
RL78/ G11

R01AN4063JJ0100

デジタル電圧計

Rev. 1.00

2018.1.19

要旨

本アプリケーションノートでは、RL78/G11 に搭載された A/D コンバータを利用して電圧計を実現します。

動作確認デバイス

RL78/G11

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.....	4
1.1	電圧測定方法.....	4
1.2	自動測定レンジへの対応.....	5
1.3	抵抗値と測定レンジの検討.....	6
1.4	測定結果の演算処理.....	7
1.5	内部基準電圧による補正処理.....	8
1.6	測定レンジの切り替え処理.....	8
1.7	AC 電源への対応.....	9
1.8	表示周期.....	9
2.	動作確認条件.....	10
3.	関連アプリケーションノート.....	10
4.	ハードウェア説明.....	11
4.1	ハードウェア構成例.....	11
4.2	使用端子一覧.....	12
5.	ソフトウェア説明.....	12
5.1	動作概要.....	12
5.2	オプション・バイトに反映される設定一覧.....	13
5.3	定数一覧.....	14
5.4	変数一覧.....	15
5.5	関数一覧.....	16
5.6	関数仕様.....	17
5.7	フローチャート.....	23
5.7.1	初期設定関数.....	23
5.7.2	システム関数.....	24
5.7.3	入出力ポートの設定.....	25
5.7.4	CPU クロック設定.....	26
5.7.5	タイマ・アレイ・ユニットの設定.....	27
5.7.6	8 ビット・インターバル・タイマの設定.....	28
5.7.7	IICA の設定.....	29
5.7.8	A/D コンバータの設定.....	30
5.7.9	DTC の設定.....	31
5.7.10	INTP0 の設定.....	31
5.7.11	メイン処理.....	32
5.7.12	R_MAIN_UserInit 処理.....	35
5.7.13	LCD の初期設定.....	36
5.7.14	LCD コマンド設定処理.....	37
5.7.15	時間待ち処理.....	38
5.7.16	60us 待ち処理.....	38
5.7.17	LCD 文字列表示処理.....	39
5.7.18	表示データ設定処理.....	40
5.7.19	I2C 送信起動処理.....	40
5.7.20	I2C 通信完了待ち処理.....	41
5.7.21	ストップ・コンディション発行処理.....	41
5.7.22	I2C バス状態チェック処理.....	42
5.7.23	INTP0 割り込み処理.....	43
5.7.24	INTIT00 割り込み処理.....	43
5.7.25	内部基準電圧確認処理.....	44
5.7.26	1us 待ち処理.....	45
5.7.27	ADC と DTC 設定処理.....	46
5.7.28	TM01 起動処理.....	46
5.7.29	A/D 変換完了割り込み処理.....	47
5.7.30	I2C 通信完了割り込み処理.....	49

5.7.31	16 ビット乗算処理.....	51
5.7.32	32 ビット除算処理.....	51
5.7.33	16 ビット除算処理.....	51
5.7.34	16 ビット剰余処理.....	52
5.7.35	16 ビット自乗平均平方根処理	53
6.	サンプルコード	54
7.	参考ドキュメント	54

1. 仕様

1.1 電圧測定方法

本アプリケーションノートでは、RL78/G11 に搭載している A/D コンバータを使用して、アナログ入力電圧を測定します。測定する電圧範囲は、 $0 \sim \pm 1\text{V}$ ($\pm 1\text{V}$ レンジ)、 $0 \sim \pm 10\text{V}$ ($\pm 10\text{V}$ レンジ)、 $0 \sim \pm 100\text{V}$ ($\pm 100\text{V}$ レンジ) の 3 種類です。

最初に、アナログ電圧入力回路 (図 1.1) を考えます。ANIO 端子は抵抗 R_0 を介して GND に接続されます。ANIO 端子と測定電圧の間には切り替えスイッチが設置され、 $0\ \Omega$ 、 R_1 、 R_2 の何れかで接続されます。

$0\ \Omega$ で接続したときは $\pm 1\text{V}$ レンジです。 R_0 の 9 倍の抵抗値である R_1 で接続したときは $\pm 10\text{V}$ レンジ、 R_0 の 99 倍の抵抗値である R_2 で接続したときは $\pm 100\text{V}$ レンジになります。

抵抗分割によって、ANIO 端子に印加する電圧を 1V 以下に抑えます。なお、RL78/G11 では、端子に VDD 以上の電圧を入力することはできません。

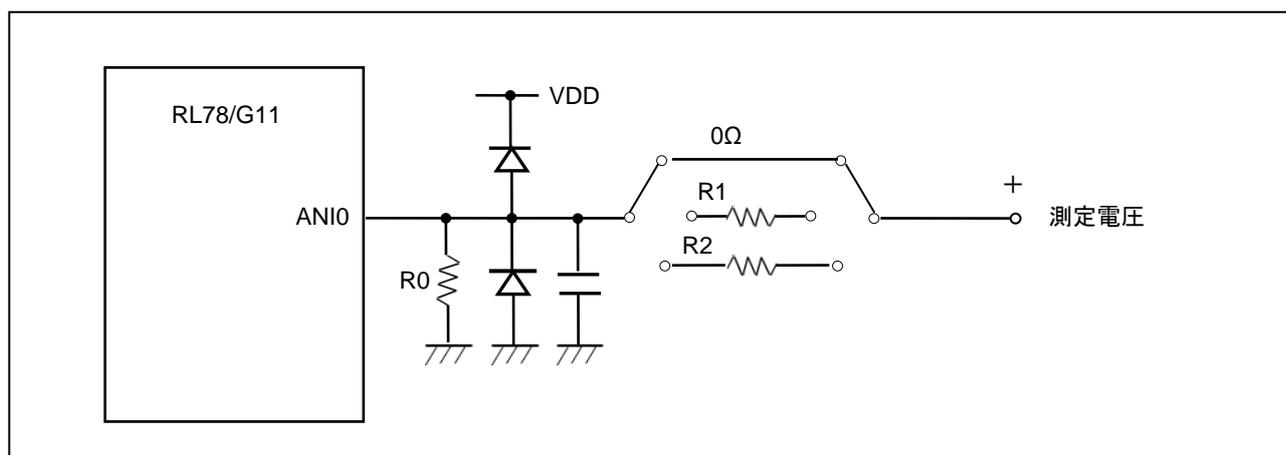


図 1.1 アナログ電圧入力回路

次に、測定電圧として、正電圧だけでなく負電圧も許可する場合を考えます。図 1.2 のように、2 つのポートを追加して、その間を 2 つの抵抗で結び、その中点をアナログ入力端子と一側の測定電圧に接続します。これらポートをハイ・レベル出力とロウ・レベル出力にして、一側の測定電圧を中間電位 (RL78/G11 基準で $1/2\text{VDD}$ 程度) にします。これにより、ANIO 端子に印加される電圧は常に正電圧となります。+側および一側の測定電圧を測定し、その差を求めることで測定対象の電圧が得られます。

