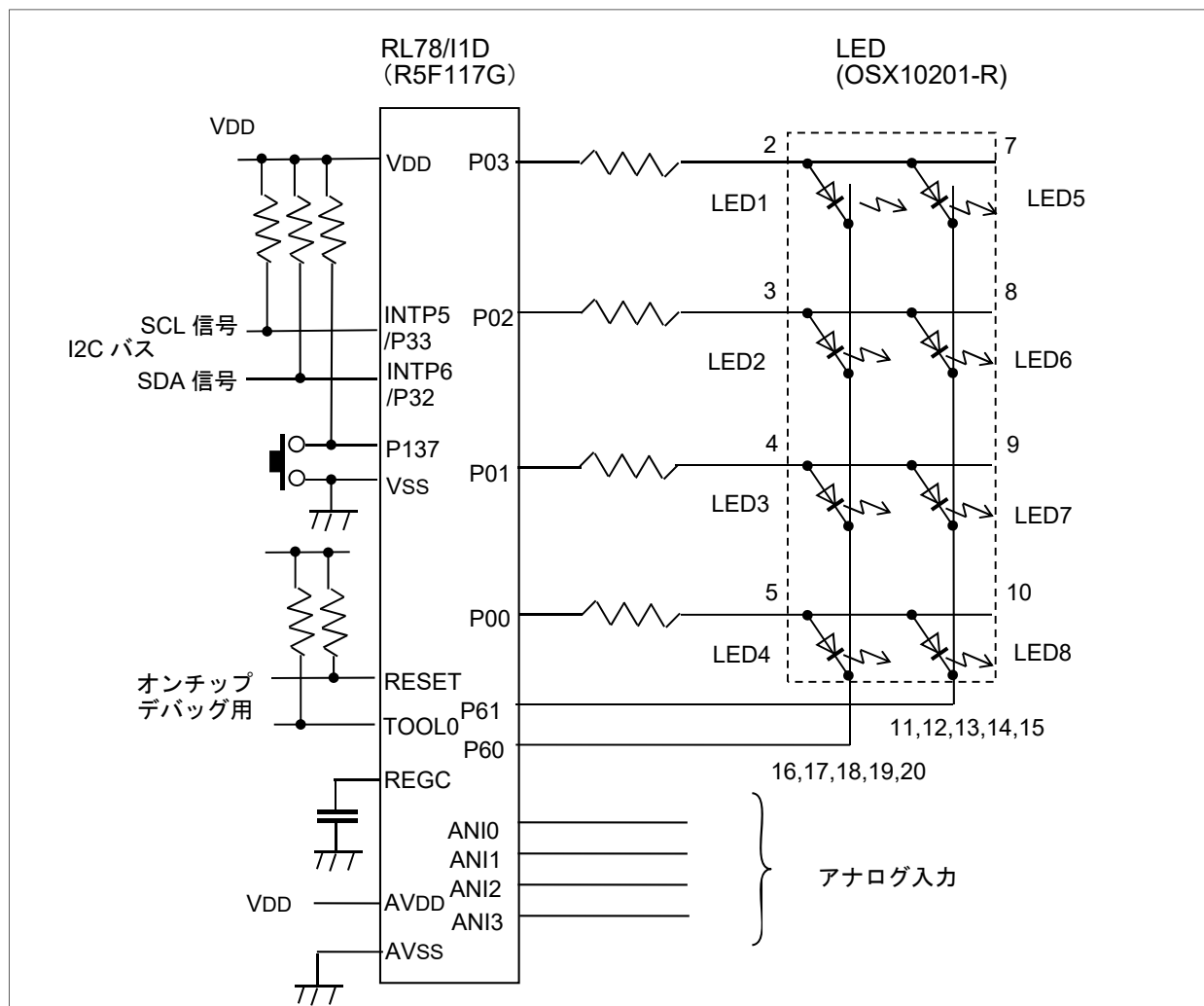


図 4.1 ハードウェア構成

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 VDD 又は VSS に接続して下さい）。

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

4.2 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
INTP6/P32	入出力	I2C 通信データ信号
INTP5/P33	入出力	I2C 通信クロック信号
P03~P00	出力	LED へのデータ出力
P61、P60	出力	LED の点灯タイミング出力
ANI3~ANI0	入力	アナログ信号入力
P137	入力	SW 入力

5. ソフトウェア説明

5.1 動作概要

- a) 初期設定
本アプリケーションノートでは、内蔵周辺機能の初期設定のみ CS+ のコード生成機能を利用します。内蔵周辺機能の初期設定が完了後、データの初期化を行い、A/D 変換と LED 点灯用タイマを起動します。
 - ・ A/D 変換と LED 点灯はタイマの割り込みによってバックグラウンドで処理します。
 - ・ I2C バスの通信は、INTP5 と INTP6 割り込みによりバックグラウンドで処理します。
- b) メイン処理
メイン処理は、4 チャンネル分の A/D 変換を待ちます。4 チャンネル分の A/D 変換が完了したら、移動平均値を I2C バスの送信用バッファに転送します。送信用バッファに転送されたデータは、マスタからの指示に対応して I2C バスに送信されます。4 チャンネル分の A/D 変換中に I2C バスでストップ・コンディション検出した場合、LED 点灯用のデータ受信バッファから点灯制御用バッファに転送します。
- c) A/D 変換完了割り込み処理
A/D 変換完了割り込み処理では、各チャンネルの変換結果を加算していきます。加算するデータが 16 以上になると、最も古いデータを最新の値に置き換えていきます。スキャン・モードでチャンネル 0~3 分の変換が完了すると、メイン処理に A/D 変換完了を通知します。
- d) 5ms のタイマ割り込み処理
5ms のタイマ割り込みは LED の表示および SW の確認に使用します。LED の点灯データを上位 4 ビット、下位 4 ビットの順に 2 時分割で点灯させます。SW の状態は 50ms に 1 回、状態を確認して、2 つの点灯データのどちらを使用するかを指定します。
- e) I2C 通信割り込み処理
SDA 信号と SCL 信号の変化により、INTP5、INTP6 割り込みが発生します。これを I2C 通信の割り込みとして使用します。I2C 通信割り込みが発生した場合、通信内容を解析し上位ソフトに通知します。1 バイト分の通信が完了後、自局への通信の場合は通信ステータスと受信データを変数に設定して、送受信完了フラグ（変数 `_g_iic_if`）をセットします。詳細は、「6.3 ライブラリとのインタフェース仕様」を参照してください。I2C バスでストップ・コンディションを検出した場合、インタフェース用の変数（`_g_stop_det`）を用いて、メイン処理に対してストップ・コンディションを検出したことを通知することで、I2C バスの通信が完了したことを示します。

以上のように、ほとんどの処理は割り込みベースとフラグで行われます。メイン処理は、割り込み処理で準備されたデータを別の割り込み処理で使用できるように、データを必要なバッファに設定する事を行います。

5.2 オプション・バイトに反映される設定一覧

表 5.1 にオプション・バイトに反映される値の例を示します。

表 5.1
オプション・バイト反映値

アドレス	設定値	内容
0x000C0	0b11101110	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)
0x000C1	0b01111111	LVD リセット・モード 2.75V (2.70V~2.87V)
0x000C2	0b11100000	HS モード、HOCO : 24MHz
0x000C3	0b10000100	オンチップ・デバッグ許可

5.3 定数一覧

表 5.2 にサンプルコードで使用する定数を示します。

表 5.2 サンプルコードで使用する定数

定数名	設定値	内容
TRUTH	1	真
FALSE	0	偽
POWER	4	A/D 変換のサンプル回数指定（2 の階乗で指定）
SAMPLE	2 << (POWER-1)	A/D 変換のサンプル回数
DATA_NUMBER	2	LED に表示するデータ数
INT_MASK	1	割り込み禁止（マスク）
INT_ENABLE	0	割り込み許可（マスク解除）
DETECT_START	0b11110010	スレーブとして選択を検出するマスク用ビット
DETECT_TRC	0b00001000	TRC（送信許可）ビット
DETECT_ACK	0b00000100	ACK 検出ビット
DETECT_STD	0b00000010	スタート・コンディション検出ビット
DETECT_STOP	0b00000001	ストップ・コンディション検出ビット
DISP_OFF_DATA	0b00000011	LED 消灯用 P6 データ
TIMING1	0b00000010	上位 4 ビット点灯用 P6 データ
TIMING2	0b00000001	下位 4 ビット点灯用 P6 データ
KEY_TIMING	10	SW 状態確認タイミング用
DATA_MAX	4	I2C 送信データ数
LED_MAX	2	LED 点灯用受信データ数
TX_LIMIT	DATA_MAX - 1	送信用ポインタのマスクデータ
RX_LIMIT	LED_MAX - 1	LED 点灯用データ受信ポインタのマスクデータ
SADR_TBL		使用するスレーブ・アドレスのテーブル
ACK_TBL		スレーブ・アドレス ID ごとの ACK 応答テーブル
P_IIC	P3	I2C バスで使用するポート
P_SCL	P3.3	SCL 信号で使用するポート
P_DATA	0b00001100	SCL と SDA を抽出するためのデータ
P_DATA_SCL	0b00001000	SCL がハイ、SDA がロー
P_SDA	P3.2	SDA 信号で使用するポート
PM_SCL	PM3.3	SCL 信号を制御するレジスタ
PM_SDA	PM3.2	SDA 信号を制御するレジスタ
ENG_SCL	EGN0.5	SCL 立下り検出許可
EPG_SCL	EPG0.5	SCL 立ち上がり検出許可
DIS_INTSCL	PMK5	SCL エッジ検出割り込みマスク
DIS_INTSDA	PMK6	SDA エッジ検出割り込みマスク
RQ_INTSCL	PIF5	SCL エッジ検出割り込み要求
RQ_INTSDA	PIF6	SDA エッジ検出割り込み要求
D_SDA	0xFFEDE.2	P3 イメージ中の SDA ビット
D_SCL	0xFFEDE.3	P3 イメージ中の SCL ビット
F_TRC	0xFFEDF.3	ステータス (g_IICS) の送信モード・ビット
F_ACKD	0xFFEDF.2	„ の ACK 検出ビット
F_STD	0xFFEDF.1	„ のスタート・コンディション検出ビット
F_SPD	0xFFEDF.0	„ のストップ・コンディション検出ビット

5.4 変数一覧

表 5.3 と表 5.4 にサンプルコードで使用する変数一覧を示します。

表 5.3 グローバル変数一覧（C 言語定義分）

Type	Variable Name	Contents	Function Used
uint16_t	g_conv_data	A/D 変換データのバッファ	R_ADC_Init() r_adc_interrupt()
uint16_t	g_sum_data	A/D の加算用バッファ	R_ADC_Init() r_adc_interrupt()
uint8_t	g_adc_end	A/D 変換完了フラグ	R_ADC_Init() r_adc_interrupt() main()
uint16_t *	gp_set_pt	A/D 変換結果格納用ポインタ	R_ADC_Init() r_adc_interrupt()
uint16_t *	gp_sum_pt	A/D 変換結果加算用ポインタ	R_ADC_Init() r_adc_interrupt() main()
uint8_t	g_disp_data_bf	LED 点灯データ	R_LED_Init() R_LED_DispData() r_tau0_channel3_interrupt()
uint8_t	g_sel_data	点灯データ指定	R_LED_Init() r_tau0_channel3_interrupt()
uint8_t	g_disp_timing	点灯タイミング	R_LED_Init() r_tau0_channel3_interrupt()
uint8_t	g_ram_area	RAM 機能用バッファ	R_IICA0_Init() _R_IIC_Rx_data() _R_IIC_Tx_data()
uint8_t	g_rx_data	受信データ用バッファ	R_IICA0_Init() R_IICA0_Get() _R_IIC_Rx_data()
uint16_t	g_tx_data	送信データ用バッファ	R_IICA0_Init() R_IICA0_Put() _R_IIC_Tx_data()
uint8_t	g_low_data_temp	送信の下位バイト保存用	_R_IIC_Tx_data()
uint8_t	g_low_data_index	下位バイト送信フラグ	R_IICA0_Init() _R_IIC_Rx_data() _R_IIC_Tx_data()
uint8_t	g_regadr	アドレス・レジスタ用フラグ	_R_IIC_Rx_data()
uint8_t	g_ptrx_data2	I2C 送受信用ポインタ	R_IICA0_Init() _R_IIC_Rx_data() _R_IIC_Tx_data()
uint_t	g_ptrx_data	I2C 送受信用ポインタ	R_IICA0_Init() _R_IIC_Rx_data() _R_IIC_Tx_data()
uint8_t	g_status	I2C 通信状態フラグ	R_IICA0_Init() R_IICA0_Status() _R_IIC_Rx_data() _R_IIC_Tx_data()

表 5.4 グローバル変数一覧（アセンブリ言語定義分）

Type	Variable Name	Contents	Function Used
uint8_t	<code>__g_stop_det</code> (<code>_g_stop_det</code>)	ストップ・コンディション検出フラグ	R_IICA0_Init() main() r_iic_SDA_interrupt
uint8_t	<code>__g_IIC_IF</code> (<code>_g_IIC_IF</code>)	送受信完了フラグ	R_IICA0_Init() r_iic_int_chk ()
uint8_t	<code>__g_IICS</code> (<code>_g_IICS</code>)	通信ステータス	r_iic_request()
uint8_t	<code>__g_IICA</code> (<code>_g_IICA</code>)	受信データ	r_iic_request ()
uint8_t	<code>g_ACKE_tbl</code>	アドレスへの ACK 応答制御用初期値の内部バッファ	<code>__R_IICSS_Init</code> <code>__set_ACKE_table</code> <code>__get_ACKE_table</code> r_iic_SCL_interrupt
uint8_t	<code>g_ACKE</code>	ACK 応答制御用	r_iic_SCL_interrupt
uint16_t	<code>next_proc</code>	次の INTP5 処理関数アドレス	<code>__R_IICSS_Init</code> () r_iic_SCL_interrupt r_iic_SDA_interrupt
uint8_t ^注	<code>bit_count</code>	送受信のビットカウント	r_iic_SCL_interrupt
uint8_t ^注	<code>g_IICA</code>	送受信でシフト中のデータ	r_iic_SCL_interrupt
uint8_t	<code>g_P_image</code>	P3 のデータ保存用	r_iic_SCL_interrupt
uint8_t	<code>g_IICS</code> ^{注2}	I2C のステータス	r_iic_SCL_interrupt

【注】1 bit_count と g_IICA をまとめて 16 ビットでのアクセスも行う。

2 I2C の通信状態を示す変数 g_IICS は IICA0 に準拠した構成としている

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ID3	ID2	ID1	ID0	F_TRC	F_ACKD	F_STD	F_SPD

【備考】変数 `__g_stop_det` (`_g_stop_det`)、`__g_IIC_IF` (`_g_IIC_IF`)、`__g_IICS` (`_g_IICS`) および `__g_IICA` (`_g_IICA`) だけが C 言語からアクセス可能な変数。他はアセンブリ言語記述のライブラリ関数でのみ使用する。

5.5 関数一覧

表 5.5 に使用する関数一覧を示します。

表 5.5 関数一覧

関数名	概要
R_IICA0_Init()	I2C 通信関係の変数を初期化します
R_IICA0_Status()	I2C 通信状態を読み出します
R_IICA0_Get()	I2C 受信データを読み出します
R_IICA0_Put()	I2C 送信バッファにデータを設定します
r_iic_int_chk()	I2C 通信完了チェック
r_iic_request()	I2C 通信完了割り込み処理
R_ADC_Init()	A/D 変換関係の変数を初期化します
R_ADC_Start()	A/D 変換を起動します
r_adc_interrupt()	A/D 変換完了割り込み処理
R_LED_Init()	LED 表示関係の変数を初期化します
R_TM03_Start()	5ms のインターバル・タイマを起動します
R_LED_DispData ()	LED 発光データを設定します
r_tau0_channel3_interrupt()	5ms インターバル・タイマ割り込み処理
__R_IICSS_Init	I2C 関係の初期化
__R_IICSS_Status	I2C の状態 (g_IICS の値) を戻します
__set_ACKE_table	スレーブ・アドレス ID の ACK 応答を設定します
__get_ACKE_table	スレーブ・アドレス ID の ACK 応答設定を読み出します
r_iic_SCL_interrupt	SCL 信号のエッジ検出割り込み処理
r_iic_SDA_interrupt	SDA 信号のエッジ検出割り込み処理
__Tx_data_sub	I2C バスのウェイトを解除し、次のデータの送信を開始します
__Tx_end_sub	通信から退避し、I2C バスのウェイトを解除します
__Rx_data_sub	I2C バスのウェイトを解除し、次のデータ受信を開始します

二重線は各モジュールの区切りを示します。

5.6 関数仕様

サンプルコードの関数仕様を示します。

[関数名] R_IICA0_Init

概要	I2C 通信の初期化処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void R_IICA0_Init(void);
説明	I2C 通信で使用する変数を初期化します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_IICA0_Status

概要	I2C 通信状態の確認処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	uint8_t R_IICA0_Status(void);
説明	I2C の通信状態を示す変数 (g_IICS) の値を読み出します。I2C の通信が完了していたら、対応する処理を行います。
引数	なし
リターン値	変数 g_IICS (g_status) の値
備考	なし

[関数名] R_IICA0_Get

概要	I2C の受信データ・バッファから受信データを読み出し処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	uint8_t R_IICA0_Get(uint8_t ptr);
説明	I2C の受信バッファの引数で指定されたデータ（点灯データ）を読み出します。I2C の通信が完了していたら、対応する処理を行います。
引数	受信データ・バッファを指定
リターン値	受信したデータ
備考	なし

[関数名] R_IICA0_Put

概要	I2C 送信バッファへのデータ設定処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void R_IICA0_Put(uint8_t ptr, uint16_t data);
説明	I2C の送信バッファの第 1 引数で示されたアドレスに第 2 引数で示されたデータ（A/D 変換結果）を格納します。I2C の通信が完了していたら、対応する処理を行います。
引数	第 1 引数 データの格納アドレス 第 2 引数 送信するデータ
リターン値	なし
備考	なし

[関数名] r_iic_int_chk

概要	I2C 通信完了チェック
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	uint8_t r_iic_int_chk (void);
説明	I2C 通信完了割り込みフラグをチェックし、通信が完了していれば、r_iic_request を呼び出します。
引数	なし
リターン値	I2C のステータス
備考	なし

[関数名] r_iic_requestr_IIC_interrupt

概要	I2C 通信完了処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void r_iic_request (void);
説明	IICA0 の INTIICA0 に相当する処理です。I2C のステータス（_g_IICS）の値に対応した処理を行います。
引数	なし
リターン値	なし
備考	_g_IICS に I2C 通信のステータス、_g_IICA に受信データ

[関数名] R_ADC_Init

概要	A/D 変換の初期設定処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void R_ADC_Init(void);
説明	A/D 変換関係の変数を初期化します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_ADC_Start

概要	A/D 変換の起動処理
ヘッダ	r_cg_userdefine.h
宣言	void R_ADC_Start(void);
説明	A/D コンバータを起動します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_adc_interrupt

概要	A/D 変換完了割り込み処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	#pragma interrupt r_adc_interrupt(vect=INTAD,bank=RB2,enable=true) __interrupt static void r_adc_interrupt(void);
説明	A/D 変換完了割り込みで起動され、得られた変換結果はバッファに格納すると同時にチャンネルごとに 16 回分加算していきます。
引数	なし
リターン値	なし
備考	なし

[関数名] R_LED_Init

概要	LED 点灯の初期化処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void R_LED_Init (void);
説明	LED 点灯制御用変数を初期化します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_TM03_Start

概要	TM03（インターバルタイマ）の起動処理
ヘッダ	r_cg_userdefine.h
宣言	void R_TM03_Start(void);
説明	TM03（5ms インターバル・タイマ）を起動します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_LED_DispData

概要	LED の発光データの設定処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void R_LED_DispData(uint8_t CH_No, uint8_t in_data);
説明	第 1 引数で指定されたバッファに第 2 引数で指定したデータを格納します。
引数	第 1 引数 データの格納チャンネル 第 2 引数 設定するデータ
リターン値	なし
備考	なし

[関数名] r_tau0_channel3_interrupt

概要	5ms インターバル・タイマ割り込み処理
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	#pragma interrupt r_tau0_channel3_interrupt(vect=INTTM03, enable=true) __interrupt static void r_tau0_channel3_interrupt(void);
説明	5ms のインターバル・タイマ割り込みで起動され、LED の 4 ビット単位でのダイナミック点灯を制御します。また、50ms に 1 回は P137 に接続された SW の状態を確認し点灯するデータの切り替えを行います。
引数	なし
リターン値	なし
備考	なし

[関数名] __R_IICSS_Init

概要	I2C の初期化処理
宣言	void __R_IICSS_Init (void);
説明	I2C 制御変数やハードウェアの初期化を行います。
引数	なし
リターン値	なし
備考	なし

[関数名] __R_IICSS_Status

概要	I2C ステータス確認処理
宣言	uint8_t __R_IICSS_Status (void);
説明	I2C バスの状態を示す変数 g_IICS の値を渡します。
引数	なし
リターン値	I2C バスのステータス
備考	なし

[関数名] __set_ACKE_table

概要	スレーブ・アドレス ID の ACK 応答を設定します。
宣言	void __set_ACKE_table(uint8_t ACKE);
説明	引数のビット 4~1 で示されたスレーブ・アドレス ID の ACK 応答をビット 0 で示された状態にします。
引数	ACK 応答
リターン値	なし
備考	なし

[関数名] __get_ACKE_table

概要	スレーブ・アドレス ID の ACK 応答設定を読み出します。
宣言	uint8_t __get_ACKE_table(uint8_t ID);
説明	引数で渡されたスレーブ・アドレス ID の ACK 応答設定を戻します。
引数	スレーブ・アドレス ID
リターン値	ACK 応答の設定値
備考	なし

[関数名] r_iic_SCL_interrupt

概要	SCL エッジ検出割り込み処理
宣言	r_iic_SCL_interrupt .VECTOR 0x00012
説明	SCL 信号のエッジを検出し、SCL と SDA を読み出し、対応する処理を行います。 処理する内容は変数 next_proc に格納されたアドレスで示します
引数	なし
リターン値	なし
備考	なし

[関数名] r_iic_SDA_interrupt

概要	SDA エッジ検出割り込み処理
宣言	r_iic_SDA_interrupt .VECTOR 0x00014
説明	SDA 信号のエッジ検出で起動され、SDA 信号と SCL 信号の状態からスタート・コンディションとストップ・コンディションを検出します。
引数	なし
リターン値	なし
備考	なし

[関数名] __Tx_data_sub

概要	次データの送信開始処理
宣言	void __Tx_data_sub(uint8_t data);
説明	引数で渡されたデータの MSB を SDA 信号に出力し、I2C バスのウェイトを解除して送信を開始します。
引数	次の送信データ
リターン値	なし
備考	なし

[関数名] __Rx_data_sub

概要	次データの受信開始処理
宣言	void __Rx_data_sub(void);
説明	I2C バスのウェイトを解除してデータ受信を開始します。
引数	なし
リターン値	なし
備考	なし

[関数名] __Tx_end_sub

概要	送信完了処理
宣言	void __Tx_end_sub(void);
説明	通信から退避し、I2C バスのウェイトを解除します。
引数	なし
リターン値	なし
備考	なし

5.7 フローチャート

図 5.1 に本アプリケーションノートの全体フローを示します。

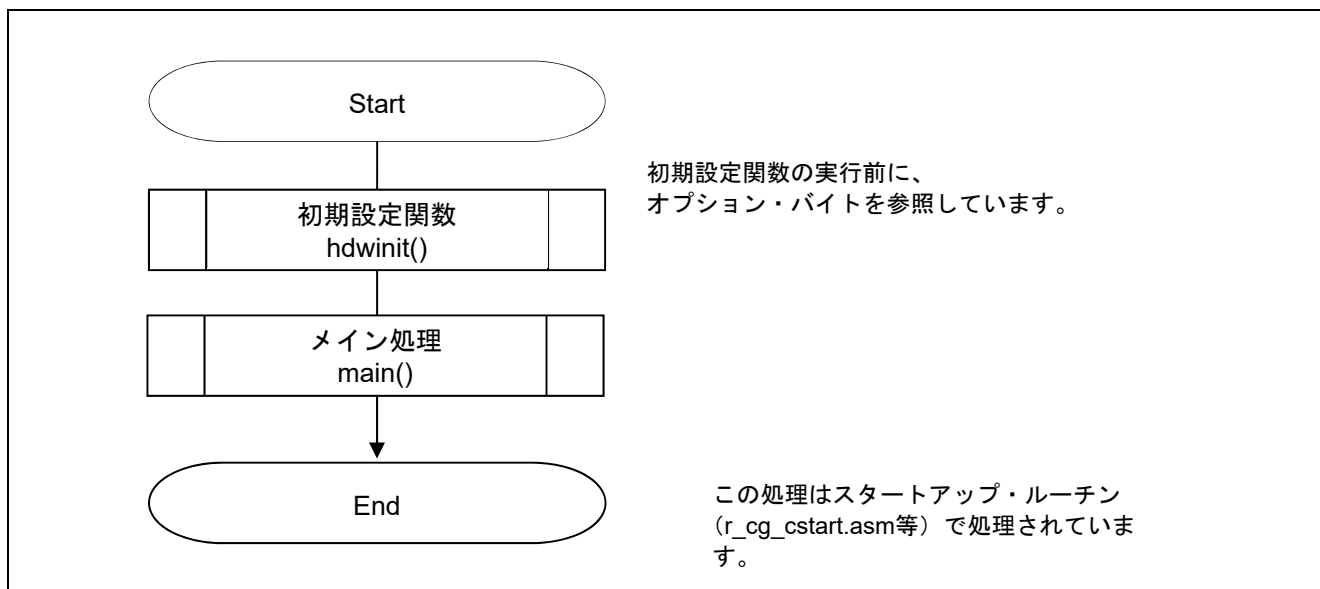


図 5.1 全体フロー

注 この処理は、スタートアップ・ルーチン (r_cg_cstart.asm 等) で処理されています。初期設定関数とメイン処理関数の呼び出しの間でメモリ関係の設定を行っています。

5.7.1 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。

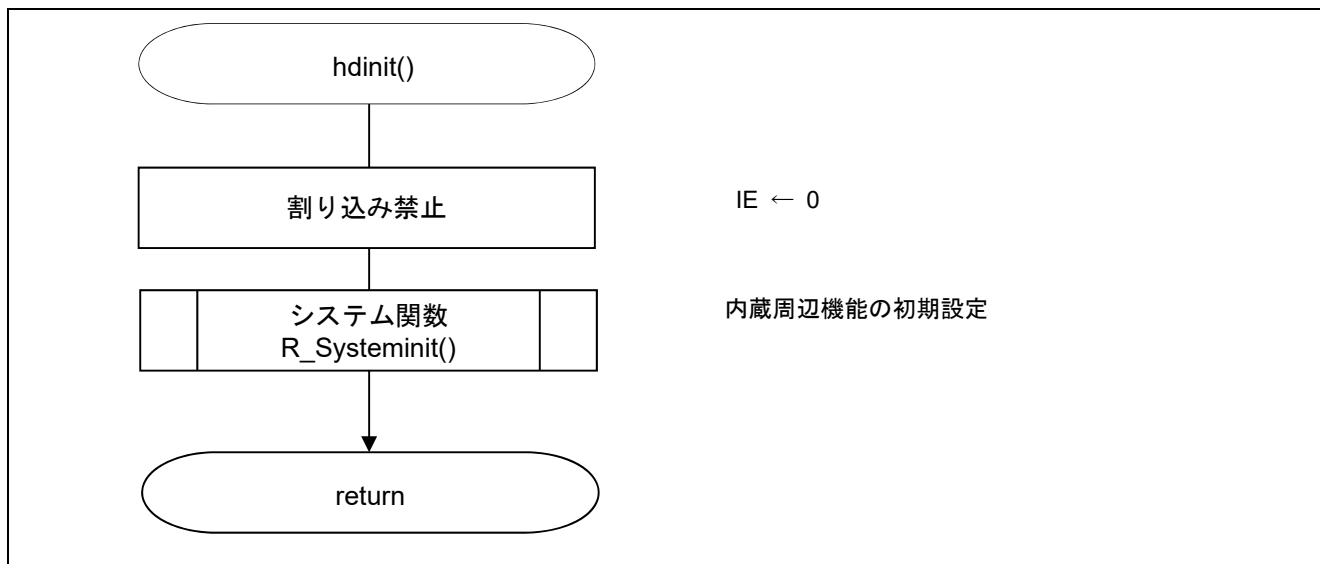


図 5.2 初期設定関数

5.7.2 システム関数

図 5.3 にシステム関数のフローチャートを示します。

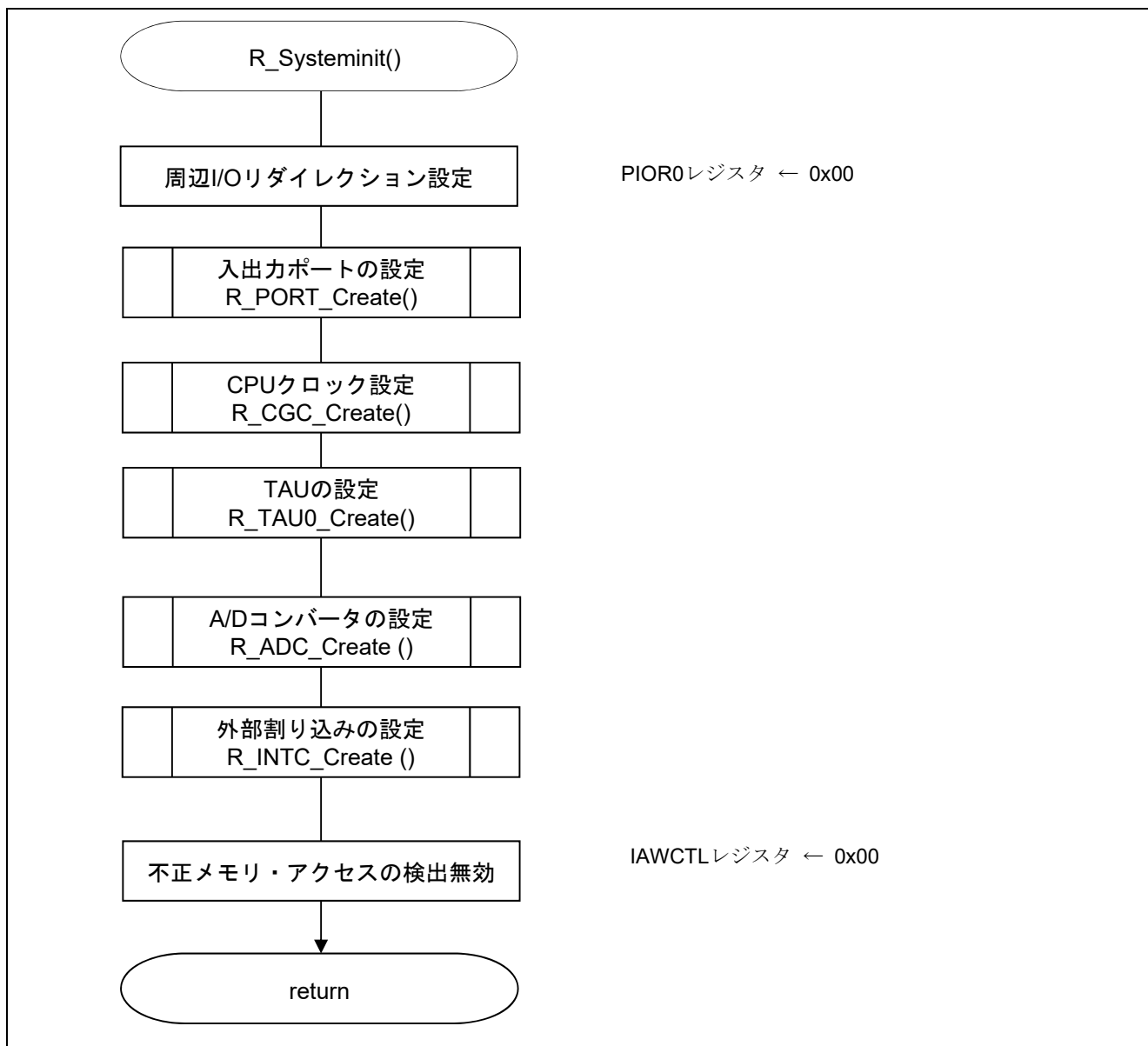


図 5.3 システム関数

5.7.3 CPU クロックの設定

図 5.4 に CPU クロック設定のフローチャートを示します。

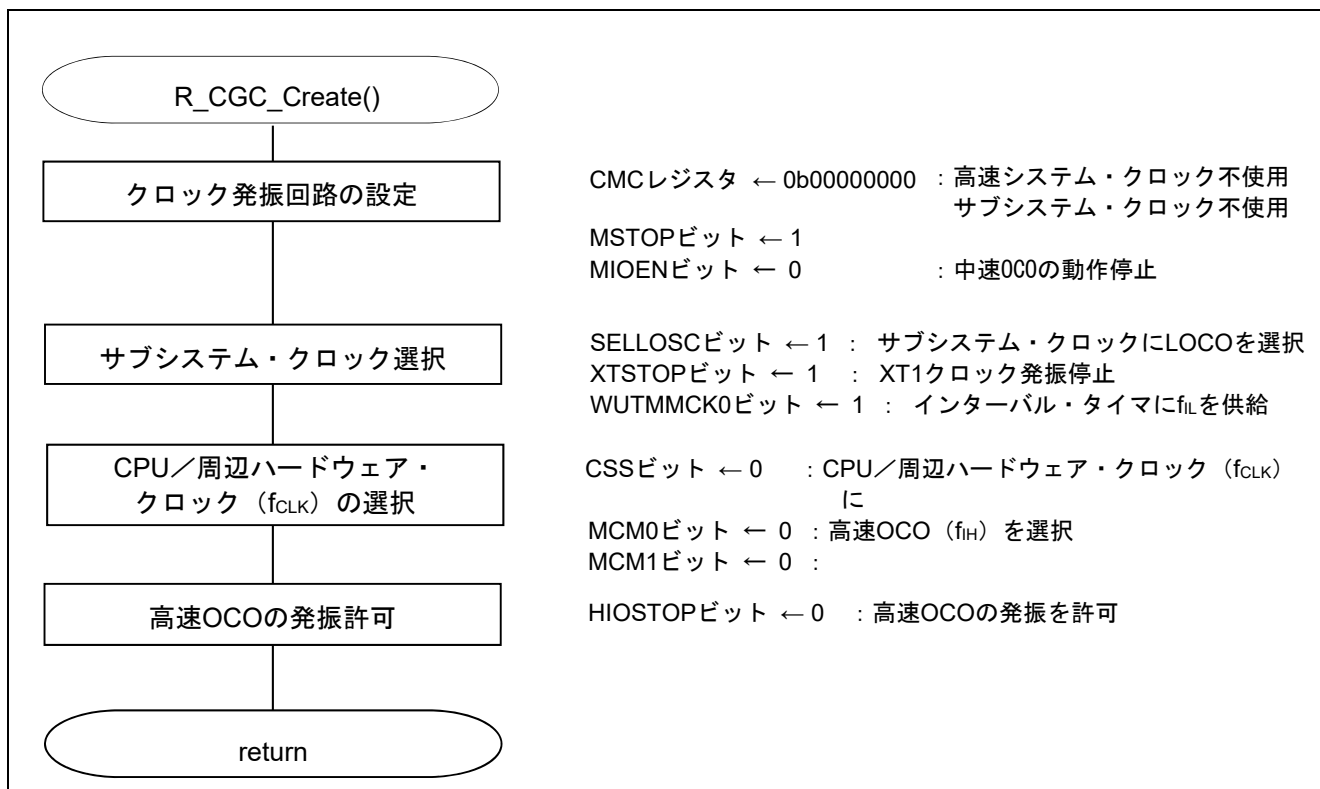


図 5.4 CPU クロックの設定

5.7.4 入出力ポート設定

図 5.5 に入出力ポート設定のフローチャートを示します。

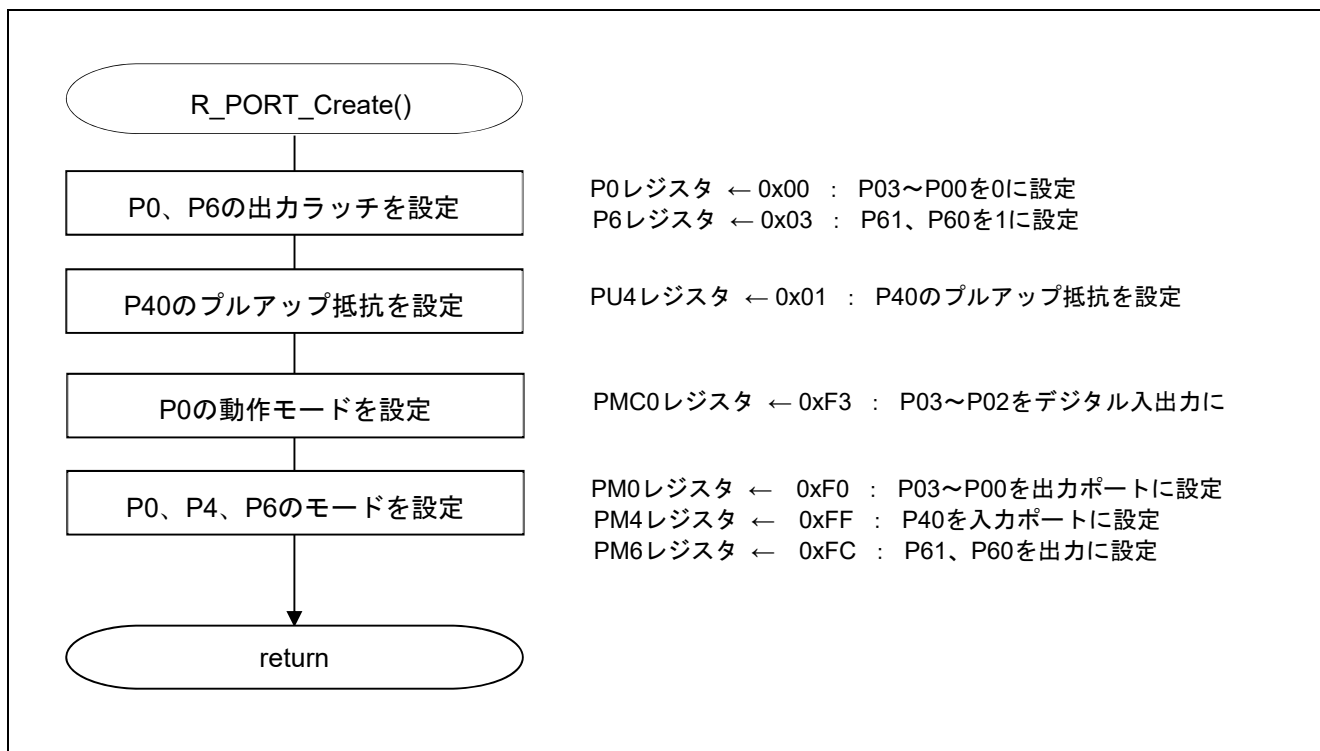


図 5.5 入出力ポート設定

注意 未使用のポートは、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。また、未使用の入力専用ポートは個別に抵抗を介して VDD 又は Vss に接続して下さい。

5.7.5 タイマ・アレイ・ユニットの設定

図 5.6 にタイマ・アレイ・ユニットの設定のフローチャートを示します。

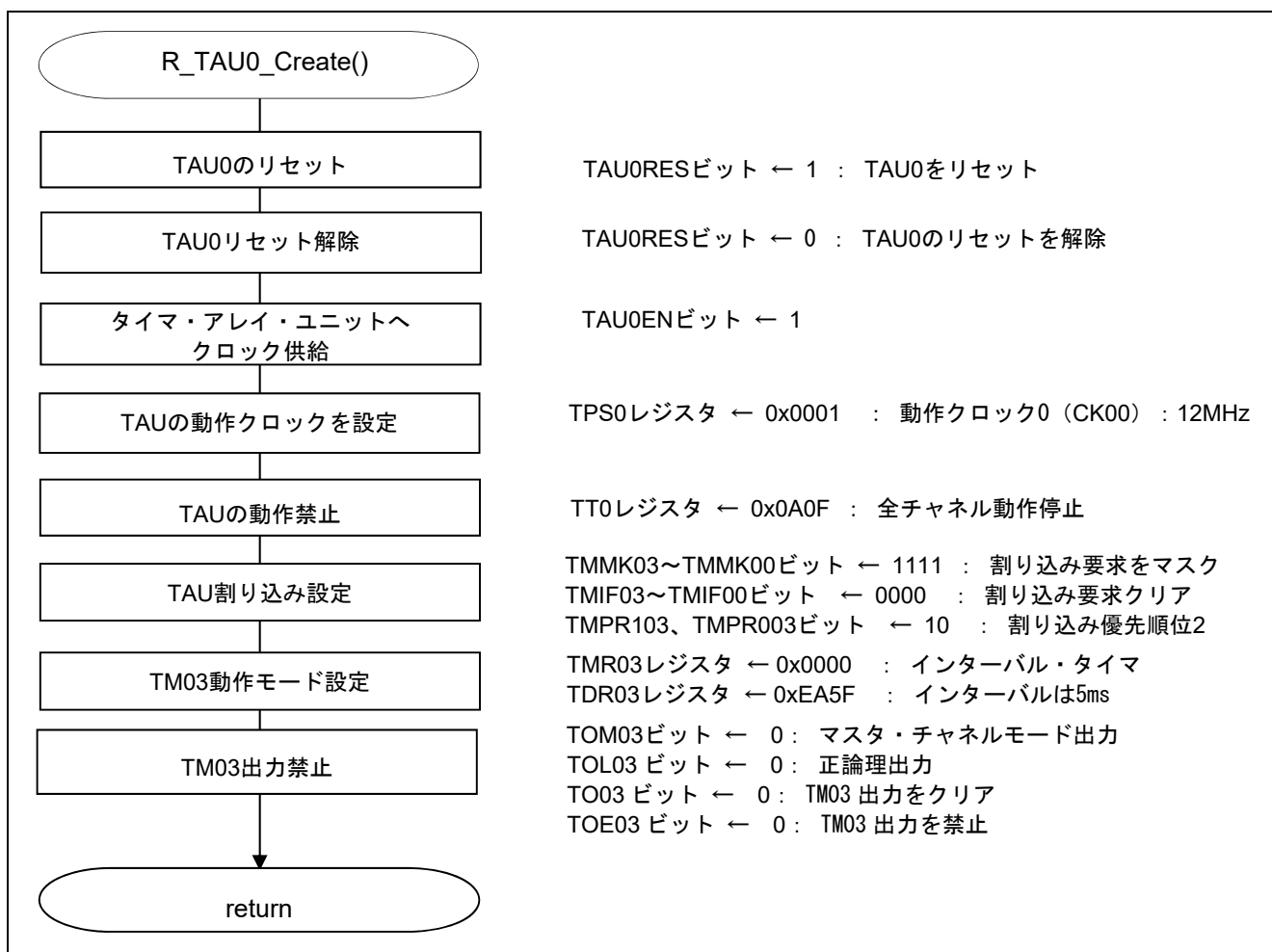


図 5.6 タイマ・アレイ・ユニットの設定

TAU0 のリセット

・周辺リセット制御レジスタ 0 (PRR0)

TAU0 をリセットします

略号 : PRR0

7	6	5	4	3	2	1	0
0	0	ADCRES	0	0	SAU0RES	0	TAU0RES
0	0	x	0	0	x	0	1/0

ビット 0

SAU0RES	タイマ・アレイ・ユニットのリセット制御
0	タイマ・アレイ・ユニットのリセット解除
1	タイマ・アレイ・ユニットはリセット状態

タイマ・アレイ・ユニット 0 へのクロック供給開始

・周辺イネーブル・レジスタ 0 (PER0)

タイマ・アレイ・ユニット 0 へのクロック供給を開始します

略号 : PER0

	7	6	5	4	3	2	1	0
RTCWEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN	
	x	0	x	0	0	x	0	1

ビット 0

TAU0EN	タイマ・アレイ・ユニット 0 の入力クロックの制御
0	入力クロック供給停止
1	入力クロック供給

タイマ・クロック周波数の設定

・タイマ・クロック選択レジスタ 0 (TPS0)

タイマ・アレイ・ユニット 0 の動作クロックを選択

略号 : TPS0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	PRS 031	PRS 030	0	0	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000
	0	0	x	x	0	0	x	x	x	x	x	x	0	0	0	1

ビット 3-0

PRS 003	PRS 002	PRS 001	PRS 000	動作クロック (CK00) の選択					
				$f_{CLK} =$ 1MHz	$f_{CLK} =$ 2MHz	$f_{CLK} =$ 4MHz	$f_{CLK} =$ 16MHz	$f_{CLK} =$ 24MHz	
0	0	0	0	f_{CLK}	1 MHz	2 MHz	4 MHz	16MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	500 kHz	1 MHz	2 MHz	8 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	250 kHz	500 kHz	1 MHz	4MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	125 kHz	250 kHz	500 kHz	2 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	62.5 kHz	125 kHz	250 kHz	1 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	31.3 kHz	62.5 kHz	125 kHz	500 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	15.6 kHz	31.3 kHz	62.5 kHz	250 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	7.81 kHz	15.6 kHz	31.3 kHz	125 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	3.91 kHz	7.81 kHz	15.6 kHz	62.5 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	1.95 kHz	3.91 kHz	7.81 kHz	31.3 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	977 Hz	1.95 kHz	3.91 kHz	15.6 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	488 Hz	977 Hz	1.95 kHz	7.81 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	244 Hz	488 Hz	977 Hz	3.91 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	122 Hz	244 Hz	488 Hz	1.95 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	61 Hz	122 Hz	244 Hz	977 Hz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	30.5 Hz	61 Hz	122 Hz	488 Hz	732 Hz

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

タイマ動作停止の設定

- ・タイマ・チャンネル停止レジスタ 0 (TT0)
タイマ・チャンネルの動作停止を選択

略号：TT0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TT03H	0	TT01H	0	0	0	0	0	TT03	TT02	TT01	TT00
0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	1

ビット n

TT0n	チャンネル n の動作停止トリガ
0	トリガ動作しない
1	TE0n ビットが 0 にクリアされ、カウント動作を停止する（停止トリガ発生）

タイマのカウント完了割り込みの設定

- ・割り込みマスク・フラグ・レジスタ (MK1L) の TMMK03 ビット
割り込みマスクの設定
- ・割り込み要求フラグ・レジスタ (IF1L) の TMIF03 ビット
割り込み要求フラグのクリア
- ・優先順位指定フラグ・レジスタ (PR01L、PR11L) の TMPR003、TMPR103 ビット
TM03 の割り込み優先度をレベル 2 に設定します

略号：MK1L

ビット 5

TMMK03	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

略号：IF1L

ビット 5

TMIF03	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

略号：PR01L、PR11L

ビット 5

TMPR103	TMPR003	INTTM03 の優先順位レベルの選択
0	0	レベル 0 を指定(高優先順位)
0	1	レベル 1 を指定
1	0	レベル 2 を指定
1	1	レベル 3 を指定(低優先順位)

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

チャンネル 3 の動作モードの設定

- ・ タイマ・モード・レジスタ 03 (TMR03)
 - 動作クロック (f_{MCK}) の選択
 - カウント・クロックの選択
 - スタート・トリガとキャプチャ・トリガの設定
 - タイマ入力の有効エッジ選択
 - 動作モード設定

略号：TMR03

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS031	CKS030	0	CCS03	SPLIT03	STS032	STS031	STS030	CIS031	CIS030	0	0	MD033	MD032	MD031	MD030
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ビット 15、14

CKS031	CKS030	チャンネル 3 の動作クロック (f_{MCK}) の選択
0	0	タイマ・クロック選択レジスタ 0 (TPS0) で設定した動作クロック CK00
0	1	タイマ・クロック選択レジスタ 0 (TPS0) で設定した動作クロック CK02
1	0	タイマ・クロック選択レジスタ 0 (TPS0) で設定した動作クロック CK01
1	1	タイマ・クロック選択レジスタ 0 (TPS0) で設定した動作クロック CK03

ビット 12

CCS03	チャンネル 3 のカウント・クロック (f_{TCLK}) の選択
0	CKS031、CKS030 ビットで指定した動作クロック (f_{MCK})
1	TI03 端子からの入力信号の有効エッジ

ビット 11

SPLIT03	チャンネル 3 の 8 ビット・タイマ/16 ビット・タイマ動作の選択
0	16 ビット・タイマとして動作
1	8 ビット・タイマとして動作

ビット 10-8

STS032	STS031	STS030	チャンネル 3 のスタート・トリガ、キャプチャ・トリガの設定
0	0	0	ソフトウェア・トリガ・スタートのみ有効（他のトリガ要因を非選択にする）
0	0	1	TI00 端子入力の有効エッジを、スタート・トリガ、キャプチャ・トリガの両方に使用
0	1	0	TI00 端子入力の両エッジを、スタート・トリガとキャプチャ・トリガに分けて使用
1	0	0	マスタ・チャンネルの割り込み信号を使用（複数チャンネル連動動作機能のスレーブ・チャンネル時）
上記以外			設定禁止

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

ビット 7 - 6

CIS031	CIS030	TIO3 端子の有効エッジ選択
0	0	立ち下がリエッジ
0	1	立ち上がりエッジ
1	0	両エッジ（ロウ・レベル幅測定時） スタート・トリガ：立ち下がリエッジ、キャプチャ・トリガ：立ち上がりエッジ
1	1	両エッジ（ハイ・レベル幅測定時） スタート・トリガ：立ち上がりエッジ、キャプチャ・トリガ：立ち下がリエッジ

ビット 3 - 0

MD 033	MD 032	MD 031	MD 030	チャンネル 3 の動作モードの設定	対応する機能	TCR のカウント動作
0	0	0	1/0	インターバル・タイム・モード	インターバル・タイマ/方形波出力/分周器機能/PWM 出力（マスタ）	ダウン・カウント
0	1	0	1/0	キャプチャ・モード	入力パルス間隔測定	アップ・カウント
0	1	1	0	イベント・カウンタ・モード	外部イベント・カウンタ	ダウン・カウント
1	0	0	1/0	ワンカウント・モード	ディレイ・カウンタ/ワンショット・パルス出力/PWM 出力（スレーブ）	ダウン・カウント
1	1	0	0	キャプチャ&ワンカウント・モード	入力信号のハイ/ロウ・レベル幅測定	アップ・カウント
上記以外				設定禁止		

ディレイ時間設定

- ・タイマ・データ・レジスタ 03（TDR03）
ディレイ時間を設定

略号：TDR03

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

タイマ出力禁止設定

- ・タイマ出力モード・レジスタ 0（TOM0L）
マスタ・モード出力に設定
- ・タイマ出力レベル・レジスタ 0（TOL0L）
正論理出力に設定
- ・タイマ出力レジスタ 0（TO0L）
出力を 0 に設定
- ・タイマ出力許可レジスタ 0（TOE0L）
各チャンネルのタイマ出力許可/禁止の値設定

略号：TOM0L

7	6	5	4	3	2	1	0
0	0	0	0	TOM03	TOM02	TOM01	0
0	0	0	0	0	x	x	0

ビット 3

TOM03	チャンネル 3 のタイマ出力モードの制御
0	マスタ・チャンネル出力モード
1	スレーブ・チャンネル出力モード

略号：TOL0L

7	6	5	4	3	2	1	0
0	0	0	0	TOL03	TOL02	TOL01	0
0	0	0	0	0	x	x	0

ビット 3

TOL03	チャンネル 3 のタイマ出力レベルの制御
0	正論理出力(アクティブ・ハイ)
1	反転出力(アクティブ・ロウ)

略号：TO0L

7	6	5	4	3	2	1	0
0	0	0	0	TO03	TO02	TO01	TO00
0	0	0	0	0	x	x	x

ビット 3

TO03	チャンネル 3 のタイマ出力レベルの制御
0	ロウ・レベル
1	ハイ・レベル

略号：TOE0L

7	6	5	4	3	2	1	0
0	0	0	0	TOE03	TOE02	TOE01	TOE00
0	0	0	0	0	x	x	x

ビット 3

TOE03	チャンネル 3 のタイマ出力許可／禁止
0	カウント動作による TO03（タイマ・チャンネル出力ビット）の動作停止。
1	カウント動作による TO03（タイマ・チャンネル出力ビット）の動作許可。

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

5.7.6 A/D コンバータの設定

図 5.7 に A/D コンバータの設定のフローチャートを示します。

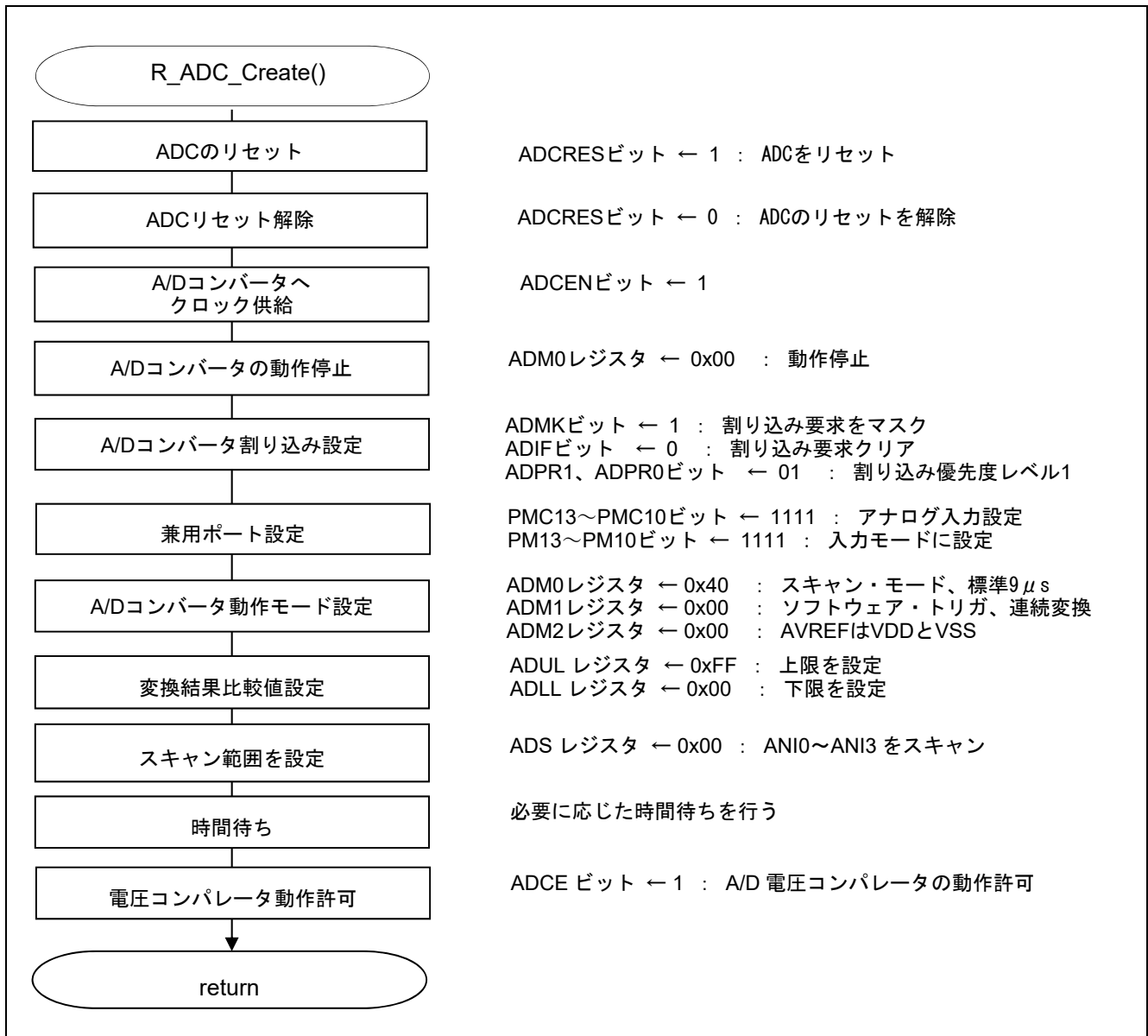


図 5.7 A/D コンバータの設定

ADC のリセット

- ・周辺リセット制御レジスタ 0 (PRR0)
ADC をリセットします

略号 : PRR0

7	6	5	4	3	2	1	0
0	0	ADCRES	0	0	SAU0RES	0	TAU0RES
0	0	1/0	0	0	x	0	x

ビット 5

ADCRES	A/D コンバータのリセット制御
0	A/D コンバータのリセット解除
1	A/D コンバータはリセット状態

A/D コンバータへのクロック供給開始

- ・周辺イネーブル・レジスタ 0 (PER0)
- A/D コンバータへのクロック供給を開始します

略号 : PER0

7	6	5	4	3	2	1	0
RTCEN	0	ADCEN	I2CA0EN	SAU1EN	SAU0EN	0	TAU0EN
x	0	1	x	x	x	0	x

ビット 5

ADCEN	A/D コンバータの入カクロックの制御
0	入カクロック供給停止
1	入カクロック供給

A/D コンバータの動作停止

- ・A/D コンバータ・モード・レジスタ 0 (ADM0)
- A/D コンバータの動作を停止します

略号 : ADM0

7	6	5	4	3	2	1	0
ADCS	ADMD	FR2	FR1	FR0	LV1	LV0	ADCE
0	x	x	x	x	x	x	0

ビット 7

ADCS	A/D 変換動作の制御
0	変換動作停止
1	変換動作許可

ビット 0

ADCE	A/D 電圧コンパレータの動作制御
0	A/D 電圧コンパレータの動作停止
1	A/D 電圧コンパレータの動作許可

A/D 変換完了割り込みの設定

- ・割り込みマスク・フラグ・レジスタ (MK1H) の ADMK ビット
割り込みマスクの設定
- ・割り込み要求フラグ・レジスタ (IF1H) の ADIF ビット
割り込み要求フラグのクリア
- ・優先順位指定フラグ・レジスタ (PR01H、PR11H) の ADPR0、ADPR1 ビット
A/D 変換完了の割り込み優先度をレベル 1 に設定します

ビット 0

ADMK	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

ビット 0

ADIF	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

ビット 0

ADPR1	ADPR0	優先順位レベルの選択
0	0	レベル 0 を指定（高優先順位）
0	1	レベル 1 を指定
1	0	レベル 2 を指定
1	1	レベル 3 を指定（低優先順位）

兼用ポートの設定

- ・ポート・モード・コントロール・レジスタ 1 (PMC1)
端子をアナログ入力に設定します。
- ・ポート・モード・レジスタ 1 (PM1)
ポートの出力バッファをオフします

略号：PMC1

7	6	5	4	3	2	1	0
PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10
x	x	x	x	1	1	1	1

ビット 3～0

PMC1n	P1n 端子のデジタル入出力／アナログ入力の選択(n = 0-7)
0	デジタル入出力(アナログ入力以外の兼用機能)
1	アナログ入力

略号：PM1

7	6	5	4	3	2	1	0
PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
x	x	x	x	1	1	1	1

ビット 3～0

PM1n	P1n 端子の入出力モードの選択
0	出力モード（出力バッファ・オン）
1	入力モード（出力バッファ・オフ）

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

A/D コンバータ動作モードの設定

- ・A/D コンバータ・モード・レジスタ 0 (ADM0)
変換動作をスキャン・モードに設定します。
変換時間を $9\mu\text{s}$ に設定します。
- ・A/D コンバータ・モード・レジスタ 1 (ADM1)
ソフトウェア・トリガ・モードに設定します。
連続変換モードに設定します。
- ・A/D コンバータ・モード・レジスタ 2 (ADM2)
基準電圧を設定します。
12 ビット分解能に設定します。

略号：ADM0

7	6	5	4	3	2	1	0
ADCS	ADMD	FR2	FR1	FR0	LV1	LV0	ADCE
0	1	1	0	1	0	0	1

ビット6

ADMD	A/D 変換チャンネル選択モードを設定
0	セレクト・モード
1	スキャン・モード

ビット5-1

FR2	FR1	FR0	LV1	LV0	12ビット分解能時の変換時間					
					f _{CLK} = 1MHz	f _{CLK} = 4MHz	f _{CLK} = 8MHz	f _{CLK} = 16MHz	f _{CLK} = 24MHz	
0	0	0	0	0	設定禁止	設定禁止	設定禁止	設定禁止	72 μs	
0	0	1					54 μs	38 μs		
0	1	0					54 μs	27 μs	18 μs	
0	1	1					40.5 μs	20.25 μs	13.5 μs	
1	0	0					33.75 μs	16.875 μs	11.25 μs	
1	0	1					54 μs	27 μs	13.5 μs	9 μs
1	1	0					27 μs	13.5 μs	6.75 μs	4.5 μs
1	1	1					54 μs	13.5 μs	6.75 μs	3.375 μs
0	0	0	0	1	設定禁止	設定禁止	設定禁止	設定禁止	88 μs	
0	0	1					66 μs	44 μs		
0	1	0					66 μs	33 μs	22 μs	
0	1	1					49.5 μs	24.75 μs	16.5 μs	
1	0	0					41.25 μs	20.625 μs	13.75 μs	
1	0	1					66 μs	33 μs	16.5 μs	11 μs
1	1	0					33 μs	16.5 μs	8.25 μs	5.5 μs
1	1	1					66 μs	16.5 μs	8.25 μs	4.125 μs

略号：ADM1

7	6	5	4	3	2	1	0
ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0
0	0	0	0	0	0	0	0

ビット7-6

ADTMD1	ADTMD0	A/D 変換トリガ・モードの選択
0	x	ソフトウェア・トリガ・モード
1	0	ハードウェア・トリガ・ノーウエイト・モード
1	1	ハードウェア・トリガ・ウエイト・モード

ビット5

ADSCM	A/D 変換動作モードの選択
0	連続変換モード
1	ワンショット変換モード

略号：ADM1

7	6	5	4	3	2	1	0
ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0
0	0	0	0	0	0	0	0

ビット 1 - 0

ADTRS1	ADTRS0	ハードウェア・トリガ信号の選択
0	0	タイマ・チャンネル 1 のカウント完了またはキャプチャ完了割り込み信号 (INTTM01)
0	1	ELC で選択されたイベント信号
1	0	リアルタイム・クロック 2 割り込み信号 (INTRTC)
1	1	2 ビット・インターバル・タイマ割り込み信号 (INTIT)

略号：ADM2

7	6	5	4	3	2	1	0
ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP
0	0	0	0	0	0	0	0

ビット 7 - 6

ADREFP1	ADREFP0	A/D コンバータの+側の基準電圧の選択
0	0	AVDD から供給
0	1	AVREFP/ANI0 から供給
1	0	内部基準電圧 (1.45 V) から供給
1	1	設定禁止

ビット 5

ADREFM	A/D コンバータの-側の基準電圧の選択
0	AVss から供給
1	AVREFM/ANI1 から供給

ビット 3

ADRCK	変換結果上限/下限値チェック
0	ADLL レジスタ \leq ADCR レジスタ \leq ADUL レジスタ (AREA1) のとき割り込み信号 (INTAD) が発生。
1	ADCR レジスタ $<$ ADLL レジスタ (AREA2), ADUL レジスタ $<$ ADCR レジスタ (AREA3) のとき割り込み信号 (INTAD) が発生

ビット 2

AWC	SNOOZE モードの設定
0	SNOOZE モード機能を使用しない
1	SNOOZE モード機能を使用する

ビット 0

ADTYP	A/D 変換分解能の選択
0	12 ビット分解能
1	8 ビット分解能

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

変換結果の上限／下限の設定

- ・変換結果比較上限値設定レジスタ（ADUL）
上限値を設定します
- ・変換結果比較下限値設定レジスタ（ADLL）
下限値を設定します

略号：ADUL

7	6	5	4	3	2	1	0
ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0
1	1	1	1	1	1	1	1

略号：ADLL

7	6	5	4	3	2	1	0
ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0
0	0	0	0	0	0	0	1

A/D 変換チャネル設定

- ・アナログ入力チャネル指定レジスタ（ADS）
ANI0-ANI3 を設定します

略号：ADS

7	6	5	4	3	2	1	0
ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0
0	0	0	0	0	0	0	0

ビット4－0

ADS4	ADS3	ADS2	ADS1	ADS0	アナログ入力チャネル
0	0	0	0	0	ANI0－ANI3
0	0	0	0	1	ANI1－ANI4
0	0	0	1	0	ANI2－ANI5
0	0	0	1	1	ANI3－ANI6
以下の組み合わせは省略					

5.7.7 外部割り込み設定

図 5.8 に外部割り込み設定のフローチャートを示します。

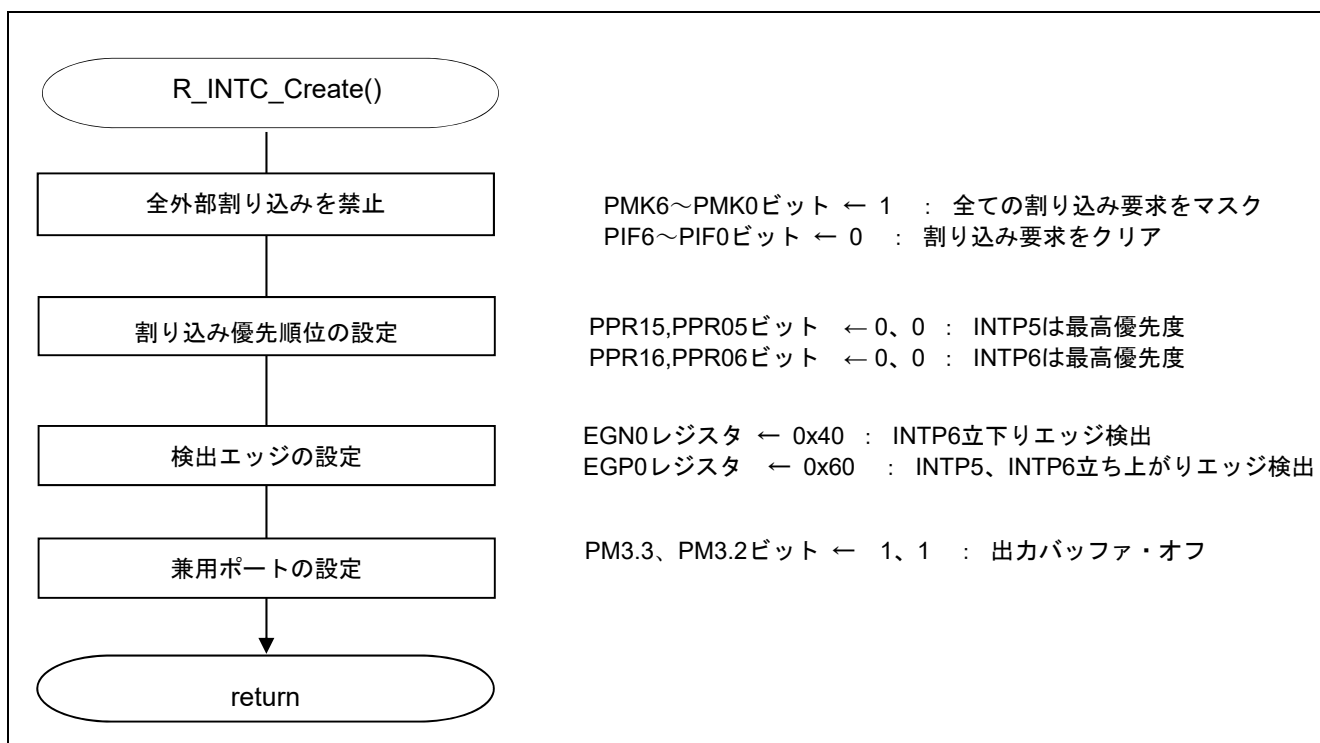


図 5.8 外部割り込み設定

全外部割り込みの禁止

- ・割り込みマスク・フラグ・レジスタ（MK0L、MK0H）
割り込み要求をマスクします
- ・割り込み要求フラグ・レジスタ（IF0L、IF0H）
割り込み要求をクリアします

略号：PMK0L

7	6	5	4	3	2	1	0
PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK
1	1	1	1	1	1	x	x

略号：PMK0H

7	6	5	4	3	2	1	0
RTITMK	TMMK00	SREMK0	1	1	SRMK0	STMK0	PMK6
x	x	x	1	1	x	x	1

ビット n

PMKn	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

略号：PIF0L

7	6	5	4	3	2	1	0
PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF
0	0	0	0	0	0	x	x

略号：PIF0H

7	6	5	4	3	2	1	0
RTITIF	TMIF00	SREIF0	0	0	SRIF0	STIF0	PIF6
x	x	x	0	0	x	x	0

ビット n

PIFn	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

割り込み優先順位の設定

- ・優先順位指定フラグ・レジスタの PPR15、PPR15、PPR16、PPR06 ビット
最高優先順位に設定

ビット 7

PPR15	PPR05	INTP5 の優先順位レベルの選択
0	0	レベル 0 を指定（高優先順位）
0	1	レベル 1 を指定
	0	レベル 2 を指定
1	1	レベル 3 を指定（低優先順位）

ビット 0

PPR16	PPR06	INTP6 の優先順位レベルの選択
0	0	レベル 0 を指定（高優先順位）
0	1	レベル 1 を指定
	0	レベル 2 を指定
1	1	レベル 3 を指定（低優先順位）

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

エッジ検出の設定

- ・外部割り込み立ち上がりエッジ許可レジスタ(EGP0)
INTP5、INTP6のエッジ検出許可
- ・外部割り込み立ち下がりエッジ許可レジスタ(EGN0)
INTP6のエッジ検出許可

略号：EGP0

7	6	5	4	3	2	1	0
0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0
0	1	1	0	0	0	0	0

略号：EGN0

7	6	5	4	3	2	1	0
0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0
0	1	0	0	0	0	0	0

ビット6、5

EGPn	EGNn	INTPn 端子の有効エッジの選択
0	0	エッジ検出禁止
0	1	立ち下がりエッジ
1	0	立ち上がりエッジ
1	1	立ち上がり、立ち下がりの両エッジ

兼用ポートの設定

- ・ポート・モード・レジスタ 3(PM3)レジスタの PM3.3、PM3.2 ビット
出力バッファをオフに設定

略号：PM3

7	6	5	4	3	2	1	0
1	1	1	1	PM33	PM32	PM31	PM30
1	1	1	1	1	1	x	x

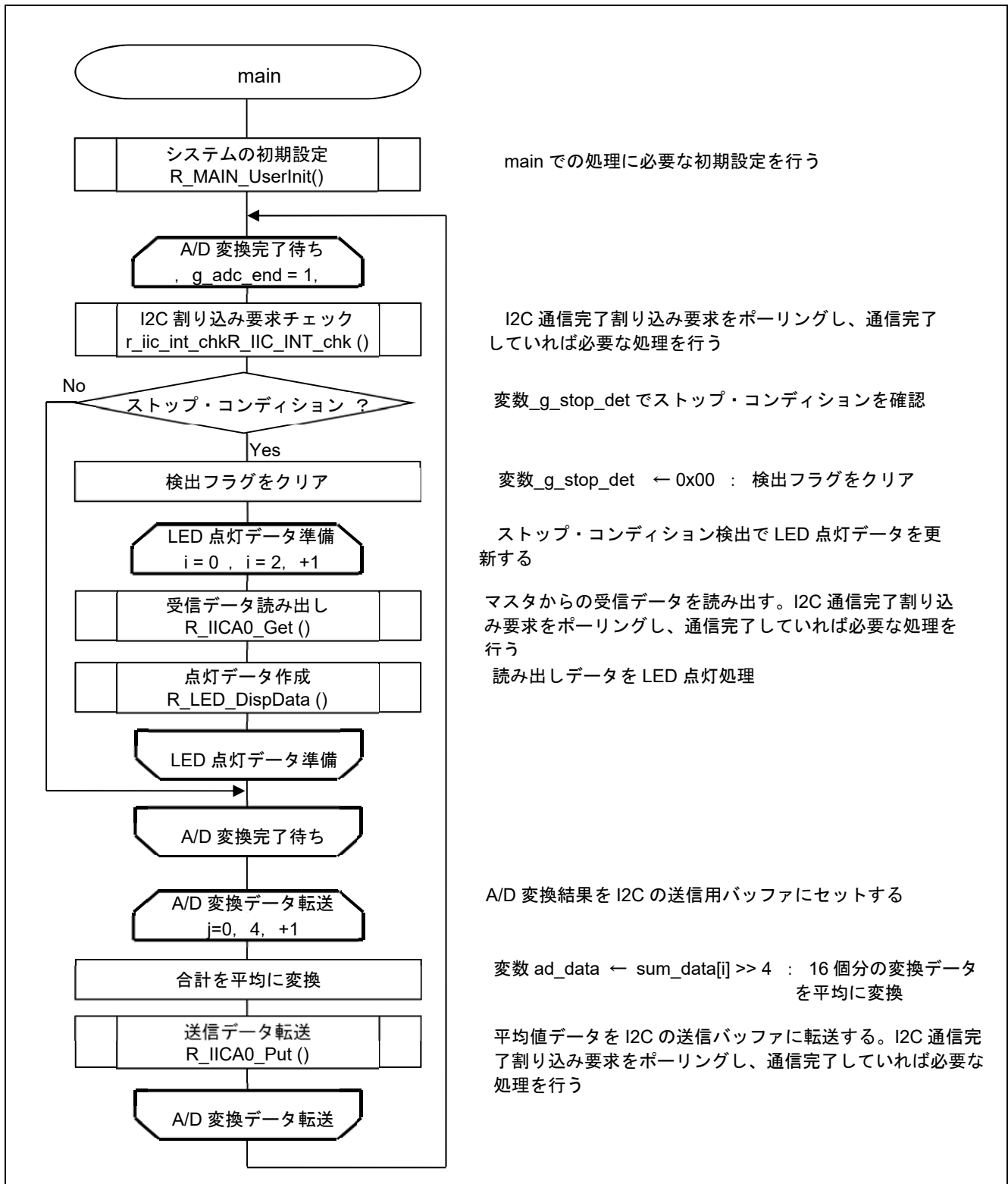
ビット3、2

PM3n	P3n 端子の入出力モードの選択
0	出力モード(出力ポートとして機能(出力バッファ・オン))
1	入力モード(入力ポートとして機能(出力バッファ・オフ))

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

5.7.8 メイン処理

図 5.9 にメイン処理のフローチャートを示します。



main での処理に必要な初期設定を行う

I2C 通信完了割り込み要求をポーリングし、通信完了していれば必要な処理を行う

変数 g_stop_det でストップ・コンディションを確認

変数 g_stop_det ← 0x00 : 検出フラグをクリア

ストップ・コンディション検出で LED 点灯データを更新する

マスタからの受信データを読み出す。I2C 通信完了割り込み要求をポーリングし、通信完了していれば必要な処理を行う

読み出しデータを LED 点灯処理

A/D 変換結果を I2C の送信用バッファにセットする

変数 ad_data ← sum_data[i] >> 4 : 16 個分の変換データを平均に変換

平均値データを I2C の送信バッファに転送する。I2C 通信完了割り込み要求をポーリングし、通信完了していれば必要な処理を行う

図 5.9 メイン処理

5.7.9 R_MAIN_UserInit 処理

図 5.10 に R_MAIN_UserInit 処理のフローチャートを示します。

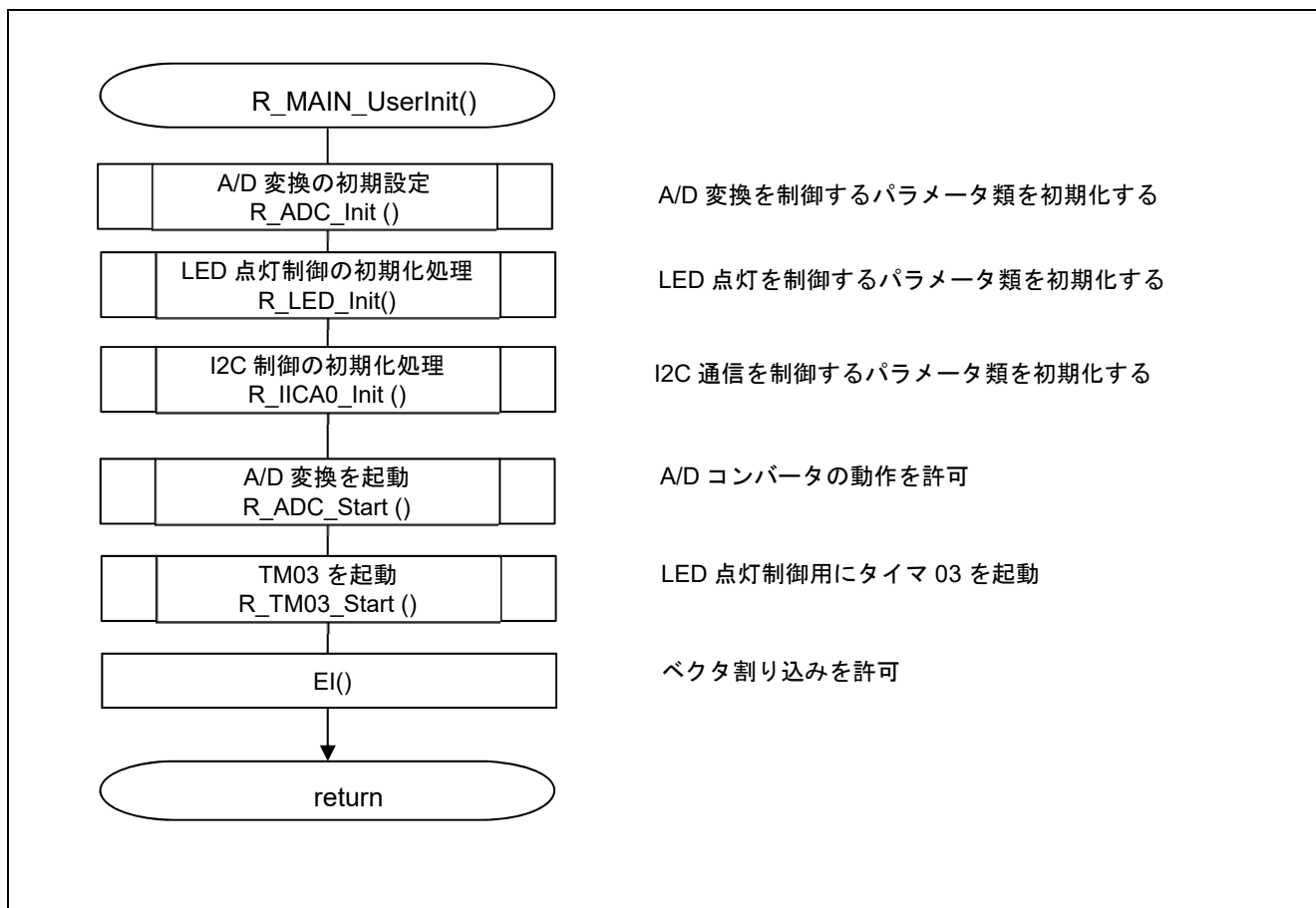


図 5.10 R_MAIN_UserInit 処理

5.7.10 A/D 変換の初期設定

図 5.11 に A/D 変換の初期設定処理のフローチャートを示します。

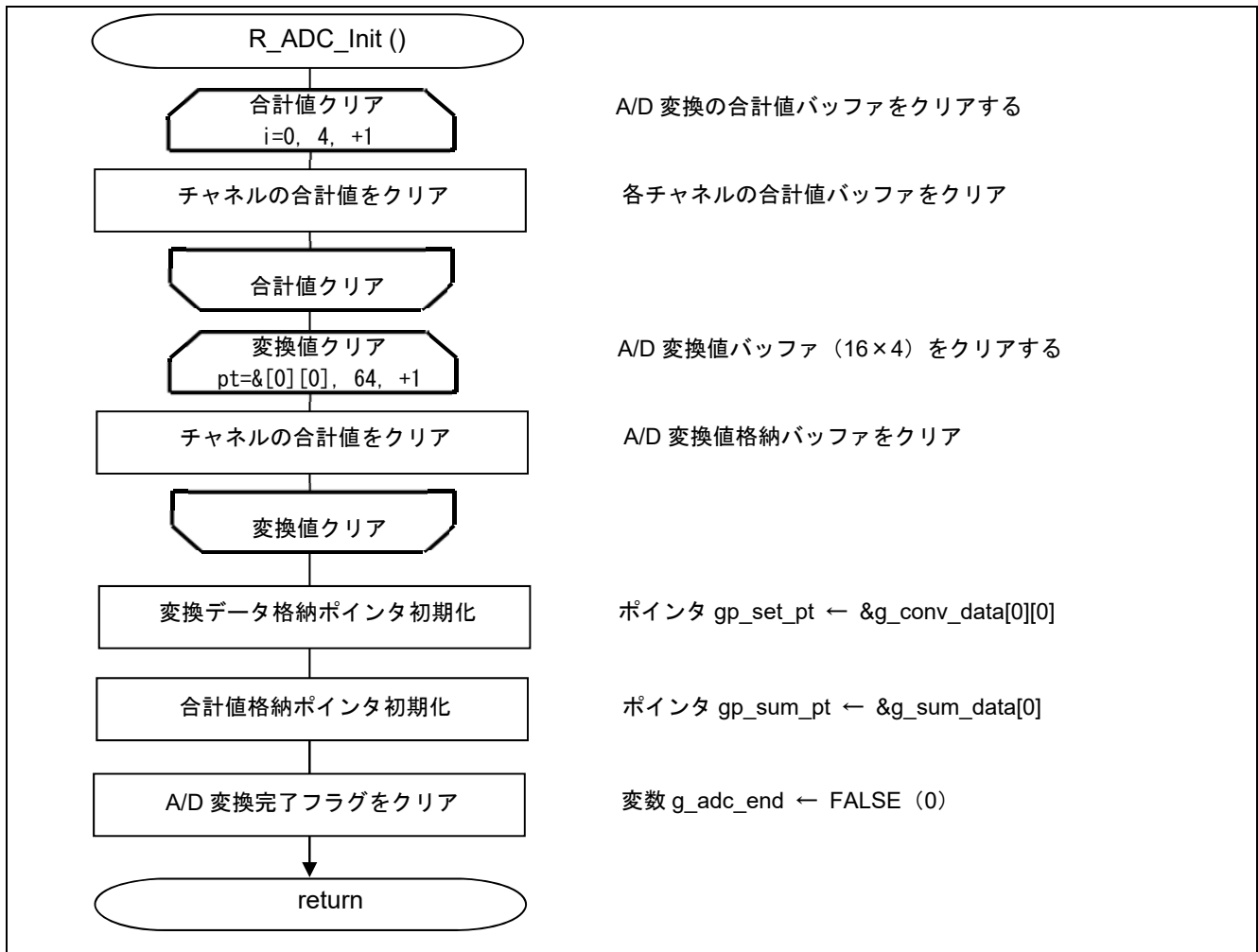


図 5.11 A/D 変換の初期設定処理

5.7.11 A/D 変換の起動

図 5.12 に A/D 変換の起動処理のフローチャートを示します。

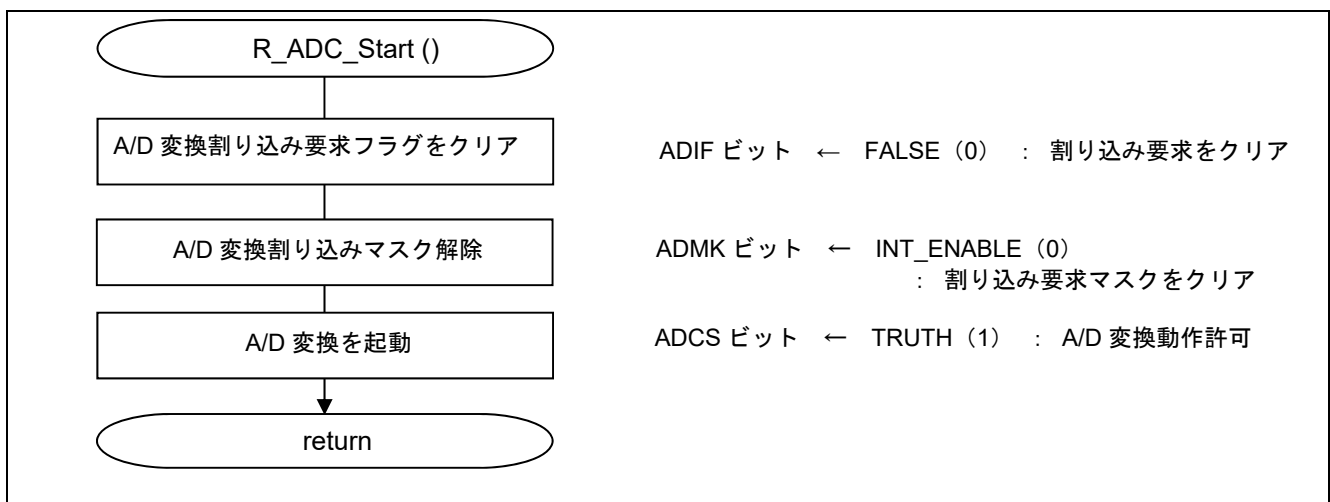


図 5.12 A/D 変換の起動処理

A/D 変換割り込みの設定

- ・ 割り込み要求フラグ・レジスタ 1H (IF1H) の ADIF ビット
割り込み要求のクリア
- ・ 割り込み要求マスク・フラグ・レジスタ 1H (MK1H) の ADMK ビット
割り込み要求マスクの解除

ビット0

ADIF	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

ビット0

ADMK	割り込み要求フラグ
0	割り込み処理許可
1	割り込み処理禁止

A/D 変換起動

- ・ A/D コンバータ・モード・レジスタ 0 (ADM0)
A/D コンバータの動作を許可します

略号：ADM0

7	6	5	4	3	2	1	0
ADCS	ADMD	FR2	FR1	FR0	LV1	LV0	ADCE
1	1	1	0	1	0	0	1

ビット7

ADCS	A/D 変換動作の制御
0	変換動作停止
1	変換動作許可

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

5.7.12 A/D 変換完了割り込み処理

図 5.13 に A/D 変換完了割り込み処理のフローチャートを示します。

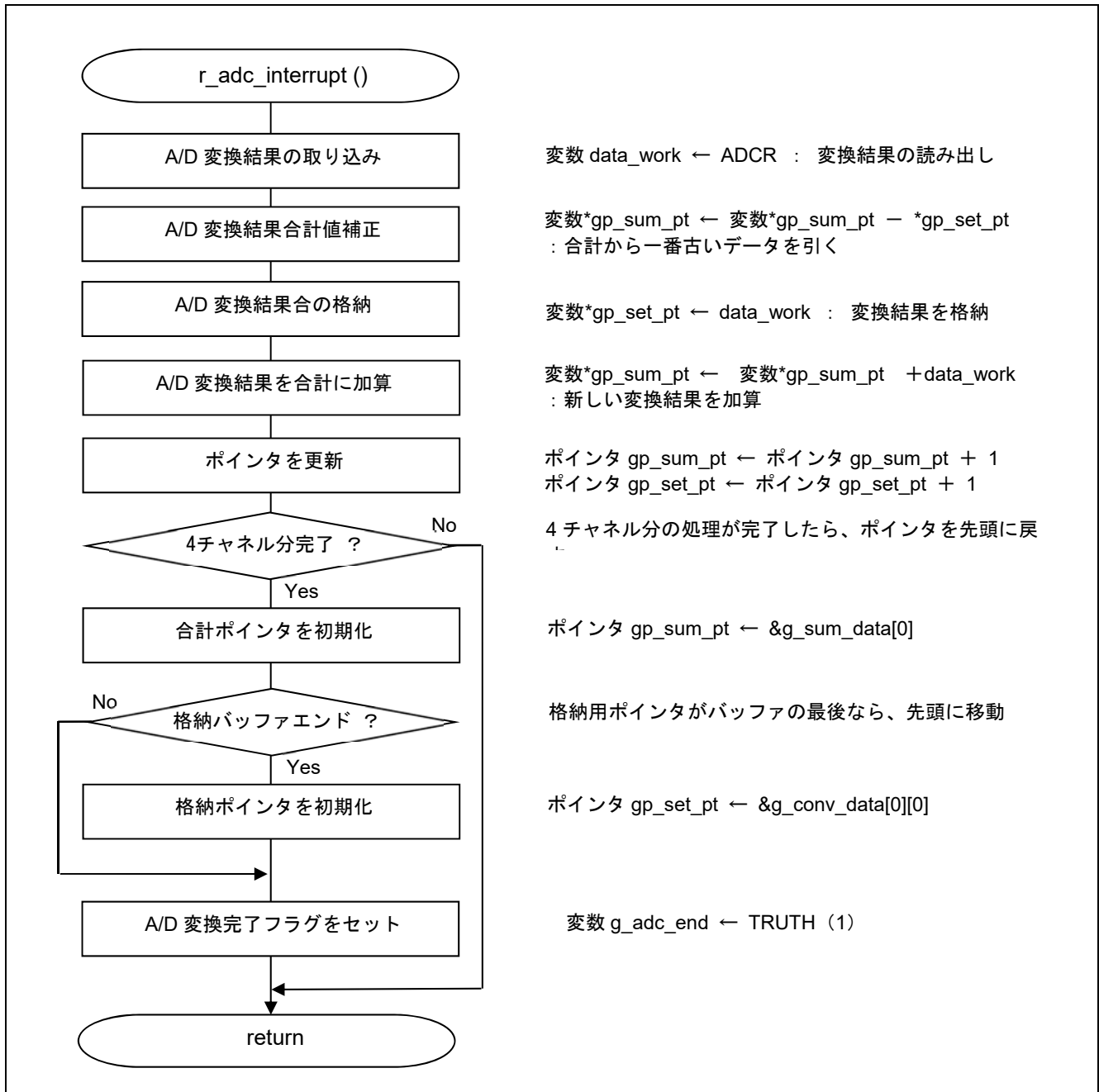


図 5.13 A/D 変換完了割り込み処理

A/D 変換結果の取り込み

- ・ 12 ビット A/D 変換結果レジスタ (ADCR)
- A/D 変換結果を読み出します

略号 : ADCR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

5.7.13 LED 点灯の初期化処理

図 5.14 に LED 点灯の初期化処理のフローチャートを示します。

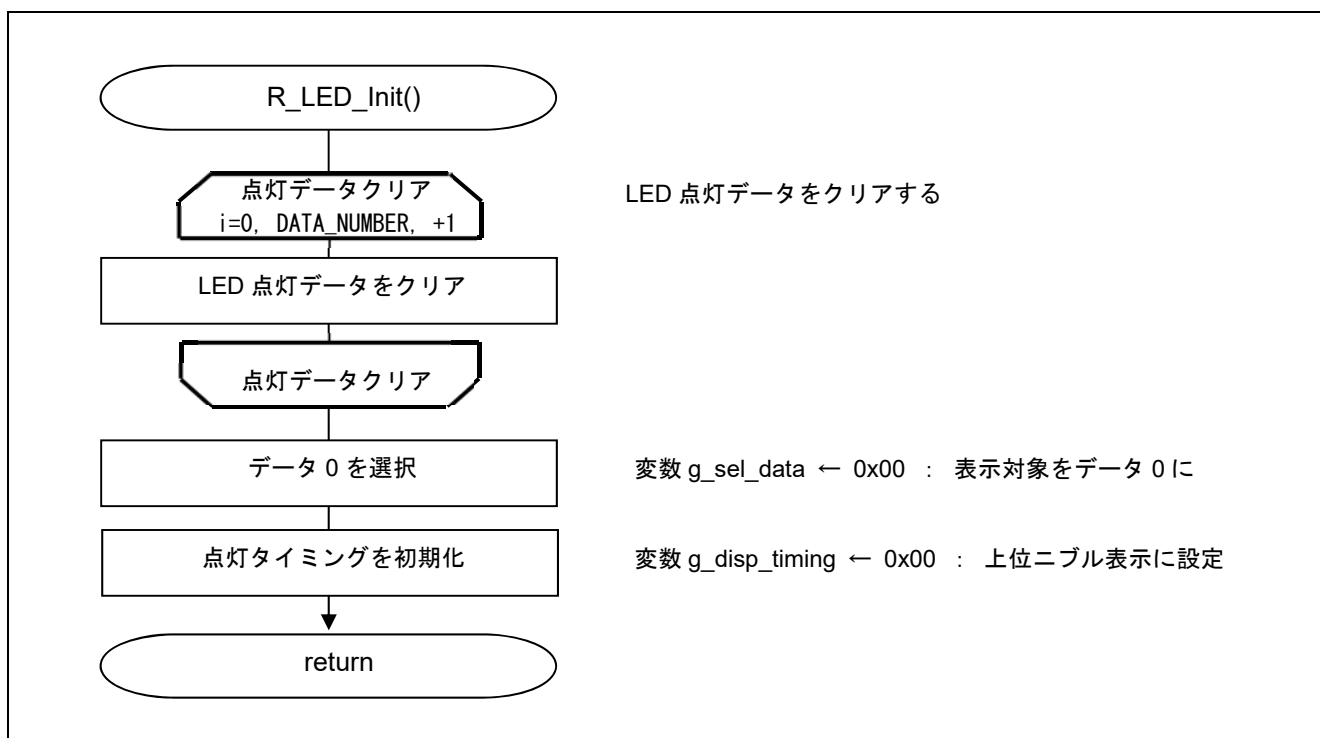


図 5.14 LED 点灯の初期化処理

5.7.14 TM03 の起動処理

図 5.15 に TM03 の起動処理のフローチャートを示します。

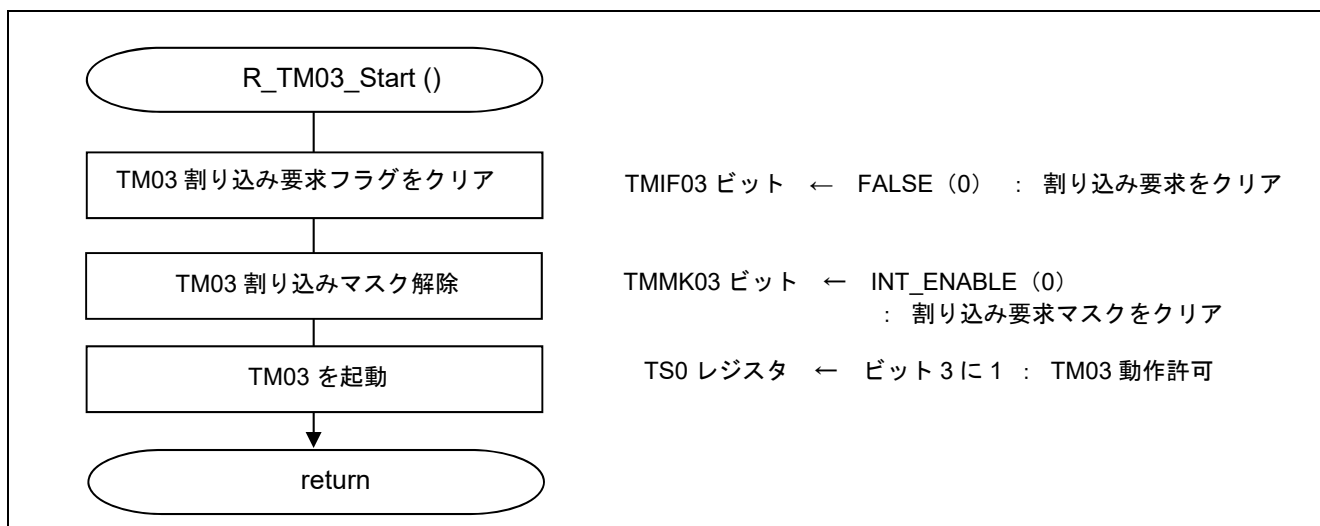


図 5.15 TM03 の起動処理

TM03 割り込みの設定

- ・ 割り込み要求フラグ・レジスタ 1L (IF1L) の TMIF03 ビット
割り込み要求のクリア
- ・ 割り込み要求マスク・レジスタ 1L (MK1L) の TMMK03 ビット
割り込み要求マスクの解除

ビット5

TMIF03	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

ビット5

TMMK03	割り込み要求フラグ
0	割り込み処理許可
1	割り込み処理禁止

TM03 起動

- ・タイマ・チャンネル開始レジスタ 0 (TS0)
TM03 の動作を許可します

略号 : TS0L

7	6	5	4	3	2	1	0
TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00
0	0	0	0	1	0	0	0

ビット3

TS03	チャンネル3の動作許可（スタート）トリガ
0	トリガ動作しない
1	TE03 ビットを1にセットし、カウント動作許可状態になる

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

5.7.15 LED の発光データの設定処理

図 5.16 に LED の発光データの設定処理のフローを示します。

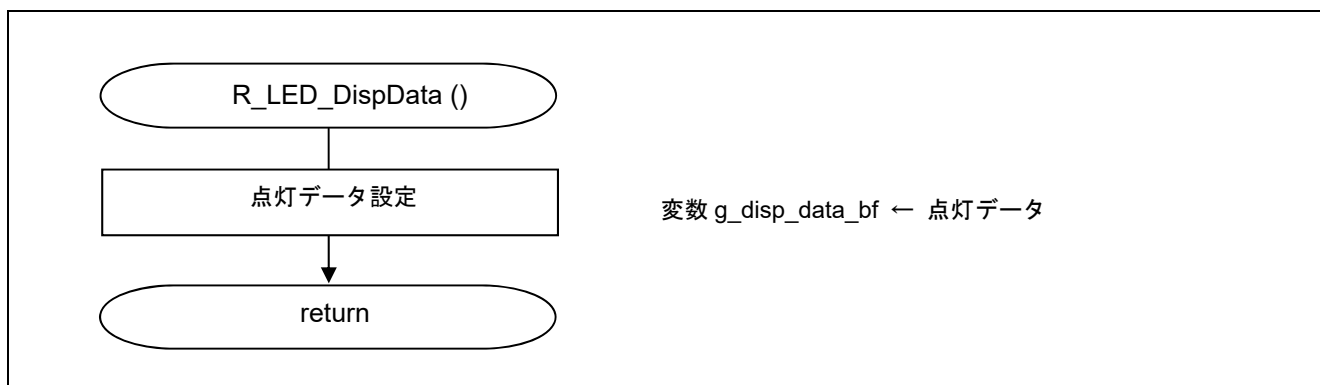


図 5.16 LED の発光データの設定処理

5.7.16 5ms インターバル・タイマ割り込み処理

図 5.17 に 5ms インターバル・タイマ割り込み処理のフローを示します。

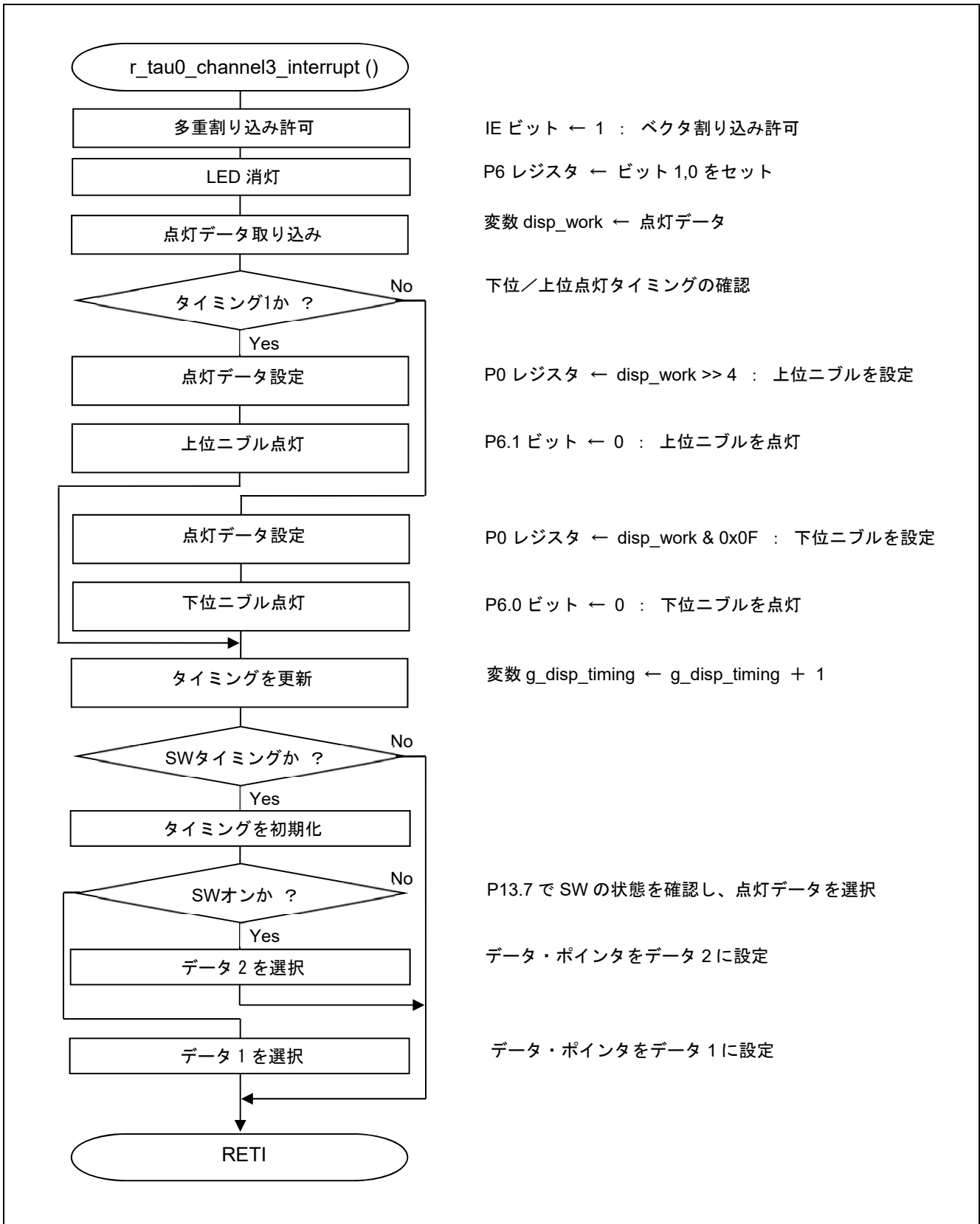


図 5.17 5ms インターバル・タイマ割り込み処理

5.7.17 I2C 通信の初期化処理

図 5.18 に I2C 通信の初期化処理のフローチャートを示します。

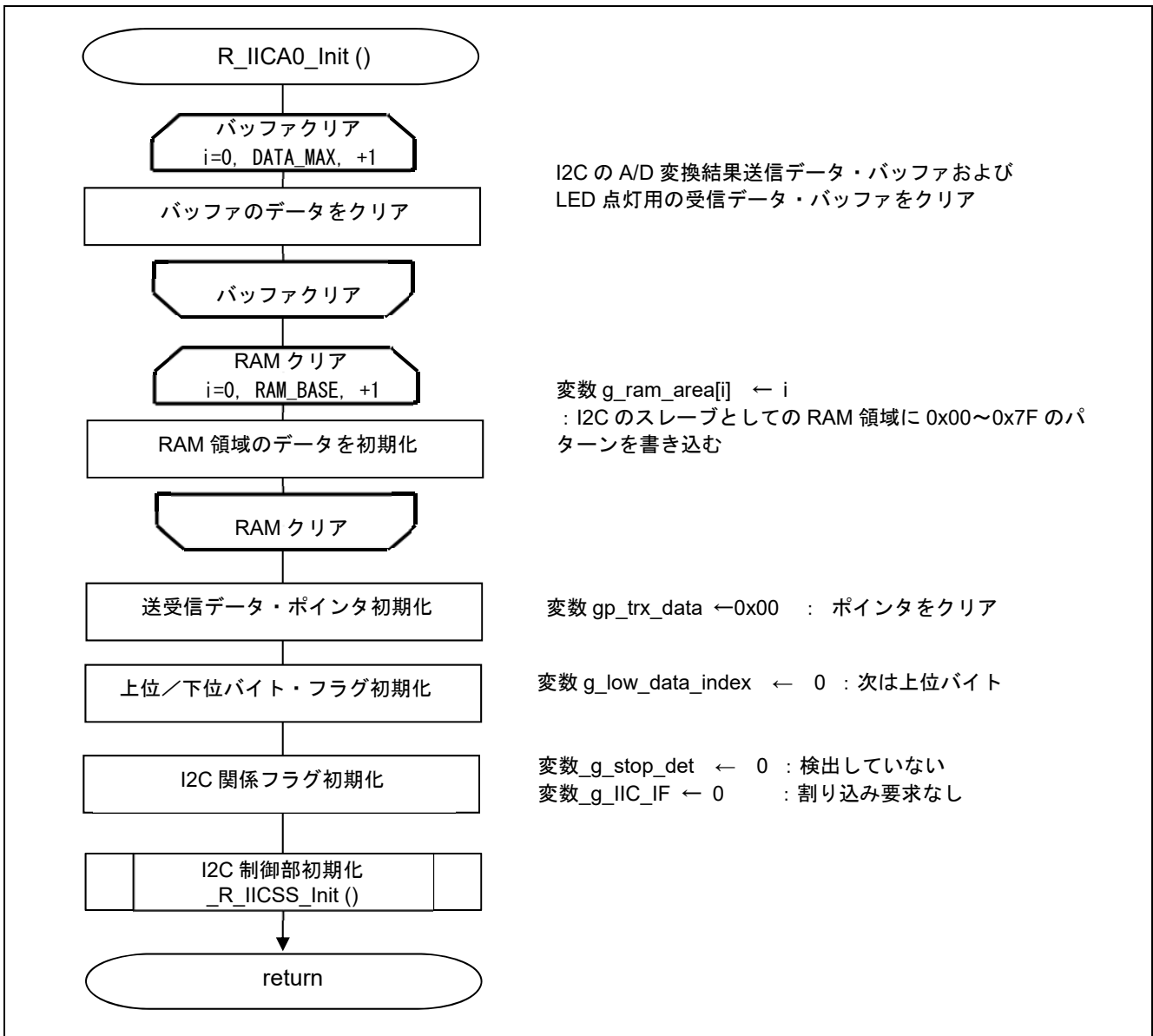


図 5.18 I2C 通信の初期化処理

5.7.18 I2C 通信状態の確認

図 5.19 に I2C 通信状態の確認処理のフローチャートを示します。

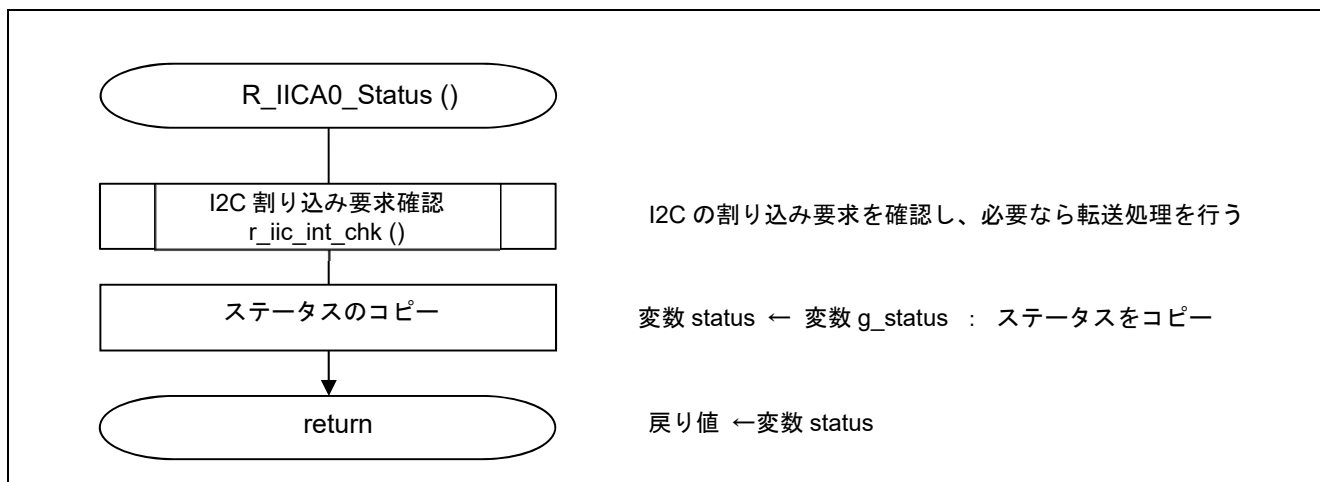


図 5.19 I2C 通信状態の確認処理

5.7.19 I2C 受信データを読み出し

図 5.20 に I2C 受信データを読み出し処理のフローチャートを示します。

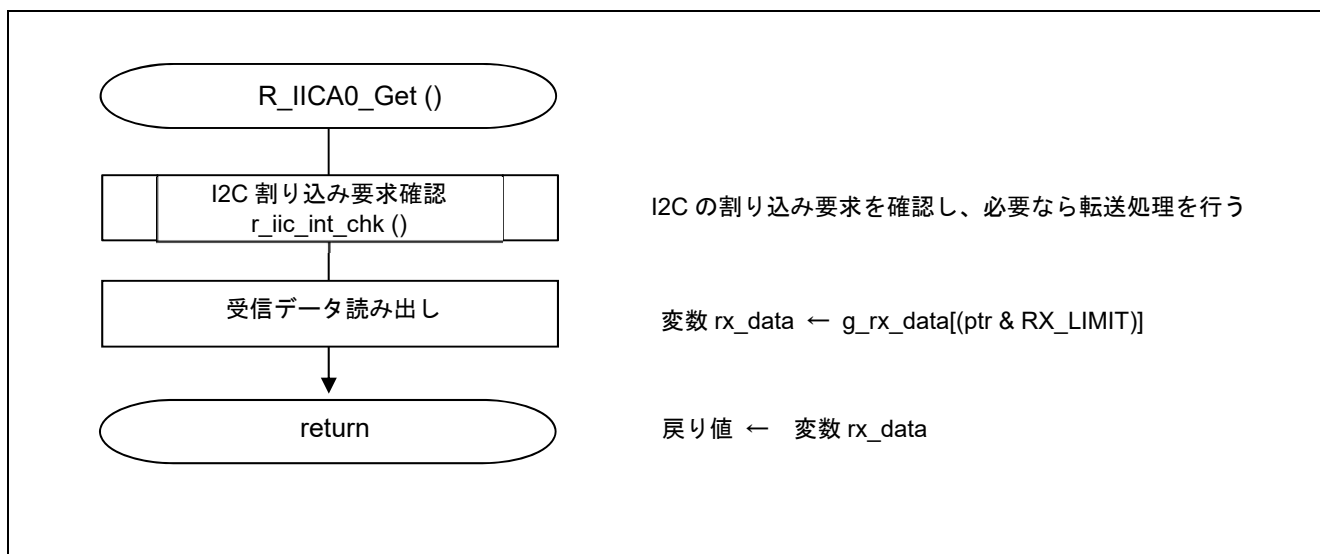


図 5.20 I2C 受信データを読み出し処理

5.7.20 I2C 送信バッファへのデータ設定処理

図 5.21 に I2C 送信バッファへのデータ設定処理のフローチャートを示します。

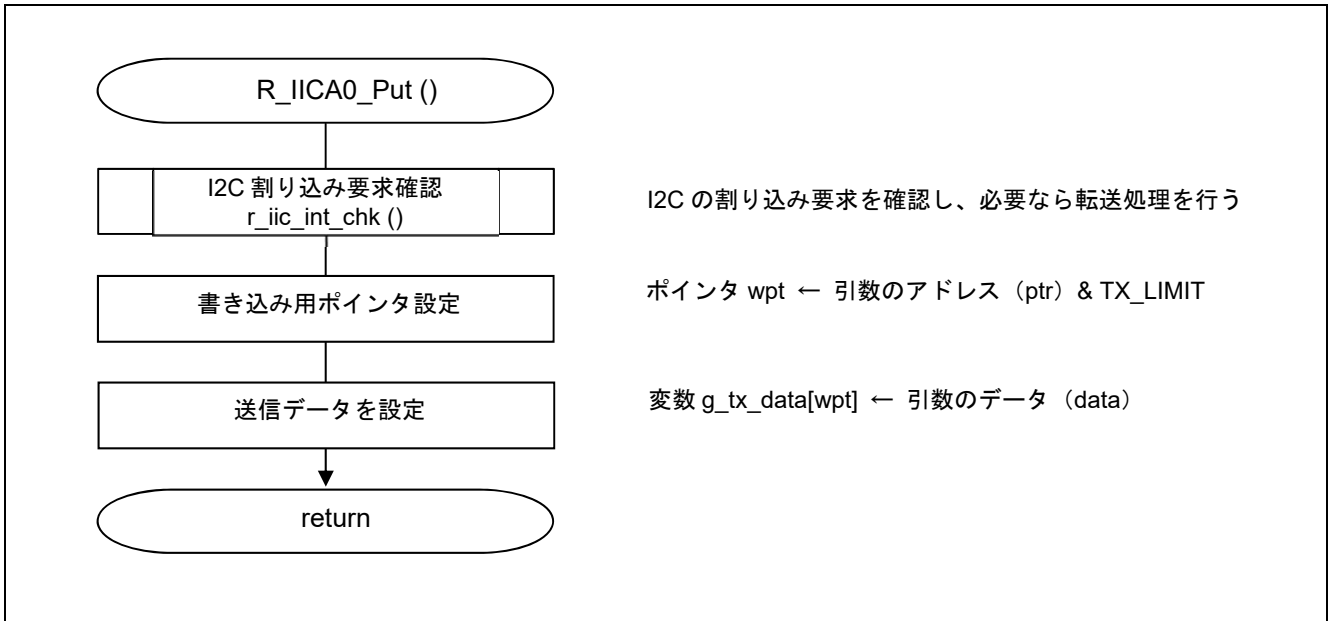


図 5.21 I2C 送信バッファへのデータ設定処理

5.7.21 I2C 通信完了割り込み要求確認処理

図 5.22 に I2C 通信完了割り込み要求確認処理のフローチャートを示します。

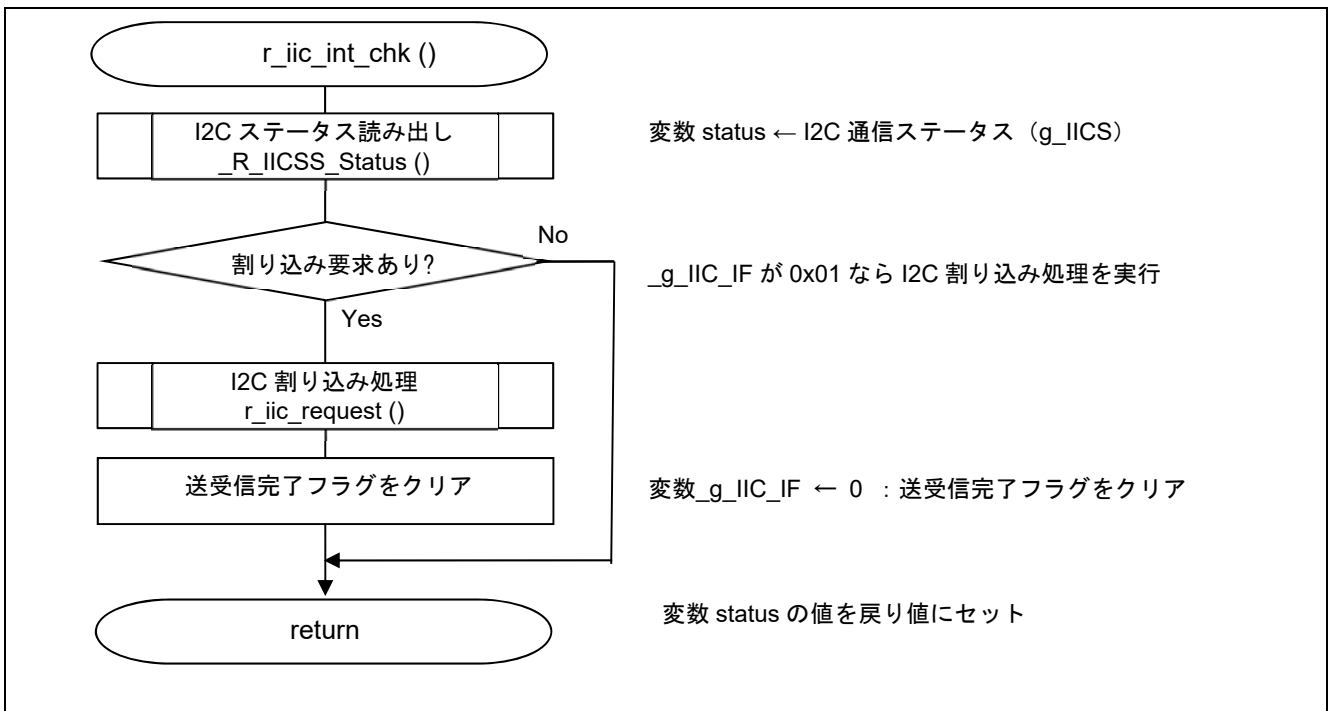


図 5.22 I2CA0 通信完了割り込み受信処理

5.7.22 I2C 通信完了割り込み処理

図 5.23～図 5.26 に I2C 通信完了割り込み処理のフローチャートを示します。

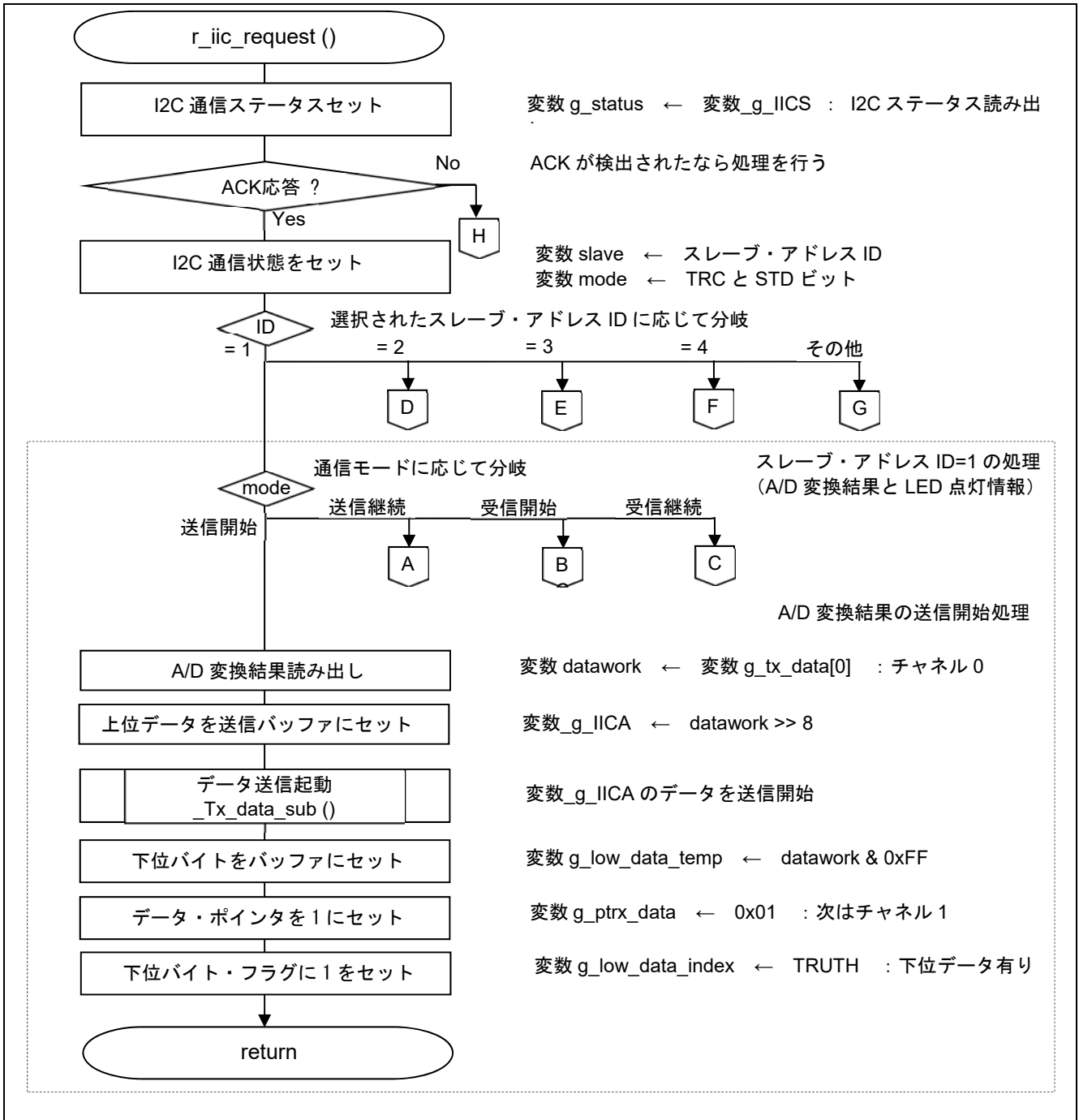


図 5.23 I2C 通信完了割り込み送信処理 (1/4)

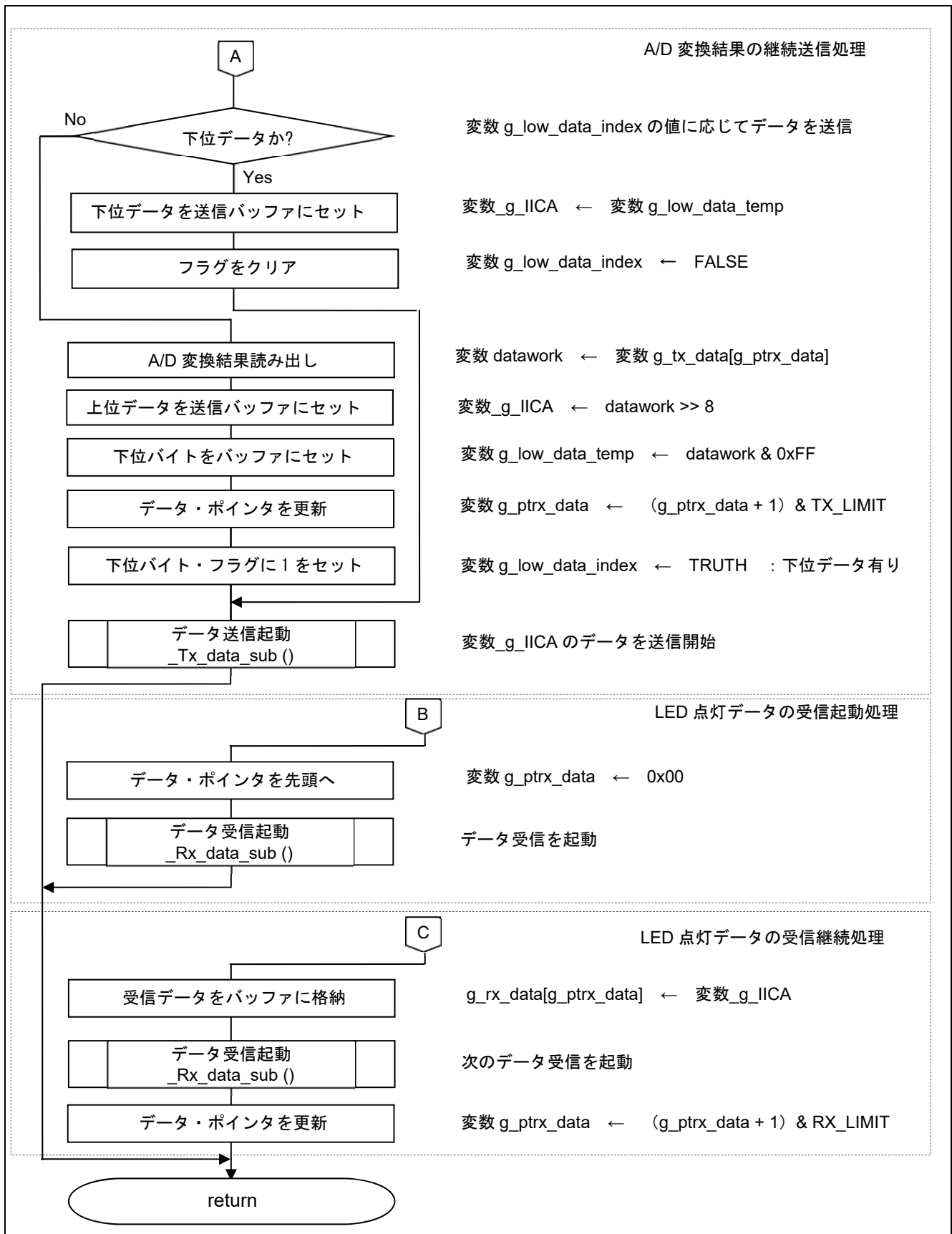


図 5.24 I2C 通信完了割り込み処理 (2/4)

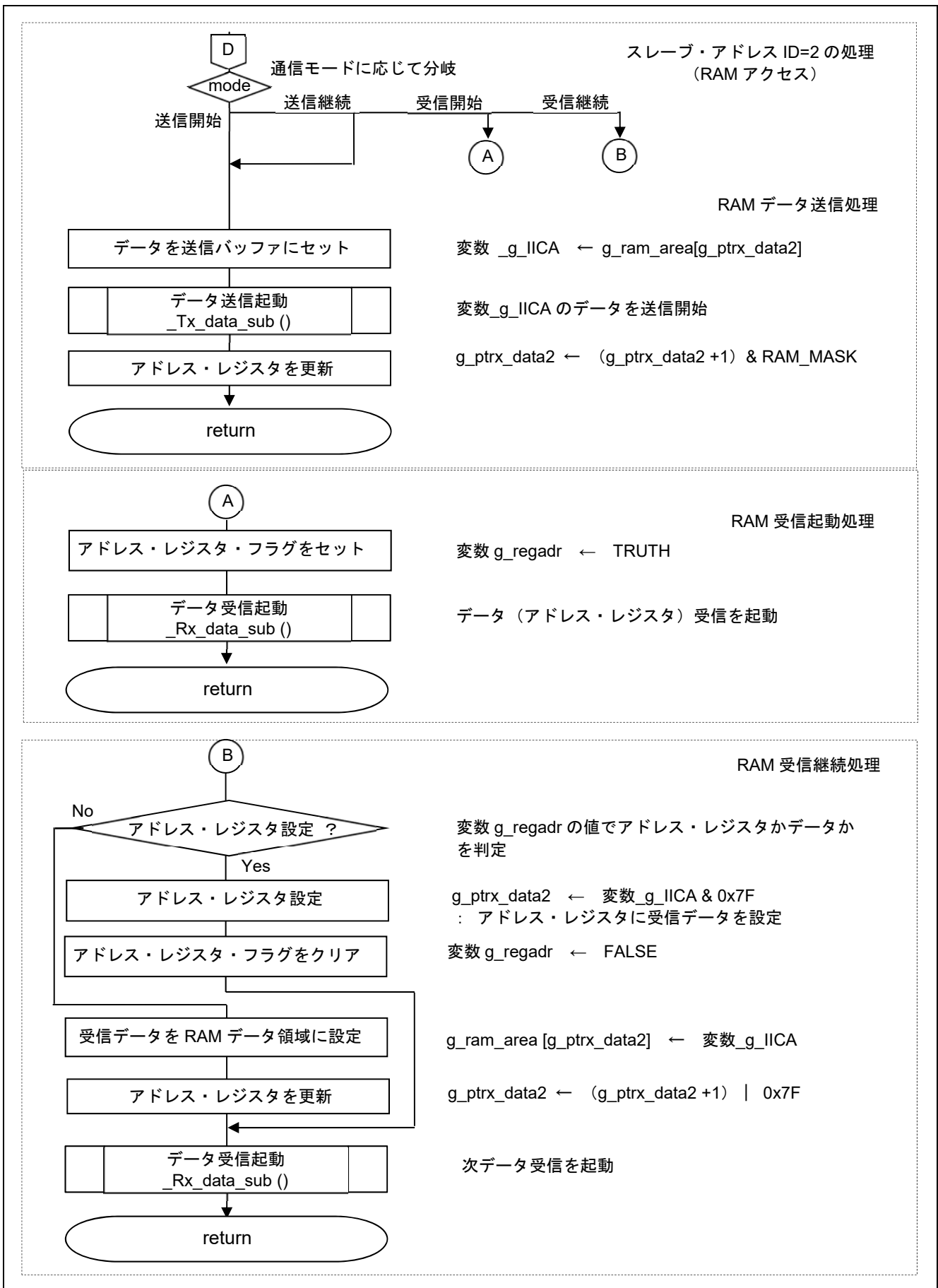


図 5.25 I2C 通信完了割り込み処理 (3/4)

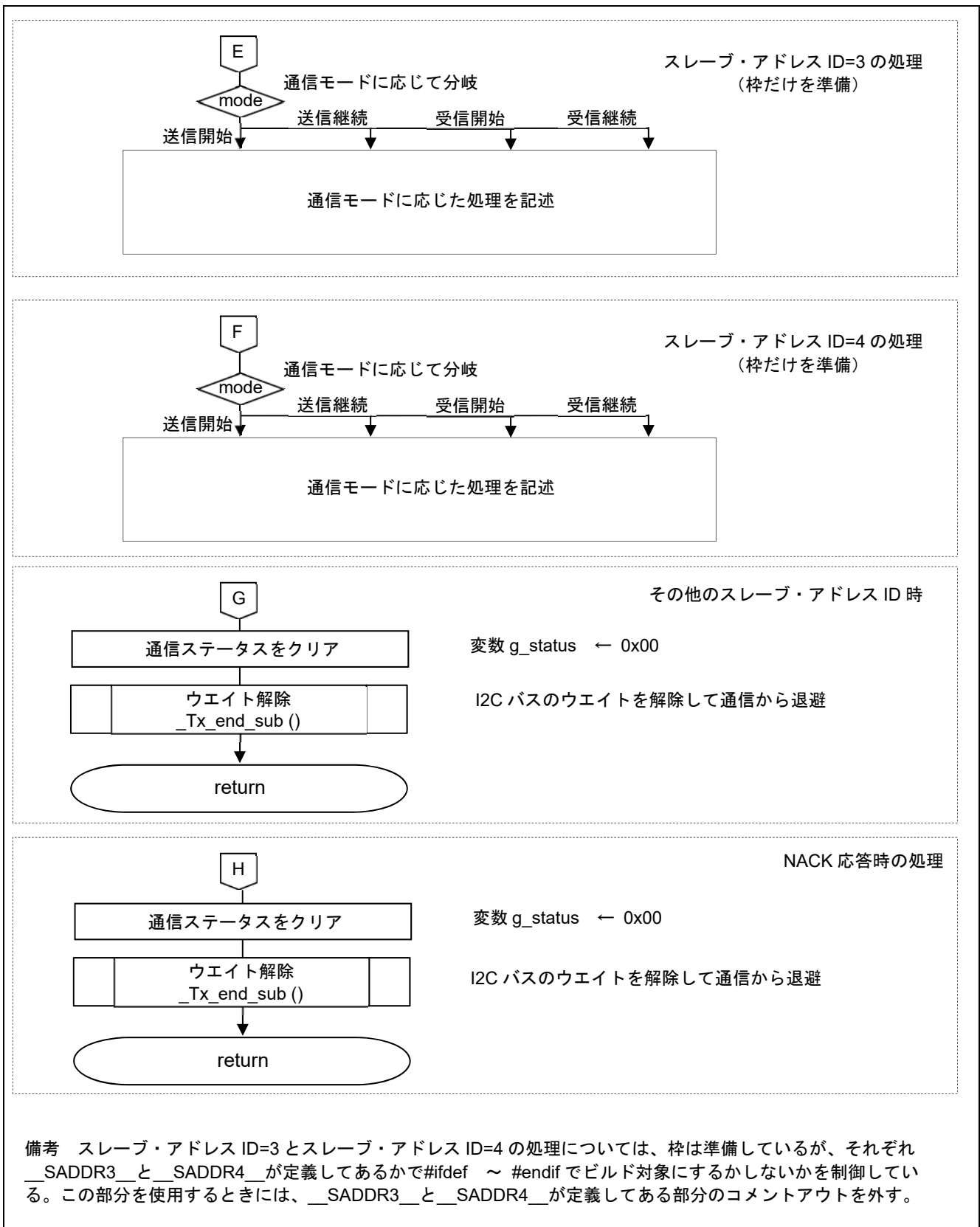


図 5.26 I2C 通信完了割り込み処理 (4/4)

5.7.23 I2C 初期化処理（アセンブラ部）

図 5.27 に I2C 初期化処理（アセンブラ部）のフローチャートを示します。

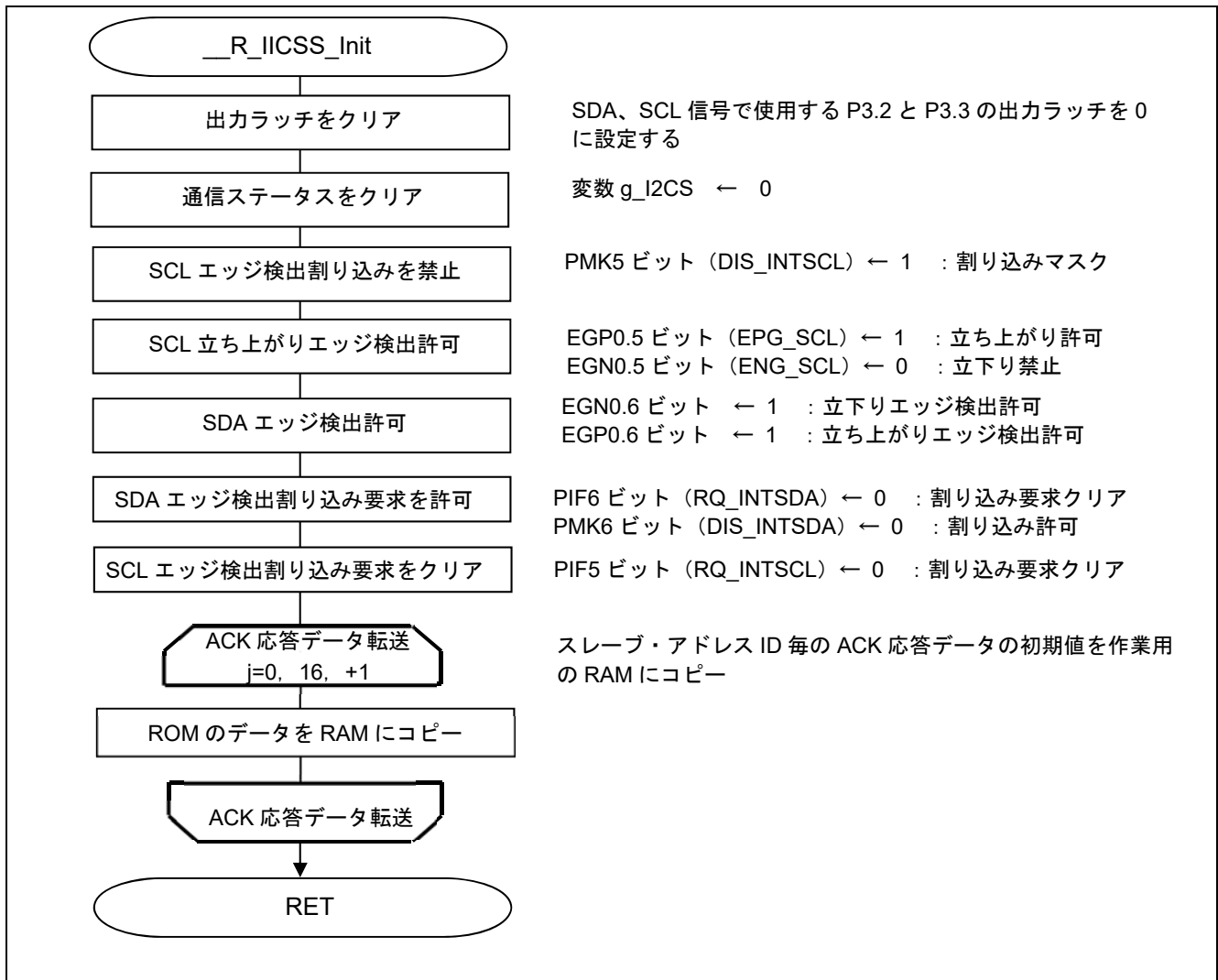


図 5.27 I2C 初期化処理（アセンブラ部）

出力ラッチクリア

- ・ポート・レジスタ 3(P3)
P3.3 および P3.2 をクリアします

略号：P3

7	6	5	4	3	2	1	0
0	0	0	0	P33	P32	P31	P30
0	0	0	0	0	0	x	x

ビット 3、2

P3n	出力ラッチへの書き込みデータ
0	0 を設定
1	1を設定

注意 レジスタ設定方法の詳細については、RL78/I1D 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。

SCL エッジ検出割り込みを禁止

- ・割り込みマスク・フラグ・レジスタ 0L(MK0L)の PMK5 ビット
INTP5 割り込み要求をマスク

ビット 7

PMK5	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

SCL、SDA エッジ検出設定

- ・外部割り込み立ち上がりエッジ許可レジスタ(EGP0)
EGP6 ビット、EGP5 ビットをセット
- ・外部割り込み立下りエッジ許可レジスタ(EGN0)
EGN6 ビットをセット、EGN5 ビットをクリア

略号：EGP0

	7	6	5	4	3	2	1	0
0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0	
0	1	1	x	x	x	x	x	x

略号：EGN0

	7	6	5	4	3	2	1	0
0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0	
0	1	0	x	x	x	x	x	x

ビット 6、5

EGPn	EGNn	INTPn 端子の有効エッジの選択
0	0	エッジ検出禁止
0	1	立ち下がリエッジ
1	0	立ち上がりエッジ (INTP5 : SCL エッジ)
1	1	立ち上がり、立ち下がりの両エッジ (INTP6 : SDA エッジ)

割り込み要求をクリア

- ・割り込み要求フラグ・レジスタ(IF0L, IF0H)
INTP5 と INTP6 の割り込み要求をクリア

略号：IF0L

	7	6	5	4	3	2	1	0
PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF	
0	x	x	x	x	x	x	x	x

略号：IF0H

	7	6	5	4	3	2	1	0
RTITIF	TMIF00	SREIF0	0	0	SRIF0	STIF0	PIF6	
x	x	x	0	0	x	x	0	0

ビット n

PIFn	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

SDA エッジ検出割り込みを許可

- ・ 割り込みマスク・フラグ・レジスタ 0H(MK0H)の PMK6 ビット
INTP6 割り込み要求のマスクを解除

略号 : MK0H

7	6	5	4	3	2	1	0
RTITMK	TMMK00	SREMK0	0	0	SRMK0	STMK0	PMK6
x	x	x	0	0	x	x	0

ビット 0

PMK6	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

5.7.24 I2C 通信ステータス読み出し処理

図 5.28 に I2C 通信ステータス読み出し処理のフローチャートを示します。

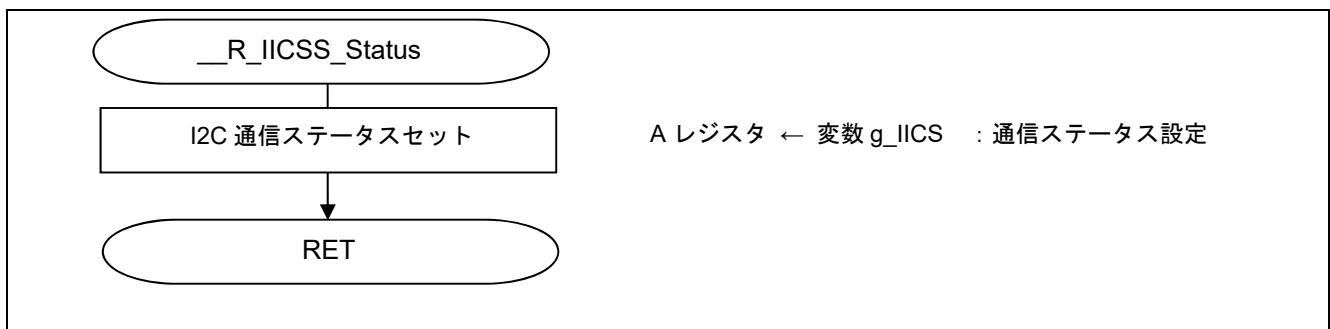


図 5.28 I2C 通信ステータス読み出し処理

5.7.25 ACK 応答設定

図 5.29 に ACK 応答設定処理のフローチャートを示します。

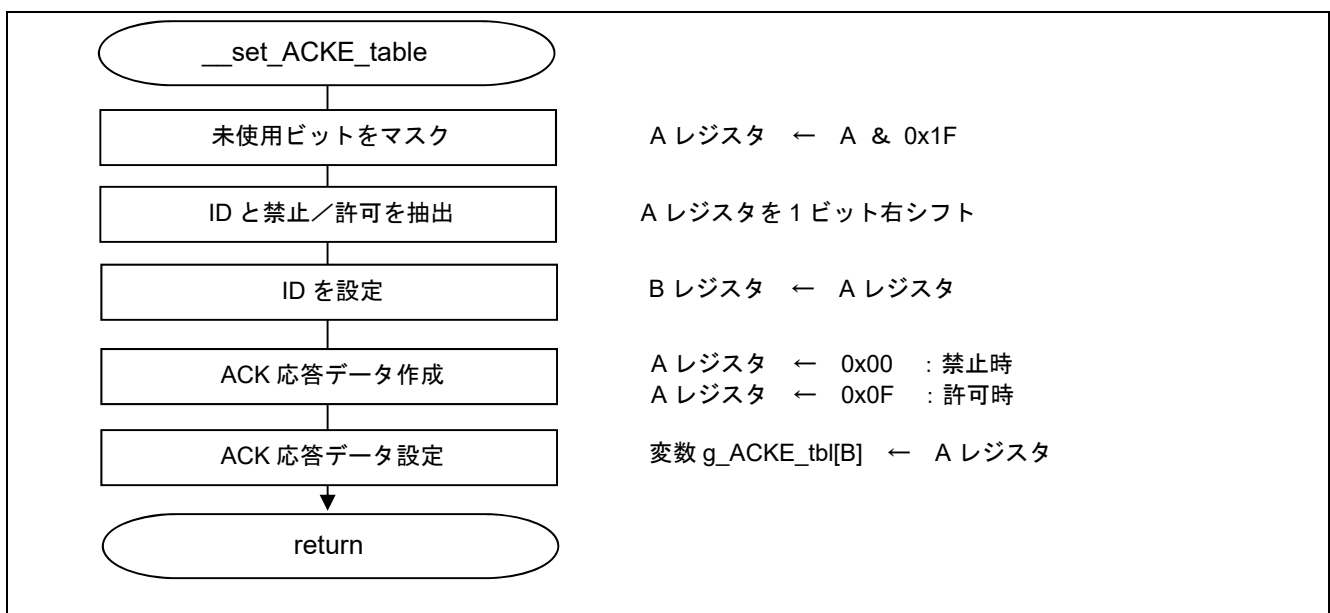


図 5.29 ACK 応答設定処理

5.7.26 ACK 応答読み出し

図 5.30 に ACK 応答読み出し処理のフローチャートを示します。

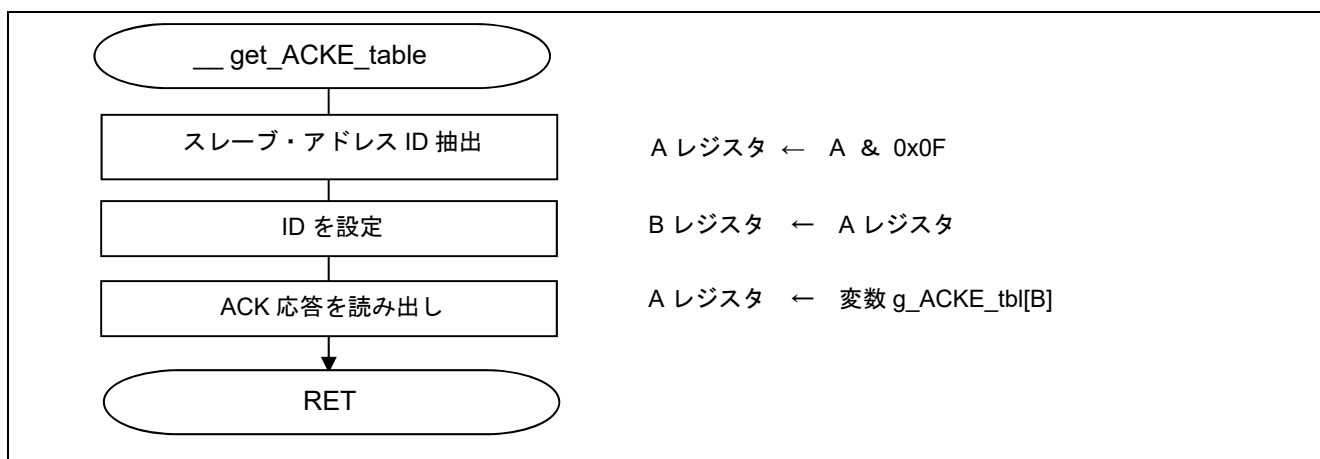


図 5.30 ACK 応答読み出し処理

5.7.27 SCL エッジ検出割り込みエントリ処理

図 5.31 に SCL エッジ検出割り込みエントリ処理のフローチャートを示します。

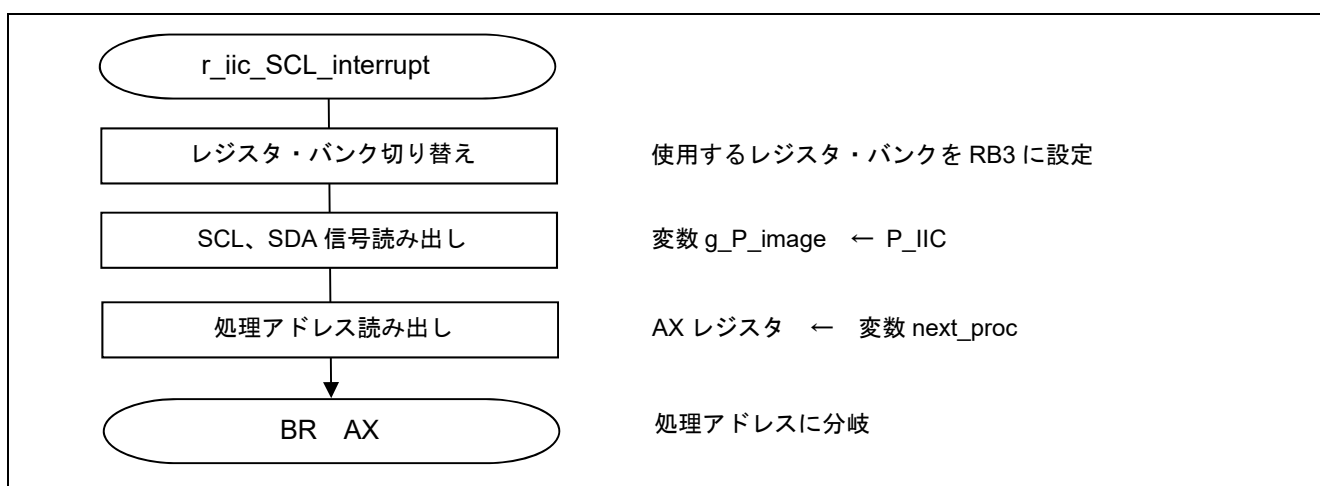


図 5.31 SCL エッジ検出割り込みエントリ処理

ここは、SCL エッジ検出割り込みのエントリ処理だけで、実際の処理は 5.8.29 SCL エッジ検出割り込み処理で説明します。

SCL、SDA 信号読み出し

- ・ポート・レジスタ 3(P3)
- SCL 信号、SDA 信号の読み出し

略号 : P3

7	6	5	4	3	2	1	0
0	0	0	0	SCL	SDA	P31	P30

5.7.28 SDA エッジ検出割り込み処理

図 5.32~図 5.33 に SDA エッジ検出割り込み処理のフローチャートを示します。

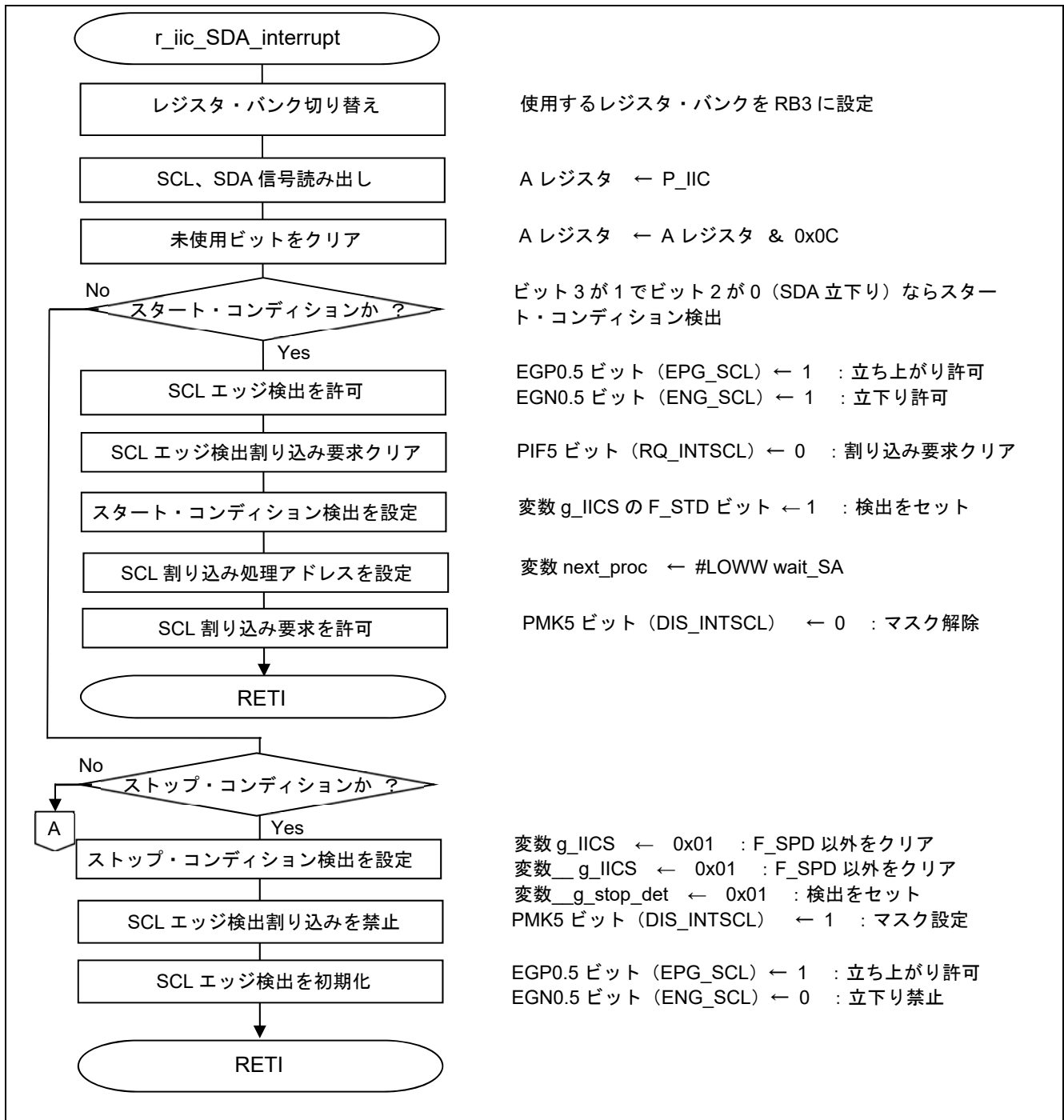


図 5.32 SDA エッジ検出割り込み処理 (1/2)

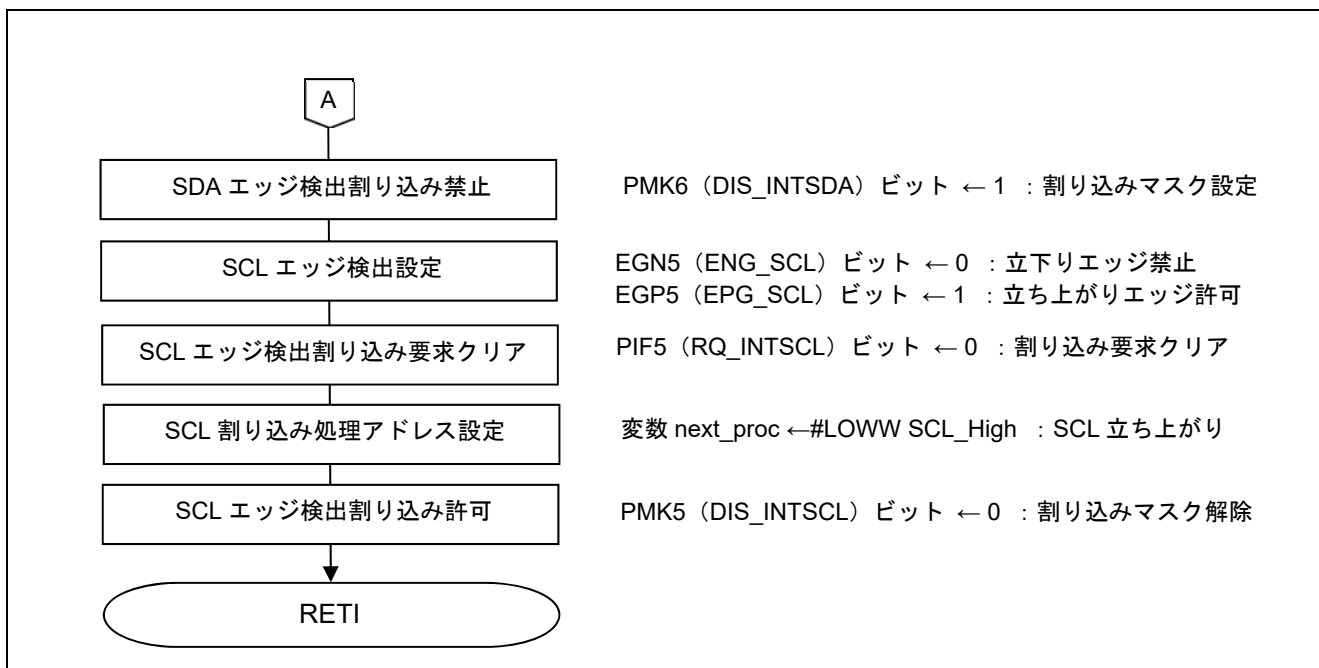


図 5.33 SDA エッジ検出割り込み処理 (2/2)

以下の制御レジスタは SCL/SDA エッジ検出割り込み処理で共通に使用するので、ここでまとめて記述しておきます。

SCL、SDA 信号読み出し

・ポート・レジスタ 3(P3)

SCL 信号、SDA 信号の読み出し

略号 : P3

7	6	5	4	3	2	1	0
0	0	0	0	SCL	SDA	P31	P30

エッジ検出の設定

・外部割り込み立ち上がりエッジ許可レジスタ(EGP0)

INTP5 のエッジ検出許可/禁止

・外部割り込み立ち下がりエッジ許可レジスタ(EGN0)

INTP5 のエッジ検出許可/禁止

略号 : EGP0

7	6	5	4	3	2	1	0
0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0
0x	1	1	x	x	x	x	x

略号 : EGN0

7	6	5	4	3	2	1	0
0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0
0	1	1/0	x	x	x	x	x

ビット5

EGP5	EGN5	INTP5 端子の有効エッジの選択
0	0	エッジ検出禁止
0	1	立ち下がりエッジ
1	0	立ち上がりエッジ
1	1	立ち上がり, 立ち下がりの両エッジ

SCL、SDA エッジ検出割り込みの制御

- ・割り込みマスク・フラグ・レジスタ（MK0L、MK0H）
割り込み要求を制御します
- ・割り込み要求フラグ・レジスタ（IF0L、IF0H）
割り込み要求を制御します

略号：MK0L

7	6	5	4	3	2	1	0
PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK
0/1	x	x	x	x	x	x	x

略号：MK0H

7	6	5	4	3	2	1	0
RTITMK	TMMK00	SREMK0	1	1	SRMK0	STMK0	PMK6
x	x	x	1	1	x	x	0/1

ビット n

PMKn	割り込み処理の制御
0	割り込み処理許可
1	割り込み処理禁止

略号：IF0L

7	6	5	4	3	2	1	0
PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF
0	0	0	0	0	0	x	x

略号：IF0H

7	6	5	4	3	2	1	0
RTITIF	TMIF00	SREIF0	0	0	SRIF0	STIF0	PIF6
x	x	x	0	0	x	x	0

ビット n

PIFn	割り込み要求フラグ
0	割り込み要求信号が発生していない
1	割り込み要求信号が発生し、割り込み要求状態

5.7.29 SCL エッジ検出割り込み処理

図 5.34～図 5.52 に SCL エッジ検出割り込み処理のフローチャートを示します。

(0) ストップ/スタート・コンディション待ち処理

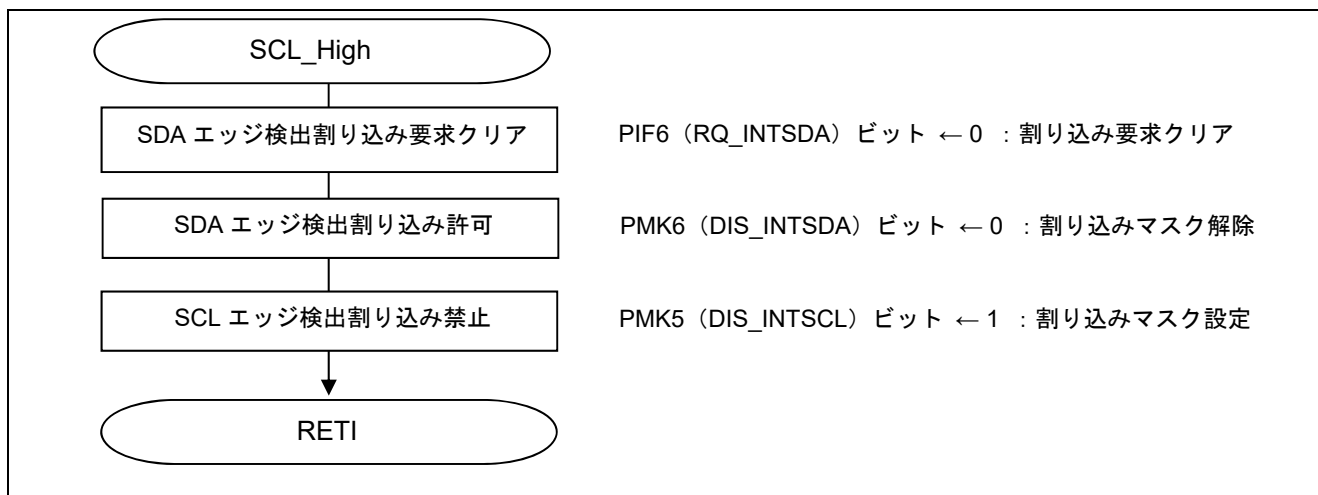


図 5.34 ストップ/スタート・コンディション待ち処理

(1) スレーブ・アドレス受信開始待ち処理 (sequence(4))

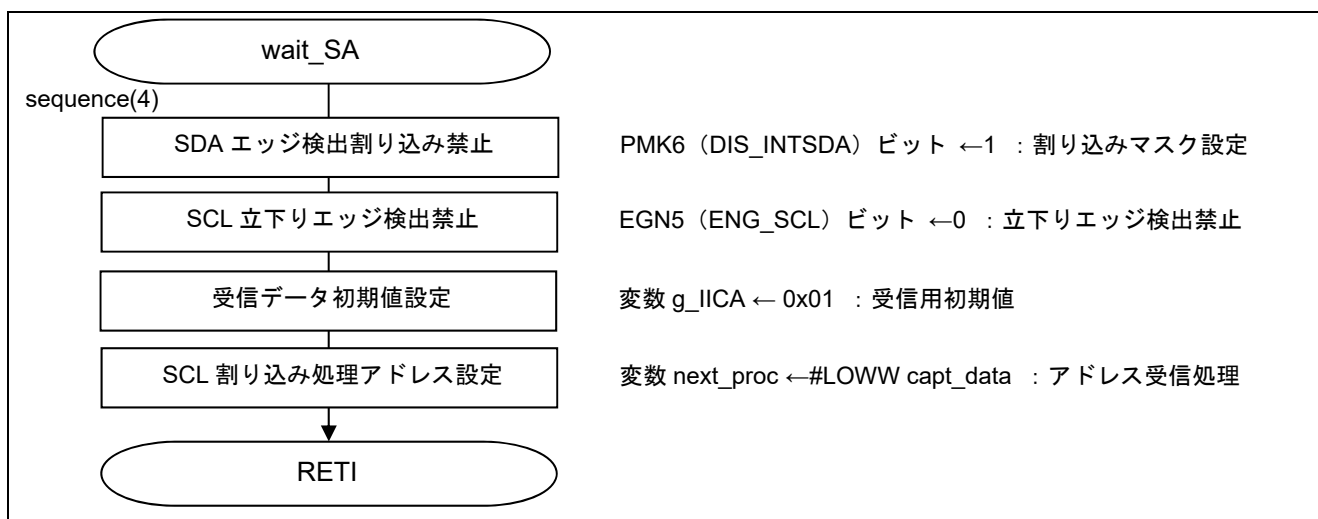


図 5.35 スレーブ・アドレス受信開始待ち処理

(2) スレーブ・アドレス受信処理(sequence(5))

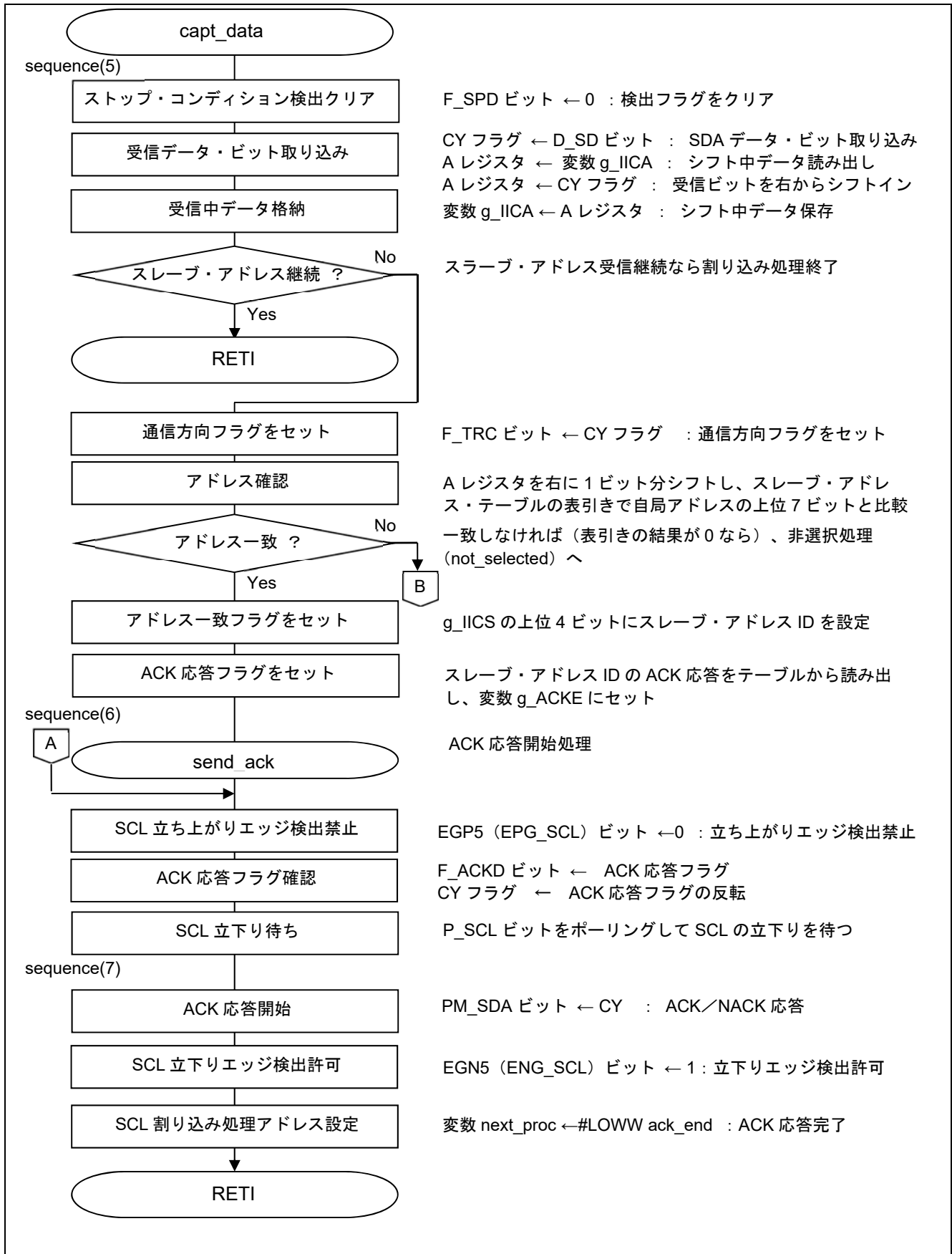


図 5.36 スレーブ・アドレス受信処理

(3) 非選択時処理（sequence(6)'）

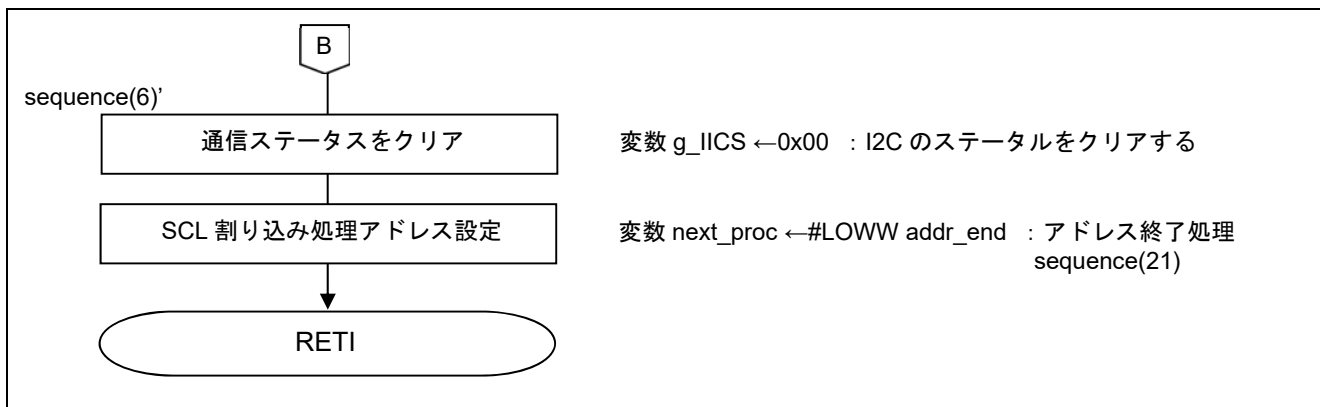


図 5.37 非選択時処理

(4) ACK 応答終了処理（sequence(8)）

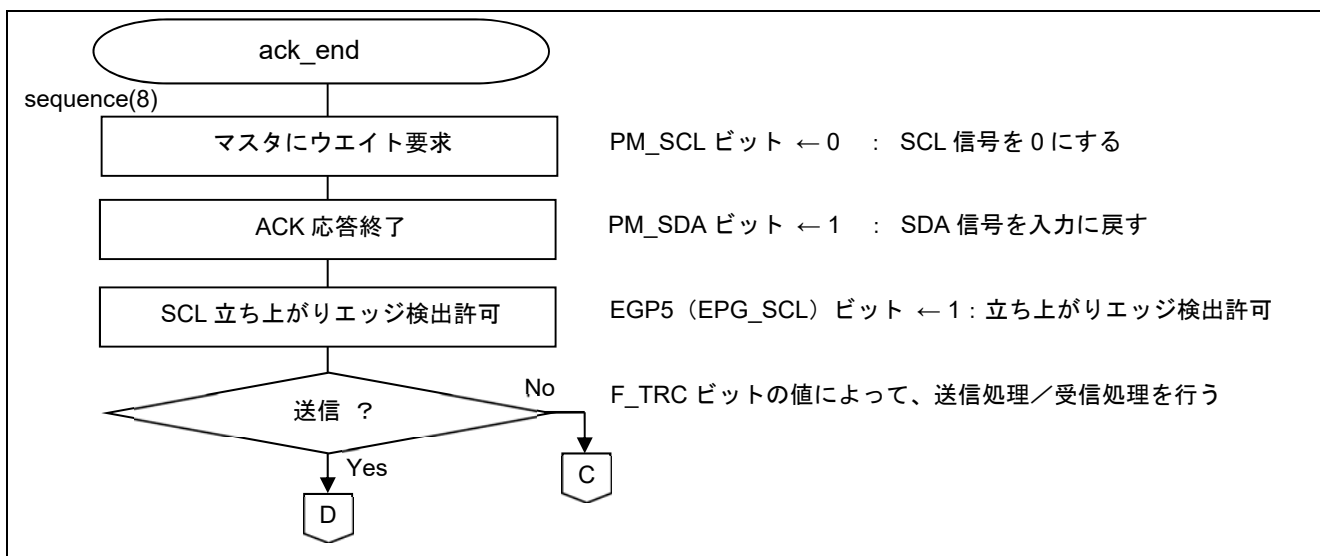


図 5.38 ACK 応答終了処理

(5) 受信完了（動作開始要求）処理（sequence(8)'）

1バイトのデータの受信が完了したら、上位ソフトに通信完了フラグをセットして処理を完了します。

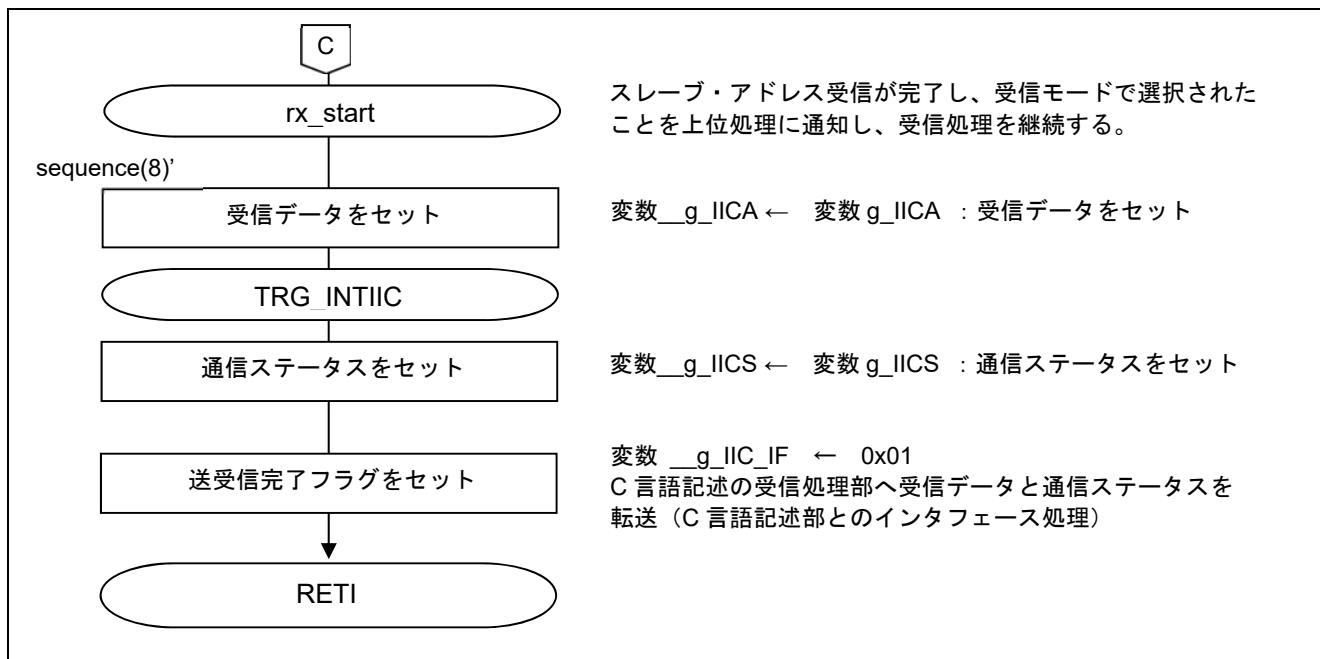


図 5.39 受信完了処理

(6) データ受信処理（sequence(9)）

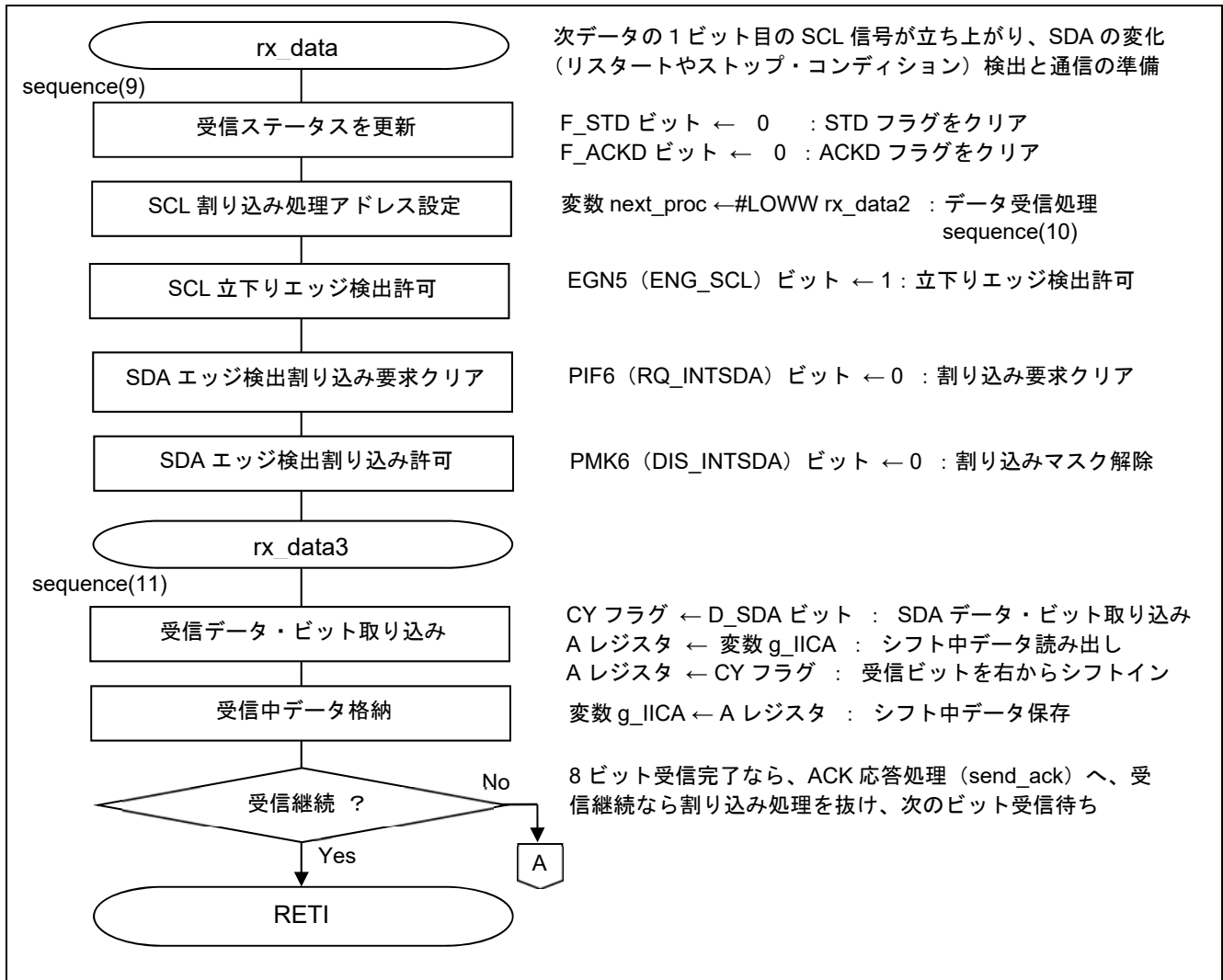


図 5.40 データ受信処理

(7) 受信動作開始処理（sequence(10)）

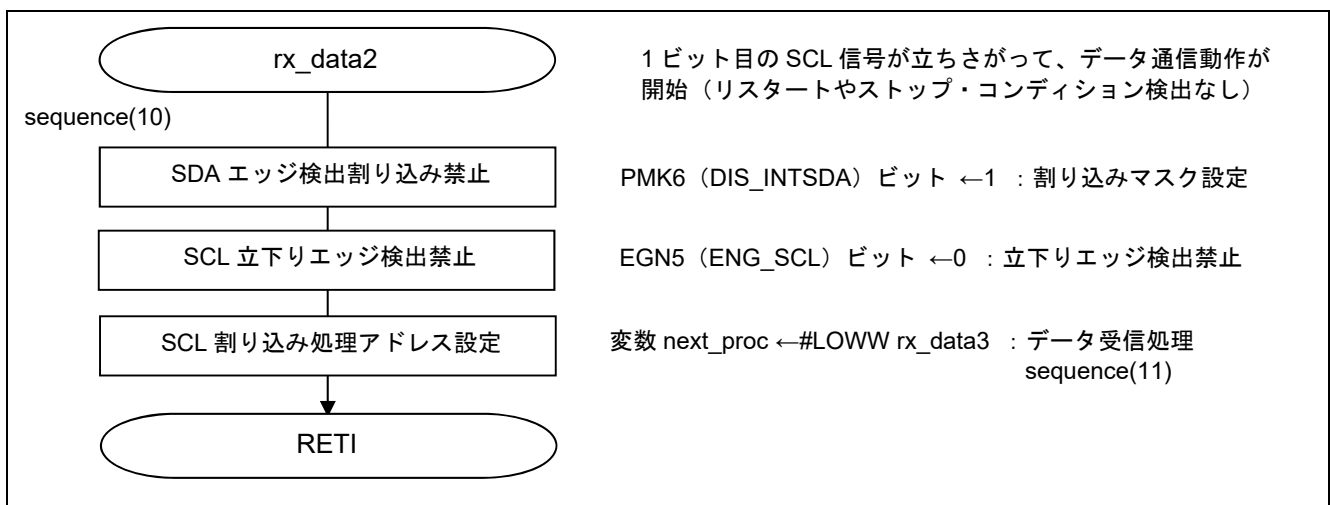


図 5.41 受信動作開始処理

(8) 送信動作開始要求処理（sequence(14)）

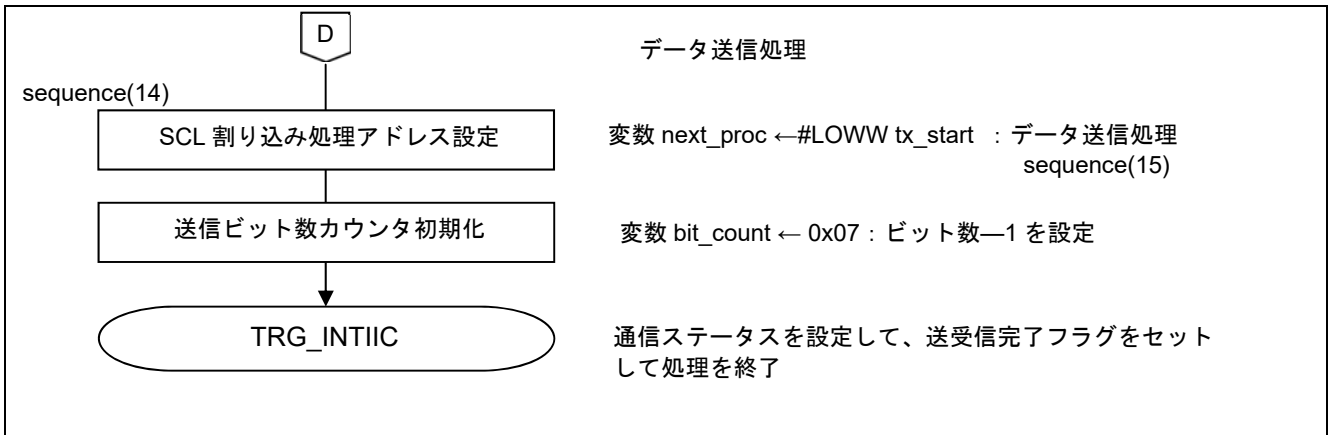


図 5.42 送信動作開始要求処理

(9) 送信動作開始処理（sequence(15)）



図 5.43 送信動作開始処理

(10) 送信処理 (sequence(16))

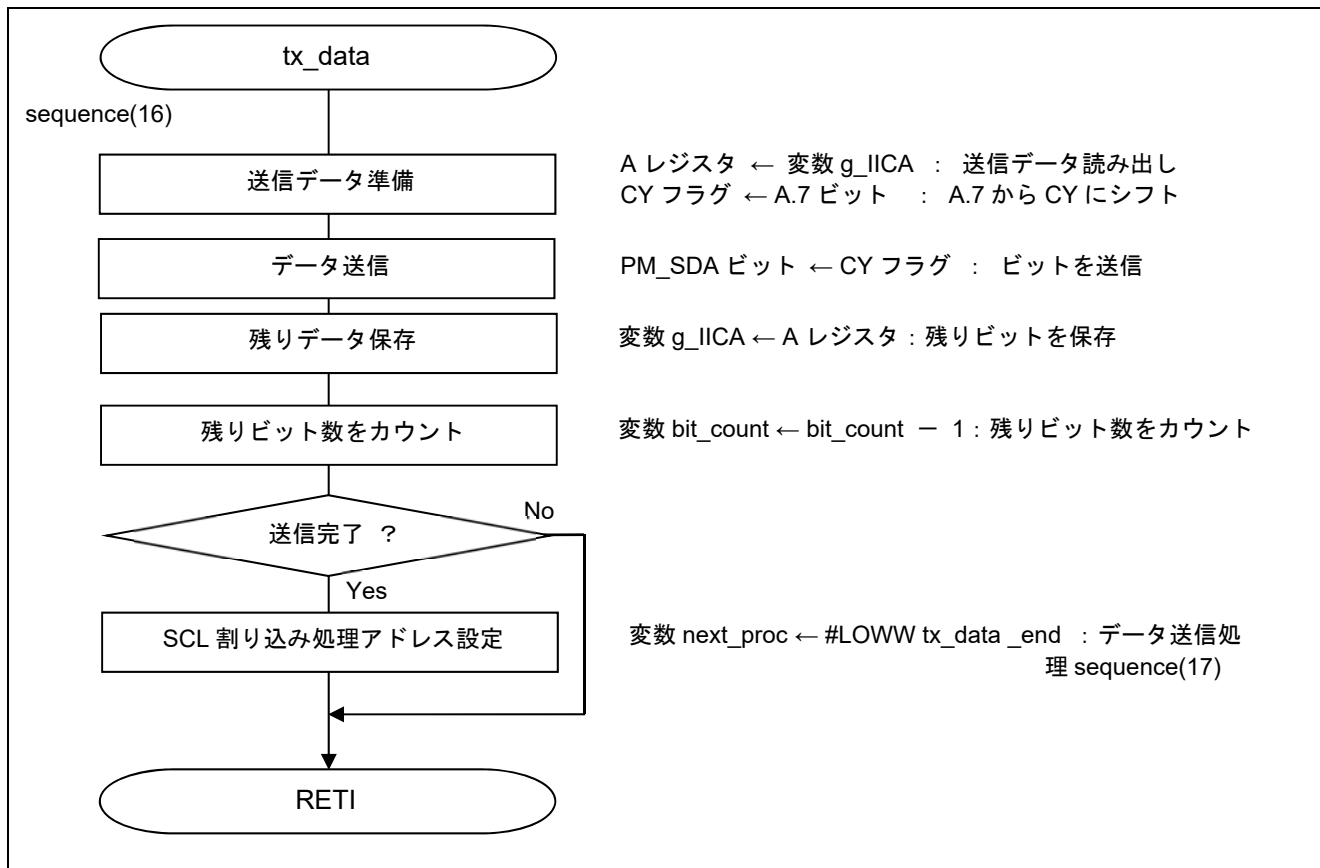


図 5.44 送信処理

(11) 送信終了処理 (sequence(17))

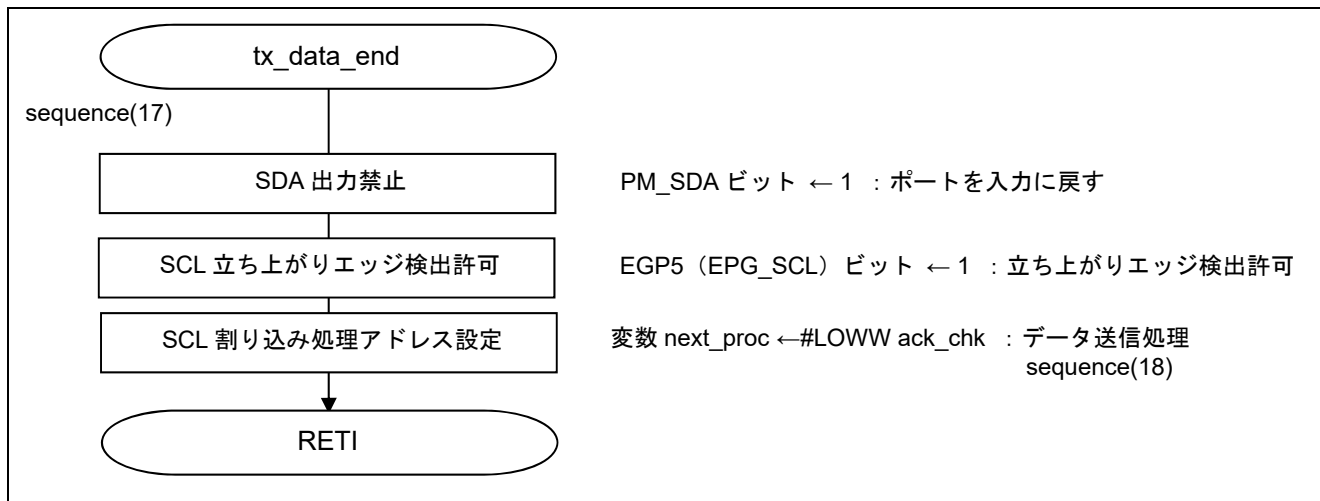


図 5.45 送信終了処理

(12) ACK 確認処理 (sequence(18))

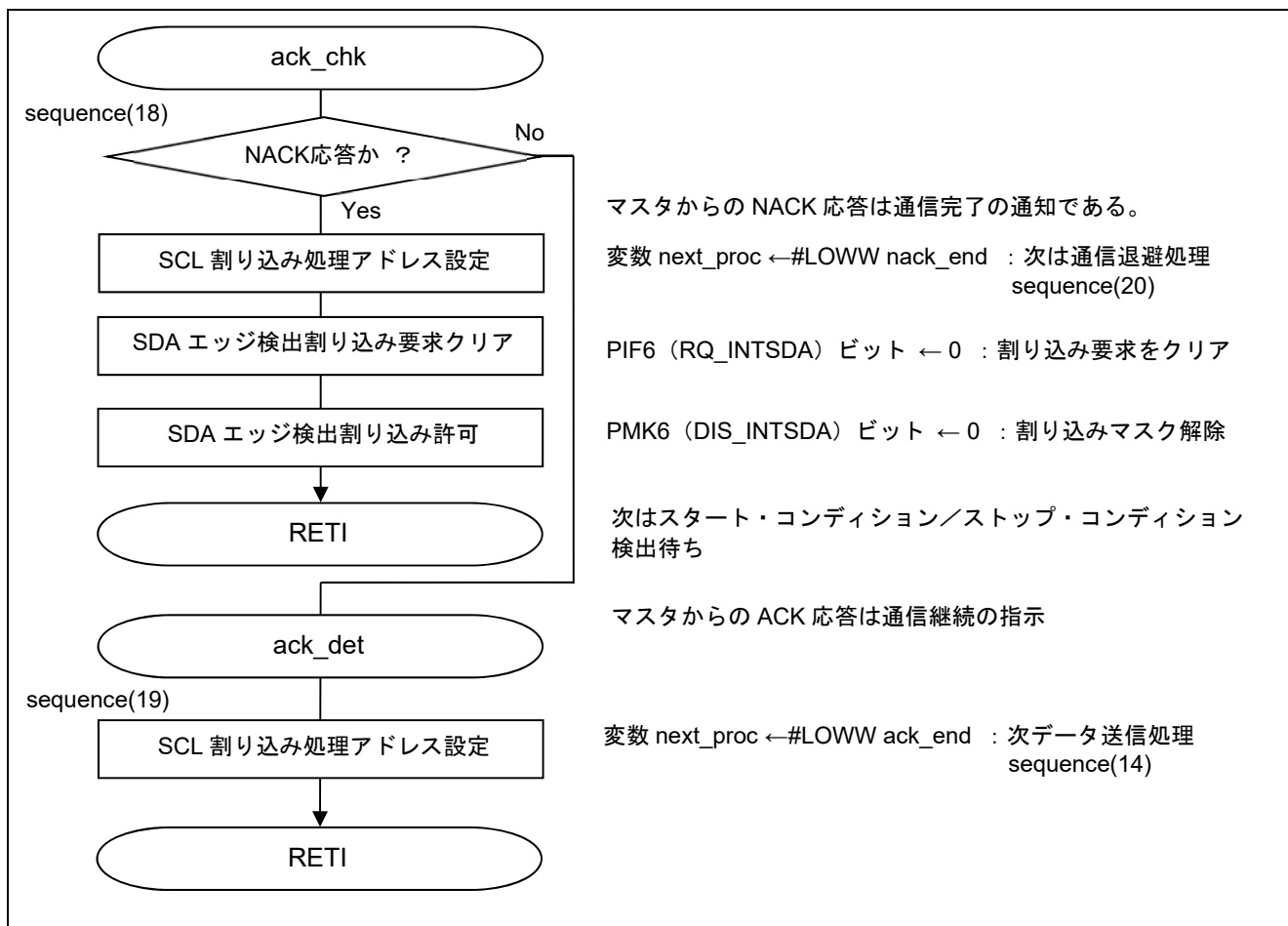


図 5.46 ACK 確認処理

(13) NACK 完了処理 (sequence(20))

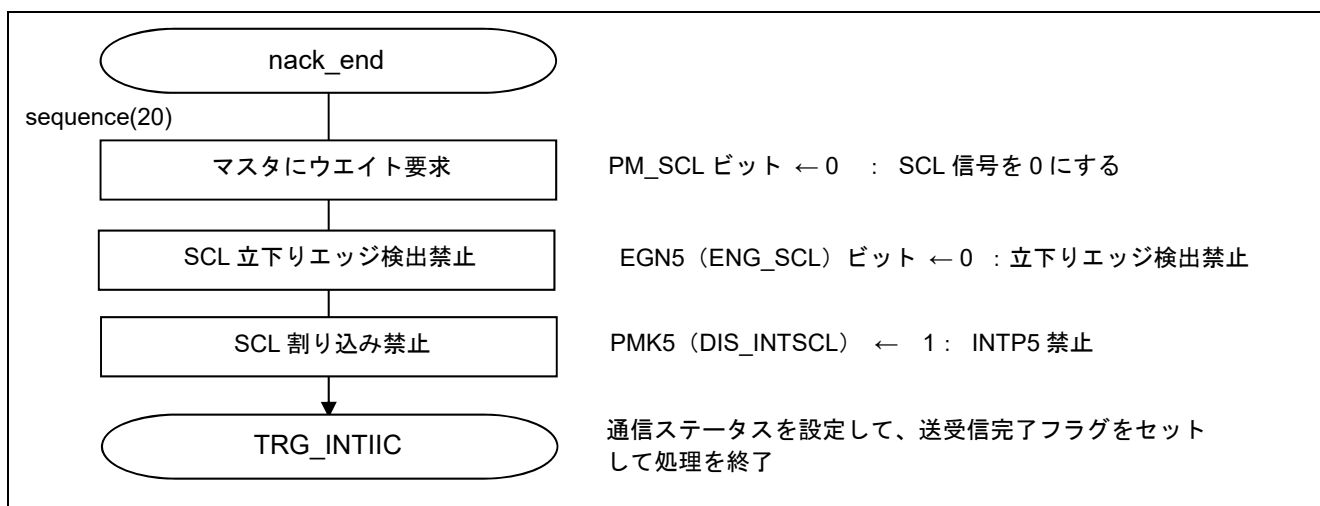


図 5.47 NACK 完了処理

(14) 非選択処理（1/2）（sequence(21)）

自身が非選択時にも、データ処理は行わなくても SCL の数をカウントすることで、どのような通信状態にあるかをモニタする必要があります。

このため、以下の処理では SCL をカウントするだけ（これを読み飛ばしと称する）にしています。。これにより、非選択時に必要な CPU の処理をできるだけ少なくします。

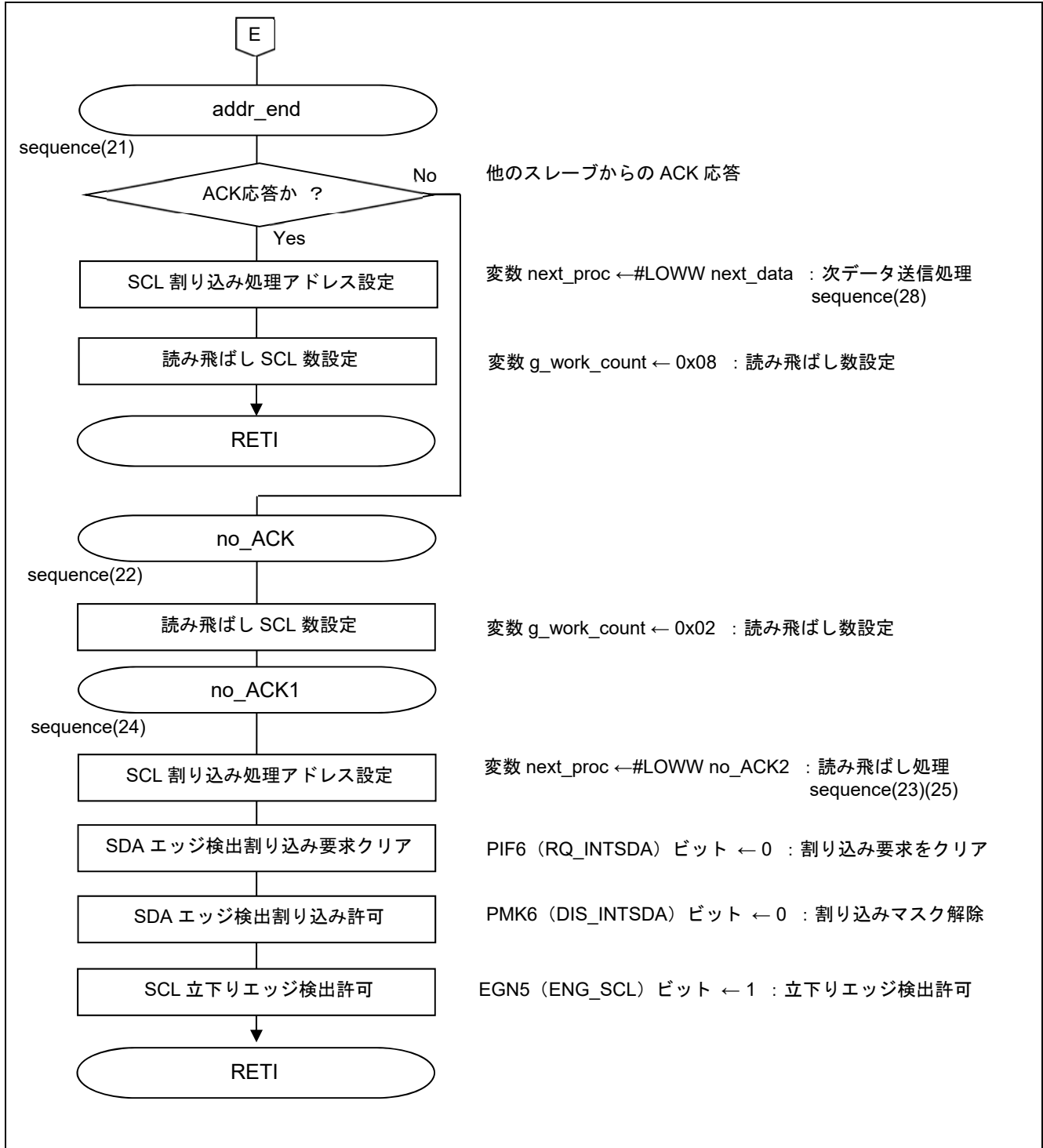


図 5.48 非選択処理（1/2）

(15) 非選択処理 (2/2) (sequence(23)、(25))

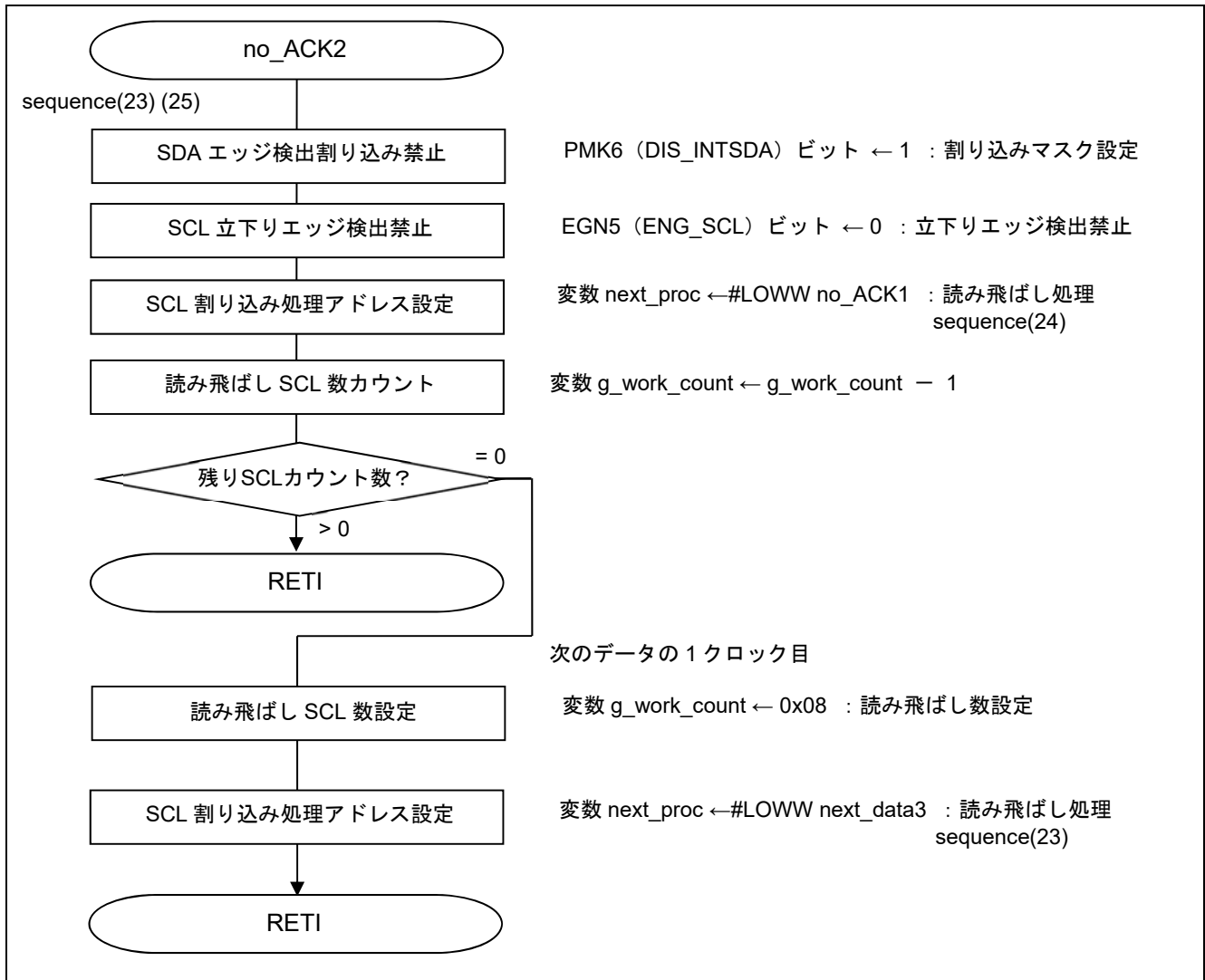


図 5.49 非選択処理 (2/2)

(16) 読み飛ばし処理 2 (sequence(26))

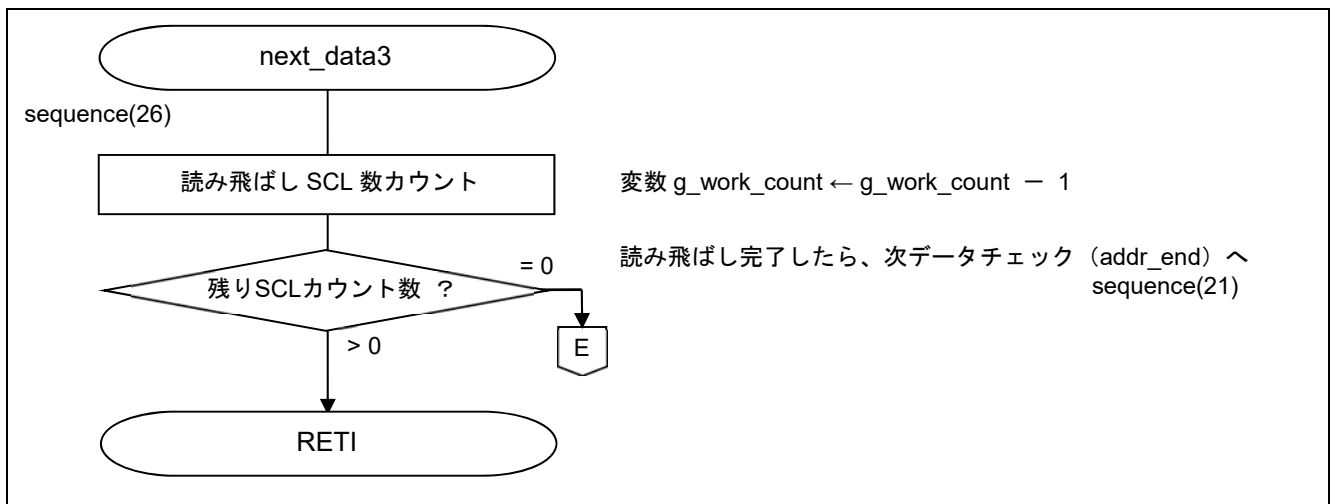


図 5.50 読み飛ばし処理 2

(17) 読み飛ばし処理 3 (sequence(28))

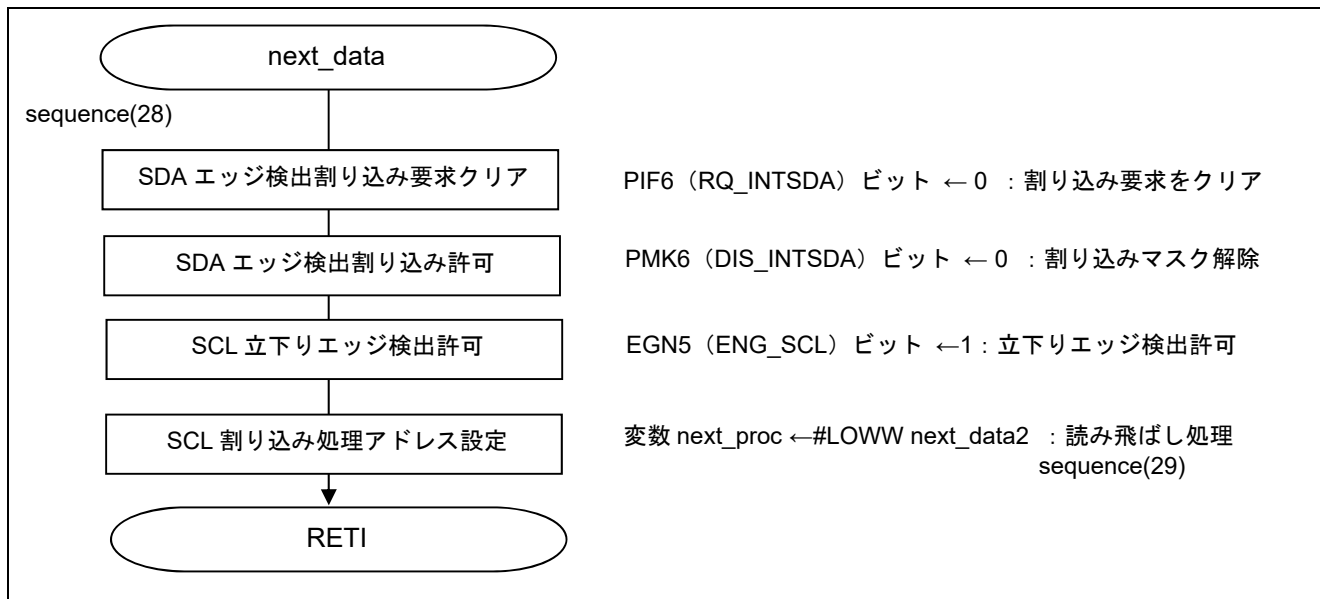


図 5.51 読み飛ばし処理 3

(18) 読み飛ばし処理 4 (sequence(29))



図 5.52 読み飛ばし処理 4

5.7.30 次データ送信開始処理

図 5.53 に次データ送信開始処理のフローチャートを示します。

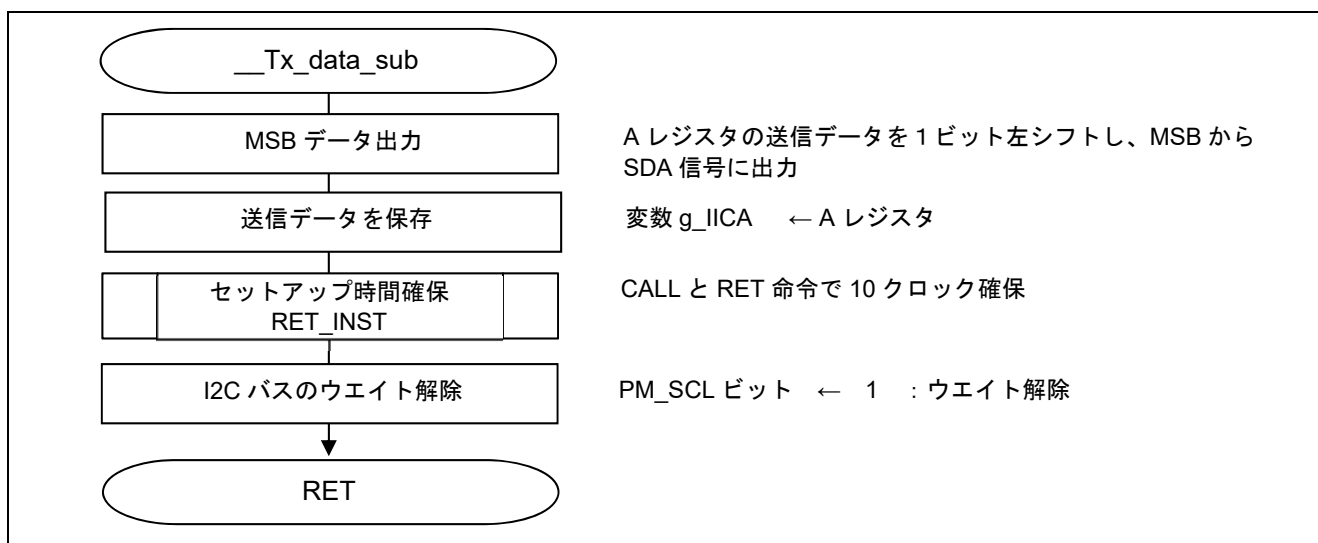


図 5.53 次データ送信開始

5.7.31 次データ受信開始処理

図 5.54 に次データ受信開始処理のフローチャートを示します。

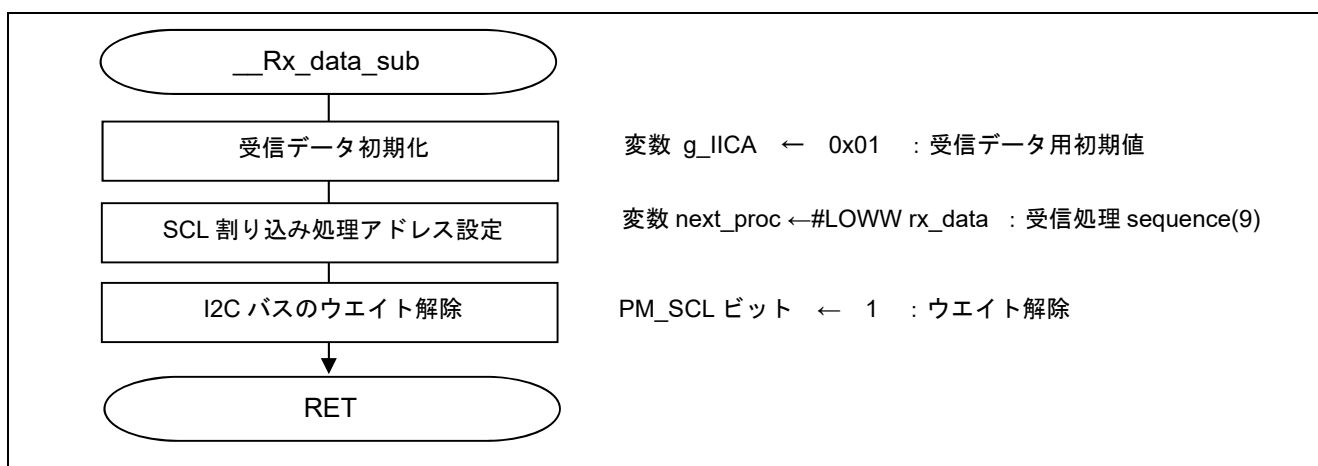


図 5.54 次データ受信開始

5.7.32 データ送信中止処理

図 5.56 にデータ送信中止処理のフローチャートを示します。

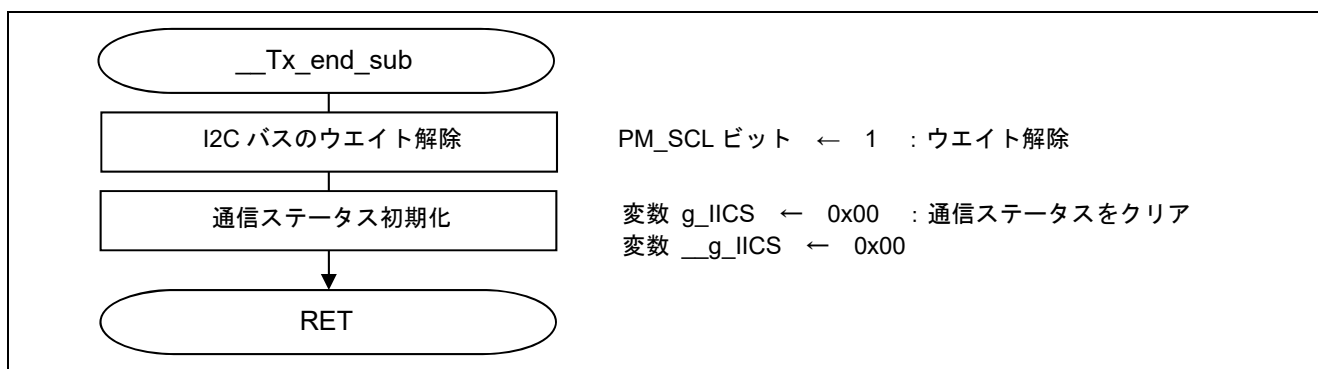


図 5.55 データ送信中止開始

6. I2C バスの基本

I2C バスでは、通信の制御は I2C バスのマスタが行います。スレーブはマスタからの指示に従って、データの送受信を行います。スレーブができることは、マスタの送信データに対して、ACK の応答もしくは NACK の応答とマスタとの同期をとるために SCL 信号をロウ・レベルに引いてマスタにウェイトをかけることだけです。ただし、マスタによっては、ウェイト機能に対応していないこともあるので、注意が必要です。

スレーブは絶対にマスタに従うだけかというだけではなく、通信の細かなプロトコルはマスタの指示に従いますが、その上位の階層では、逆にマスタはスレーブの仕様に合わせる必要があります。これは、スレーブは I2C バスを介して、何らかの機能をマスタに提供しますが、提供する機能はスレーブが規定しているからです。

そのため、最初にスレーブとして、「どのような機能を提供するか」を明確に定義します。それによって、マスタが通信を行うこととなります。

このように、スレーブでは、提供する機能で I2C バスからのアクセスに対する処理が決まってしまうので、マスタのように I2C バスの制御を階層化することは難しいです。ここからは、マスタからの指示に応じて処理が変化します。

6.1 ソフトウェアでの通信の実現

RL78/I1D にはスレーブに対応した I2C 通信機能が搭載されていません。従って I2C バスにスレーブとして接続する為にはポートや外部割り込みを用いたプログラムの処理を用意する必要があります。この場合、通信速度と各コンディションおよび信号に制限が発生します。

本アプリケーションノートにおける SCL 信号の規格を図 6.1 に示します。スタート・コンディションやストップ・コンディションのセットアップ時間やホールド時間も、ここに示す値にする必要があります。

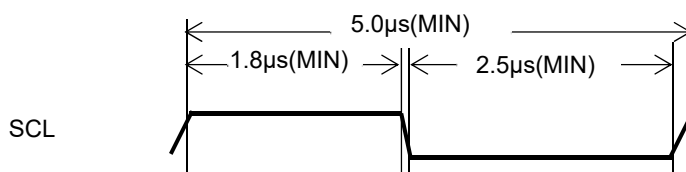


図 6.1 対応する SCL の波形例

また、図 6.2 に示すポートと割り込みを使用します。なお、P32 と P33 には、出力を N-ch O.D.にする機能は搭載されていないので、出力ラッチは 0 を設定しておき、PM で制御することで同じ機能を実現します。

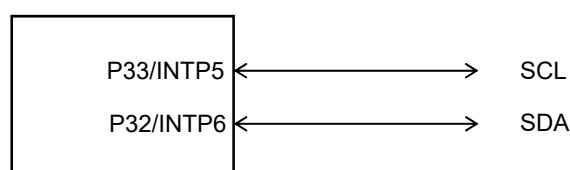


図 6.2 使用する端子

ソフトウェアによる実現では、多くの CPU のパワーを必要とします。特に、自分自身が選択されていないときにも常に I2C バスの状態をモニタしている必要があるために注意が必要です。

I2C バスの状態をモニタし、通信を行うには、信号変化に対する応答時間が重要になります。特に他の割り込みとの関係での処理遅れが大きな影響を及ぼすことから、INTP5 と INTP6 割り込みの優先度は最優先に設定し、他の割り込み処理は、処理の最初で割り込み許可にする必要があります。このため、他の割り込みについては `#pragma interrupt` 宣言時に `enable=true` も追加宣言しています。

さらに、I2C のハードウェアの制御処理部はライブラリとして独立させ、アセンブリ言語で記述していません。利便性を考慮し C 言語で記述したインタフェース部を準備し、C 言語のプログラムからも簡単に使う事ができます。

6.2 スレーブとしての機能

6.2.1 LED 表示機能

LED 表示器としては、8 ビットのデータを表示するために 8 個の LED を使用しています。表示可能なデータは 2 バイトで、SW 入力により 2 バイトのデータのどちらかを指定できます。SW を押していないときは、レジスタ・アドレス 0x00 のデータ、押しているときはアドレス 0x01 のデータを表示します。

表示は上位 4 ビットと下位 4 ビットの 2 回に分けた時分割で表示します。表示の周期は 10ms です。マスタからの表示データはマスタがストップ・コンディションを発行した時点で表示データとして確定します。確定したデータは 50ms 以内に表示可能になります。

6.2.2 A/D 変換機能

4 チャンネルのアナログ入力を変換し、最新の 16 回分の移動平均を得ることが可能です。A/D 変換の仕様は以下の通りです。

- ・アナログ入力 : チャンネル 0~3 の 4 チャンネル
- ・変換方式 : スキャン・モードで連続変換モード
- ・変換分解能 : 12 ビット
- ・変換時間 : 18 μ 秒/チャンネル
- ・バッファ : 16 データ/チャンネル（合計 128 バイト）

変換結果はチャンネル 0、チャンネル 1・・・の順番に読み出すことができます。各チャンネルの 12 ビットの変換結果を上位 4 ビット、下位 8 ビットの順で読み出せます。チャンネル 3 の下位 8 ビットを読み出したあとは、図 6.3 に示すようにチャンネル 0 に戻ります。

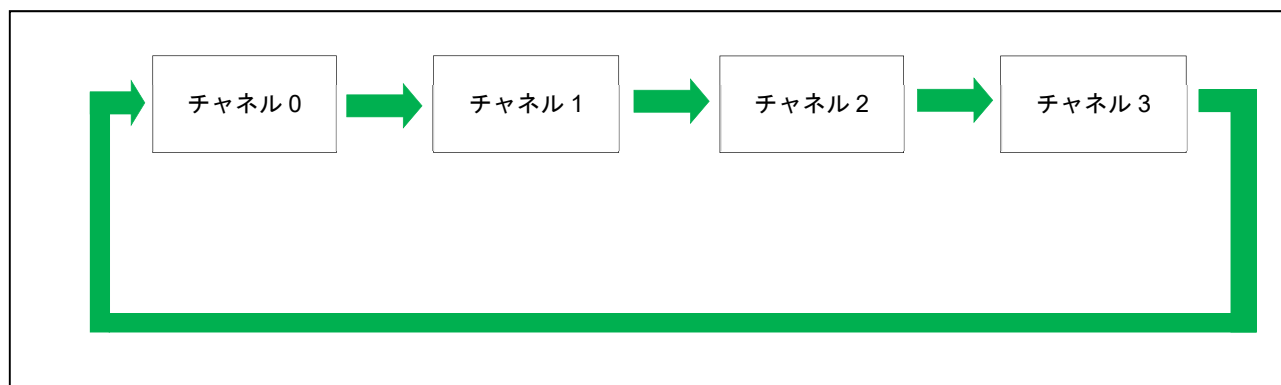


図 6.3 A/D 変換結果の読み出し

6.2.3 RAM 機能

128 バイトの一時保管に使用できる 128 バイトの領域です。初期状態では、0x00~0x7F のデータが格納されています。スレーブ・アドレス 0x70 で指定した場合に RAM へのアクセスとなります。受信したデータは直ちに RAM に書き込まれます。RAM へのアクセスは自動的にアドレスが更新されていきます。アドレス 0x7F へのアクセスの次はアドレス 0x00 へのアクセスとなります。

6.3 ライブラリとのインタフェース仕様

アセンブリ言語記述のライブラリは以下の 3 種のインタフェースを準備しています。

- ・送受信完了フラグ
- ・ストップ・コンディション検出フラグ
- ・通信再開処理関数

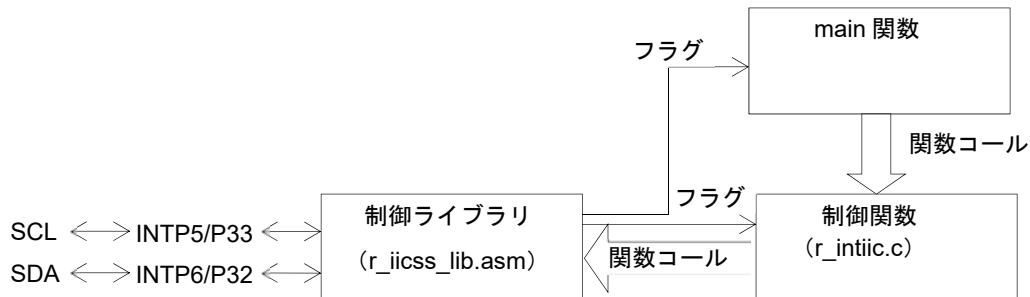


図 6.4 I2C バスの制御構造

6.3.1 I2C 通信フラグ

自局あての 1 バイトのデータの送受信が完了すると、マスタにウエイトをかけて通信を停止し、以下の変数／フラグをセットします。

- ・変数 `_g_IICA` : 受信時の受信データを格納
- ・変数 `_g_IICS` : IICS0 レジスタをまねた通信ステータスを格納
- ・変数 `_g_IIC_IF` : 1 バイトの通信が完了したことを示すフラグ（送受信完了フラグ）

上位のソフトは送受信完了フラグをチェックして、フラグがセットされていたなら、通信ステータス（変数 `_g_IICS`）を参照して対応する処理を行います。

処理が完了したら、ライブラリの関数（6.3.2 次の通信開始関数に示します）を呼び出して、次の通信への準備を行い I2C バスへのウエイトを解除し、次の通信を起動します。

6.3.2 次の通信開始関数

I2C バスでの通信を再開させるために以下の 3 つの関数を準備しています。

- ・`_Rx_data_sub` 関数 : 受信処理で、次のデータ受信を起動
- ・`_Tx_data_sub` 関数 : 送信処理で、引数で渡したデータの送信を起動
- ・`_Tx_end_sub` 関数 : マスタからの NACK 応答への対応でウエイトを解除し、通信から退避

通信状況に応じて、どれかの関数を呼び出します。

6.3.3 ストップ・コンディション検出フラグ

アセンブリ言語記述のライブラリが、ストップ・コンディションを検出したら、変数 `_g_stop_det` に `0x01` を設定します。I2C のステータス（変数 `_g_IICS`）と異なり、上位プログラムでクリアするまでセットされたままです。ストップ・コンディションを検出して処理を開始するような用途で使用します。

本アプリケーションノートの main 処理では、受信した LED 点灯データを点灯処理プログラムに送り、実際の LED 点灯処理を行うためのトリガとして使用しています。

6.4 スレーブ・アドレスの指定

6.4.1 スレーブ・アドレス・テーブル

このライブラリでは、使用するスレーブ・アドレスの情報をテーブルとしてもっています。マスタから受信したスレーブ・アドレスの上位 7 ビットでテーブルを参照します。

得られた値の上位 4 ビットがアドレス ID となります（下位 4 ビットは 0 です）。得られた値が 0x00 の場合には受信したアドレスは自局のアドレスではないので、通信には参加しません。

アドレス ID が 1~F の 15 種類は自局が選択されたことを示します。つまり、15 の独立したスレーブ・アドレスが使用可能です。

なお、複数のスレーブ・アドレスに同じアドレス ID を割り当てることも可能ですが、同じアドレス ID を使用した場合には、基本的に同じ処理となります。

得られたアドレス ID は通信ステータス（変数 `_g_iICS`）の上位 4 ビットにセットされます。通信ステータスの構成を図 6.5 に示します。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ID3	ID2	ID1	ID0	F_TRC	F_ACKD	F_STD	F_SPD

図 6.5 通信ステータスの構成

ACK 応答の初期値とアドレス・テーブルは定数ファイルとして、`r_iicss_adr.asm` に格納されています。図 6.6 にスレーブ・アドレス・テーブルの例を示します。この例では 0x30（アドレスは 0x60）、0x38（アドレスは 0x70）、0x40（アドレスは 0x80）、0x48（アドレスは 0x90）に 0x10（アドレス ID は 1）、0x20（アドレス ID は 2）、0x30（アドレス ID は 3）、0x40（アドレス ID は 4）が設定されています。

```

SADR_TBL:
:0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x00-0x07
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x08-0x0F
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x10-0x17
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x18-0x1F
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x20-0x27
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x28-0x2F
.DB 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x30-0x37
.DB 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x38-0x3F
.DB 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x40-0x47
.DB 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x48-0x4F
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x50-0x57
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x58-0x5F
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x60-0x67
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x68-0x6F
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x70-0x77
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x78-0x7F

```

図 6.6 スレーブ・アドレス・テーブル

6.4.2 ACK 応答フラグ

各アドレス ID には、ACK 応答を制御するフラグが準備されています。ACK 応答を行う場合には 0x0F が ACK 応答を行わない場合には 0x00 がセットされます。

図 6.7 に ACK 応答テーブルの構成を示します。この例では、アドレス ID が 1~4 は ACK 応答するようになっていきます。

```

ACK_TBL:
;
:0/8 1/9 2/A 3/B 4/C 5/D 6/E 7/F
.DB 0x00, 0x0F, 0x0F, 0x0F, 0x0F, 0x00, 0x00, 0x00 ; 0x00-0x07
.DB 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 ; 0x08-0x0F

```

図 6.7 ACK 応答テーブルの構成

6.5 スレーブをアクセスするためのプロトコル

6.5.1 LED への表示

LED に表示する場合のアクセス方法を図 6.8 に示します。

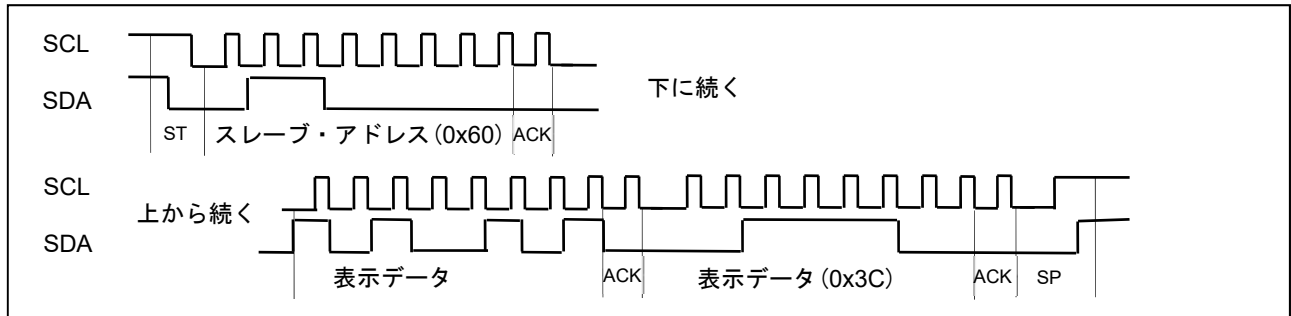


図 6.8 LED 表示データ書き込みタイミング

この例では、スタート・コンディション (ST)、スレーブ・アドレス (0x60) に続けて、0xA5、0x01 に 0x3C を書き込んでいます。最後に、ストップ・コンディション (SP) でスレーブに送信完了を通知しています。

これに対して、スレーブは ACK 応答を行っています。

6.5.2 A/D 変換結果の読み出し

A/D 変換結果を読み出す場合のアクセス方法を図 6.9 に示します。

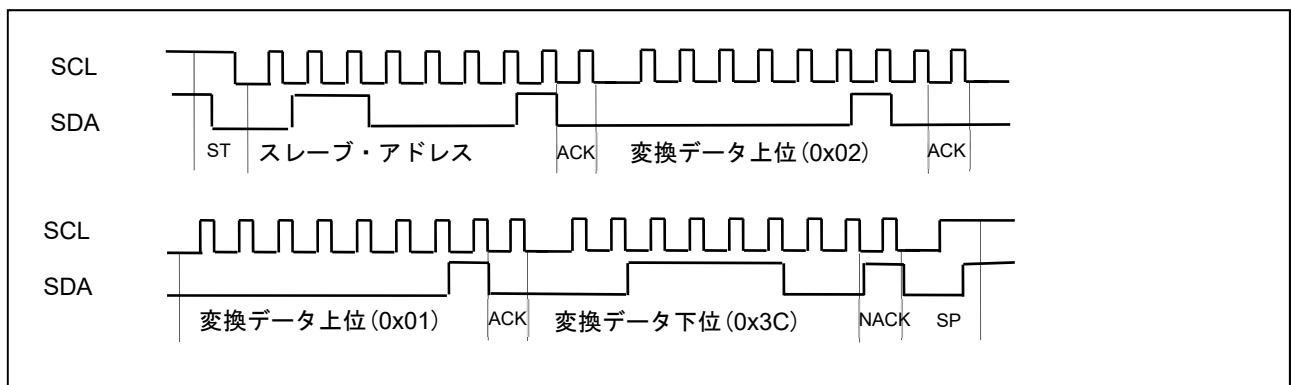


図 6.9 A/D 変換結果読み出しタイミング

この例では、最初にスタート・コンディション (ST)、スレーブ・アドレス (0x61) で選択することで、チャンネル 0 の A/D 変換結果を上位、下位の順に読み出します。この図では、チャンネル 0 の変換結果の上位が 0x02 となっています。

一番下のタイミングでは変換結果 (0x013C) を読み出したところで、マスタが NACK 応答を戻してきたので、スレーブは通信完了として通信から退避します。最後に、ストップ・コンディション (SP) でマスタは I2C バスを開放して通信を完了します。

4 チャンネル分の A/D 変換が完了したら、得られた移動平均値を IIC の変換結果送信用バッファに設定します。一方、A/D 変換結果の読み出しはチャンネル 3 の下位の読み出し後はチャンネル 0 の上位になるので、A/D 変換結果の読み出しを繰り返すことで、最新の変換結果を得ることが可能です。

6.5.3 RAM のデータ読み出し

RAM のデータを読み出す場合のアクセス方法を図 6.10 に示します。

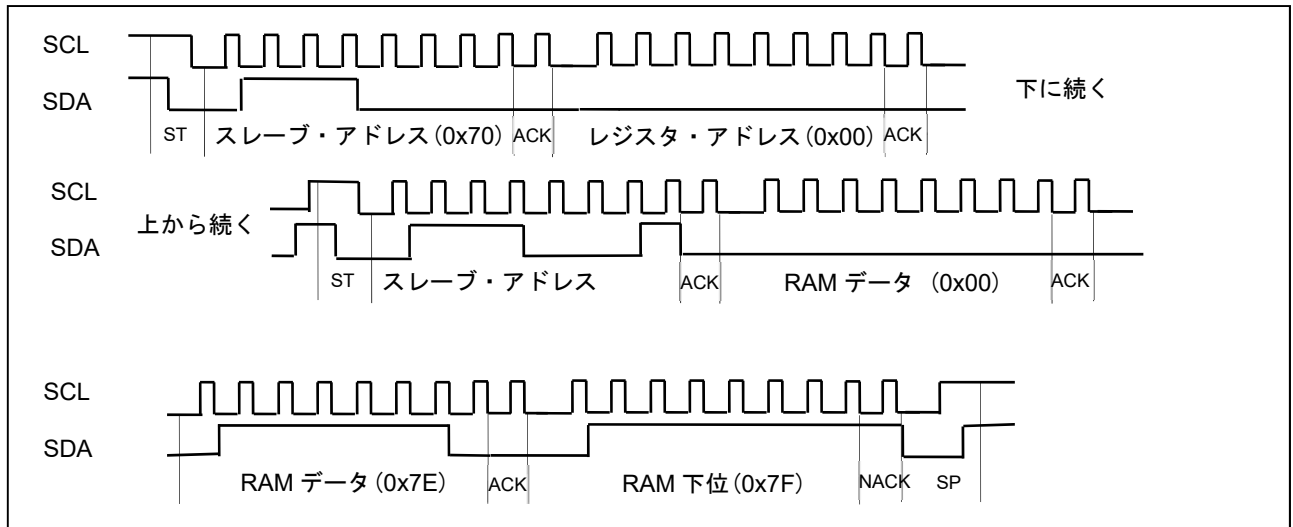


図 6.10 RAM データの読み出しタイミング

この例では、最初にスタート・コンディション（ST）、スレーブ・アドレス（0x70）に続けて、レジスタ・アドレス 0x00（=RAM のアドレス 0x00）を指定しています。

その後、リスタートして、読み出しでスレーブを選択（0x71）することで、指定したアドレス 0x00 番地の RAM の値を読み出します。この図では、アドレス 0x00 の値は 0x00 になっています

一番下のタイミングでは RAM アドレス 0x7E の値を読み出し、0x7F の値を読み出したところで、マスターが NACK 応答を戻してきたので、スレーブは通信完了として通信から退避します。最後に、ストップ・コンディション（SP）で I2C バスを開放して通信を完了します。

6.5.4 RAM へのデータ書き込み

RAM にデータを書き込む場合のアクセス方法を図 6.11 に示します。

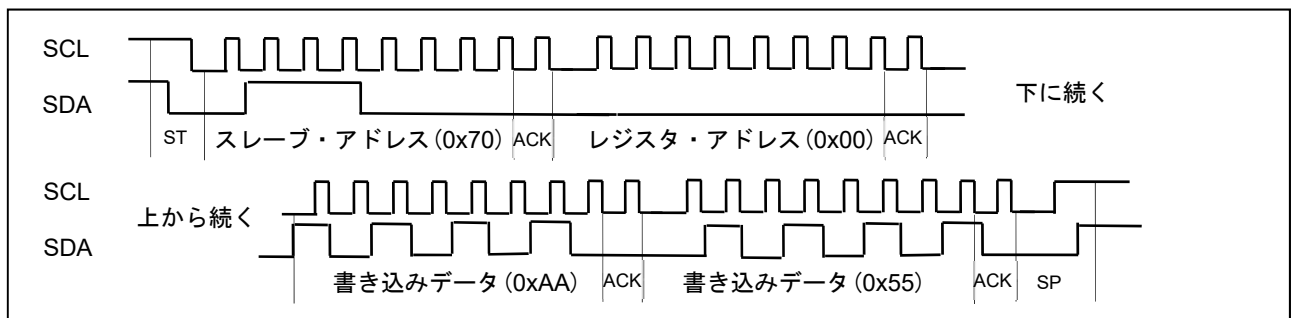


図 6.11 RAM へのデータ書き込みタイミング

この例では、最初にスタート・コンディション（ST）、スレーブ・アドレス（0x70）に続けて、レジスタ・アドレス 0x00 を指定しています。

その後、0x00 番地への書き込みデータとして 0xAA を次の番地への書き込みデータとして 0x55 を送信しています。2 バイトのデータを送信して通信を完了し、ストップ・コンディションを発行して I2C バスを開放しています。

7. ソフトウェアによる I2C バスの基本的な制御

ソフトウェアで I2C バスのスレーブ機能を実現するためには、SCL 信号と SDA 信号の立ち上がり／立下りエッジを検出する必要があります。

このために INTP5（SCL 信号のエッジ検出）と INTP6（SDA 信号のエッジ検出）を使用します。

7.1 エッジ検出割り込み

限られた時間の中で検出されたエッジの処理を行うために最優先の割り込みを使用し、他の割り込みに対して多重割り込みで処理します。それでも、ハードウェアでの割り込み処理開始までに 9~16 クロックかかりますし、優先度の低い割り込みを受け付けた直後に割り込み要求が発生すると、さらに 9 クロック必要になります。このことを考慮して、できるだけ高速の処理を行います。

7.1.1 SCL エッジ検出

基本的に、SCL のエッジはデータの送受信に使用します。そのため、立ち上がりエッジを検出するか立下りエッジを検出するかを細かに切り替えながら処理をすすめることで、全体としての処理時間を少なくしています。図 7.1 にアドレス受信での SCL 信号と SDA 信号の有効エッジを示します。

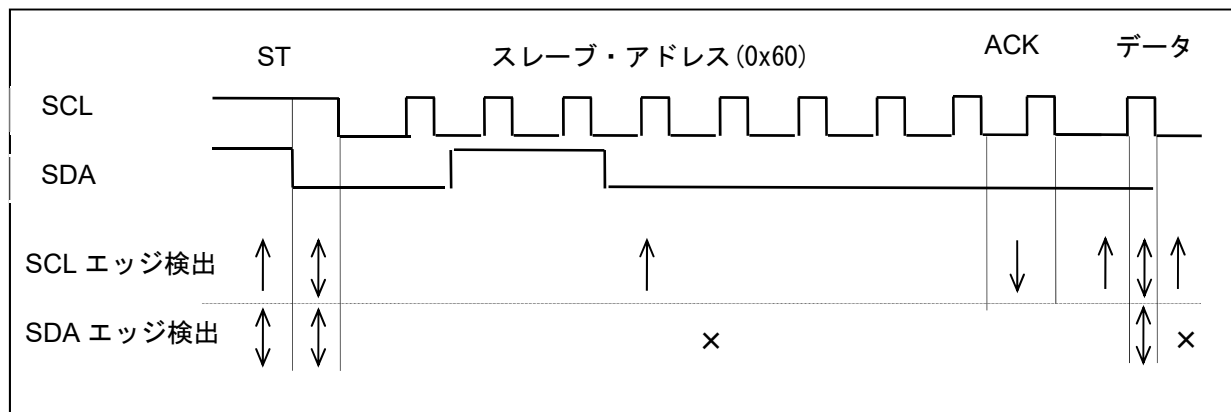


図 7.1 タイミングによる SCL と SDA の使用エッジ

スタート・コンディション検出までは SDA の両エッジ検出が有効になっています。SDA 信号の立下りでストップ・コンディションを検出した後も SCL 信号の立下りを検出して、スレーブ・アドレス受信がスタートするまでは全てのエッジ検出を有効にしています。これにより、スタート・コンディションに続けてストップ・コンディションが発行されるケースに対応しています。

スレーブ・アドレス受信を開始すると、SDA 信号のエッジ検出割り込みは禁止します。8 クロック目の SCL 信号の立ち上がりでスレーブ・アドレスの受信が完了したら、アドレスを比較します。アドレスが一致したら、プログラムで SCL 信号の立下りを待ち、ACK 応答を開始し、SCL 信号を立下りエッジ検出に変更します。（処理が多いので、1 周期の中で 2 回の割り込み受け付けは無駄が多いので、SCL 信号の立下りはプログラムで待っています。）

その後、SCL 信号の立下りエッジで ACK 応答を終了して、スレーブ・アドレス受信を完了します。次に備えて、SCL 信号の立ち上がりエッジ検出を有効にします。

次の SCL 信号が立ち上がったハイの期間（1 クロック目）では、スタート・コンディションやストップ・コンディションが発行されることがあるので、すべてのエッジ検出を有効にします。

7.1.2 SDA エッジ検出

通常の通信動作は SCL のエッジ検出でのシーケンシャルな動作です。SDA 信号のエッジ検出は、シーケンス動作の打ち切りに使用します。このため、受け付け可能にするタイミングが重要になります。図 7.1 に示すように、特定のタイミングの SCL 信号がハイの期間でのみ許可します。

7.2 制御処理

処理時間を短縮するために、割り込み処理部はアセンブリ言語で記述しています。さらに、INTP5 割り込みの処理については、応答時間を短くするために、あらかじめ次の処理内容に応じた処理アドレスを変数 `next_proc` に設定しています。

これは、SCL 信号のエッジによる処理はシーケンシャルになっており、次の処理が限られているからです。それに対して、SDA 信号のエッジによる処理は割り込み処理のようにシーケンシャルな処理を中断するような機能（SP：ストップ・コンディション、ST：スタート・コンディション）になっています。

さらに、INTP5 割り込みの最初に、SCL 信号と SDA 信号を入力するポートの状態を変数 `g_P_image` に取り込んでいます。

このような処理を意識しなくても使えるように、上位ソフトとのインタフェースを限定しています。

7.2.1 SCL エッジ検出割り込みによるシーケンス①

通信が始まる前の SCL 信号と SDA 信号が共にハイの状態をシーケンス 1 としてシーケンスを定義しています。表 7.1 にアドレス受信および非選択時のシーケンスを示します。

表 7.1 アドレス受信および非選択時の処理

シーケンス番号	状態／処理
シーケンス1	初期状態（SCL信号とSDA信号はハイ状態）
シーケンス2	ストップ・コンディション検出（次はスタート・コンディション待ち）
シーケンス3	スタート・コンディション検出（SCL立下り待ち）
シーケンス4	スタート・コンディション検出後のSCL立下り（アドレス受信待ち）
シーケンス5	SCLの立ち上がりエッジ（スレーブ・アドレスの取り込み）
シーケンス6	8回目のSCLの立ち上がり（スレーブ・アドレスが一致）
シーケンス6'	8回目のSCLの立ち上がり（スレーブ・アドレスが不一致）
シーケンス7	8クロック目のSCLの立下り（ACK応答開始）
シーケンス8	9回目のSCLの立下り（ACK応答完了）
シーケンス21	9回目のSCLの立ち上がり（スレーブ・アドレスが不一致）
シーケンス22	9回目のSCLの立ち上がりでACK応答なし（ST、SP待ち）
シーケンス23、25	9回目または1回目のSCLの立下り（ST、SP待ちを中止、SCLを読み飛ばし）
シーケンス24	9回目のSCLの立ち上がり（ST、SP待ち）
シーケンス26	SCLの立ち上がりカウント（9回目のSCL立ち上がり待ち）

7.2.2 SCL エッジ検出割り込みによるシーケンス②

表 7.2 に受信処理のシーケンスを示します。受信処理では、8 ビットのデータが揃うと、SCL バスにウェイトをかけて、上位処理に通知（変数 `g_IICA` に受信データ、変数 `g_IICS` に通信ステータスをセットし、`g_IIC_IF` を `0x01` にセット）します。

また、SCL の 1 クロック目は、リスタート／ストップ及び次のデータ受信開始の可能性があるので、すべての要因を許可しておきます。リスタート／ストップが検出されれば、受信処理のシーケンスはキャンセルされます。SCL が立ち下がれば、それは受信動作の継続なので、SDA のエッジ検出を禁止して処理を継続します。

表 7.2 受信処理シーケンス

シーケンス番号	状態／処理	備考
シーケンス8'	9回目のSCLの立下り（受信完了で上位ソフトに情報伝達）	次は <code>Rx_data_sub</code>
<code>Rx_data_sub</code>	I2Cバスのウェイトを解除	次はシーケンス9
シーケンス9	1回目のSCLの立ち上がり（受信データ取り込み）	次はシーケンス11
シーケンス6	8回目のSCLの立ち上がり（ACK応答準備）	
シーケンス7	8クロック目のSCLの立下り（ACK応答開始）	次はシーケンス8→8'
シーケンス11	SCLの立ち上がりでデータ取り込み（ST、SP、SCL立下り待ち）	次はシーケンス6
シーケンス10	1回目のSCLの立下り（ST、SP待ちを中止）	次はシーケンス11

上位処理は変数 `g_IICA` の受信データを処理したら、関数 `Rx_data_sub` を呼び出すことで、I2C バスへのウェイトを解除して次の受信を再開させます。

7.2.3 SCL エッジ検出割り込みによるシーケンス③

表 7.3 に送信処理のシーケンスを示します。送信処理では、スレーブ・アドレス一致での ACK 応答後に SCL バスにウェイトをかけて、上位処理に通知（変数 `_g_iicss` に通信ステータスをセットし、`_g_iic_if` を `0x01` にセット）します。

表 7.3 送信処理シーケンス

シーケンス番号	状態／処理	備考
シーケンス14	送信での選択（アドレス受信完了で上位ソフトに情報伝達）	次は <code>_Tx_data_sub</code>
<code>_Tx_data_sub</code>	1ビット目を出力し、I2Cバスのウェイト解除	次はシーケンス15
シーケンス15	1回目のSCLの立ち上がり、ACKD、STD、SPDをクリア	次はシーケンス16
シーケンス16	SCLの立下り（データ送信タイミング、7回繰り返す）	
シーケンス17	8クロック目のSCLの立下り（ACK受信開始）	次はシーケンス18
シーケンス18	9クロック目のSCLの立ち上がりでACK受信	
シーケンス19	9クロック目のSCLの立ち上がりでACK検出	次はシーケンス14
シーケンス20	9クロック目のSCLの立下り（上位ソフトにNACKを通知）	次は <code>_Tx_end_sub</code>
<code>_Tx_end_sub</code>	通信ステータスをクリアし、I2Cバスを開放	

上位処理は引数（A レジスタ）に送信データをセットしたら、関数 `_Tx_data_sub` を呼び出すことで、次のデータを SDA に出力し、I2C バスへのウェイトを解除して次のデータの送信を再開させます。

スレーブ送信の場合には、9 クロック目以外ではスレーブが SDA 信号をドライブしているので、マスタが何か行うにはスレーブに NACK 応答を戻して、スレーブに送信を中止させる必要があります。NACK を受信したら、NACK 応答を上位ソフトに通知し、上位ソフトからの関数 `_Tx_end_sub` の呼び出しで I2C バスのウェイトを解除し、通信から退避（SCL のエッジ検出を禁止）します。

7.3 I2C スレーブのファイル構成

このソフトウェアによる I2C スレーブ機能のライブラリは3つのファイルから構成されます。

- ・ `r_iicss_lib.asm` : I2C の制御を行っているプログラムの本体（変更しないことを推奨）
- ・ `r_iicss_adr.asm` : I2C バスでのスレーブ・アドレス等の定義。必要に応じて変更してください。
- ・ `r_intiic.c` : IICA0 での INTIICA0 処理部に該当する部分です。I2C バスを使った処理を記述してください。このサンプルプログラムでは、6.2 スレーブとしての機能に示す機能を実装しています。他の使い方をする場合には、ここの処理を変更してください。

8. コード生成での設定内容

プロパティの「ファイル生成モード」の「API 関数の出力制御」を「初期化関数のみ出力する」に設定し、以下の設定を行います。

(1) クロック発生回路の設定

(a) 端子割り当て設定：そのまま確定

(b) クロック設定

- ・動作モード設定：高速メイン・モード $2.7(V) \leq VDD \leq 3.6(V)$
- ・メイン・システム・クロック (fMAIN) 設定：高速オンチップオシレータクロック (filH)
- ・高速オンチップオシレータクロック設定：24 (MHz)
- ・中速オンチップオシレータクロック設定：動作をチェックしない
- ・高速システム・クロック設定：動作をチェックしない
- ・サブシステム・クロック (fSUB) 設定：動作をチェックしない
- ・低速内蔵発振クロック (fil) 設定：周波数 15 (kHz)
- ・RTC、FMC、インターバル・タイマ、PCLBUZ 動作クロック設定：fil
- ・CPU と周辺クロック設定：24000 (filH) (kHz)

(c) オンチップ・デバッグ設定

- ・オンチップ・デバッグ動作設定：使用する
- ・RRM 機能設定：使用しない
- ・セキュリティ ID 設定：セキュリティ ID を設定する
- ・セキュリティ ID 認証失敗時の設定：フラッシュ・メモリのデータを消去する

(d) リセット要因確認

- ・リセット要因を確認する関数を出力する：チェックを外す

(e) 安全機能：全て「使用しない」を選択

(f) データ・フラッシュ：データ・フラッシュのアクセス禁止

(2) ポートの設定

- ・P0.0～P0.3 は出力（データは 0）
- ・P6.0 と P6.1 は 1 を出力
- 他は全て初期値（使用しない）のまま

(3) タイマの設定

(a) 一般設定 チャンネル 3：インターバル・タイマ

(b) チャンネル 3

- ・動作モード設定：16 ビット
- ・インターバル時間（16 ビット）設定：5ms
- これ以外は初期状態のまま

(4) 周波数測定回路の設定

初期値（使用しない）のまま

(5) 12 ビット・インターバル・タイマの設定

初期値（使用しない）のまま

(6) 8 ビット・インターバル・タイマの設定

初期値（使用しない）のまま

(7) クロック出力／ブザー出力制御の設定

全て初期値（使用しない）のまま

(8) ウォッチドッグ・タイマの設定

- ・ HALT/STOP/SNOOZE モード時の動作設定：停止
- ・ ウォッチドッグ・タイマ動作設定：使用しない

(9) A/D コンバータの設定

- ・ A/D コンバータ動作設定：使用する
- ・ コンパレータ動作設定：許可
- ・ 分解能設定：10 ビット
- ・ VREF(+)設定：AVDD
- ・ VREF(-)設定：AVSS
- ・ トリガ・モード設定：ソフトウェア・トリガ・モード
- ・ 動作モード設定：連続スキャン・モード
- ・ ANI0 – ANI3 アナログ入力端子設定：ANI0 – ANI3
- ・ ANI16 – ANI18 アナログ入力端子設定：全てチェックを外す
- ・ 変換開始チャネル設定：ANI0 – ANI3
- ・ 基準電圧： $2.7V \leq AVDD \leq 3.6V$
- ・ 変換時間モード：標準 1
- ・ 変換時間：9 (216/fCLK) (μs)
- ・ 変換結果上限/下限設定：ADLL \leq ADCRH \leq ADUL で割り込み要求信号 (INTAD) 発生
- ・ 割り込み設定：A/D の割り込み許可をチェック（優先順位はレベル 1）

(10) コンパレータの設定

初期値（使用しない）のまま

(11) オペアンプの設定

初期値（使用しない）のまま

(12) シリアル・アレイ・ユニットの設定

初期値（使用しない）のまま

(13) データ演算回路の設定

初期値（使用しない）のまま

(14) データ・トランスファ・コントローラの設定

初期値（チェックなし）のまま

(15) イベントリンクコントローラの設定

初期値（チェックなし）のまま

(16) 割り込みの設定

- ・ INTP5 設定：立上りエッジ、優先順位：高
- ・ INTP6 設定：両エッジ、優先順位：高
- その他は初期値（チェックなし）のまま

(17) キー割り込みの設定

初期値（チェックなし）のまま

(18) 電圧検出回路の設定

- ・ 電圧検出動作設定：使用する
- ・ 動作モード設定：リセットモード
- ・ 検出電圧設定：2.75 (V)

9. 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。

10. 参考ドキュメント

RL78/I1D ユーザーズマニュアル ハードウェア編 Rev.2.10 (R01UH0474J)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 Rev.2.00 (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/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。)

ホームページとサポート窓口<website and support,ws>

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/>

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	2016.11.15	－	初版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子

（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違くと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 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に関し、当社は、何らの責任を負うものではありません。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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様が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。
9. 本資料に記載されている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。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍用用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

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とその総株主の議決権の過半数を直接または間接に保有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。



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>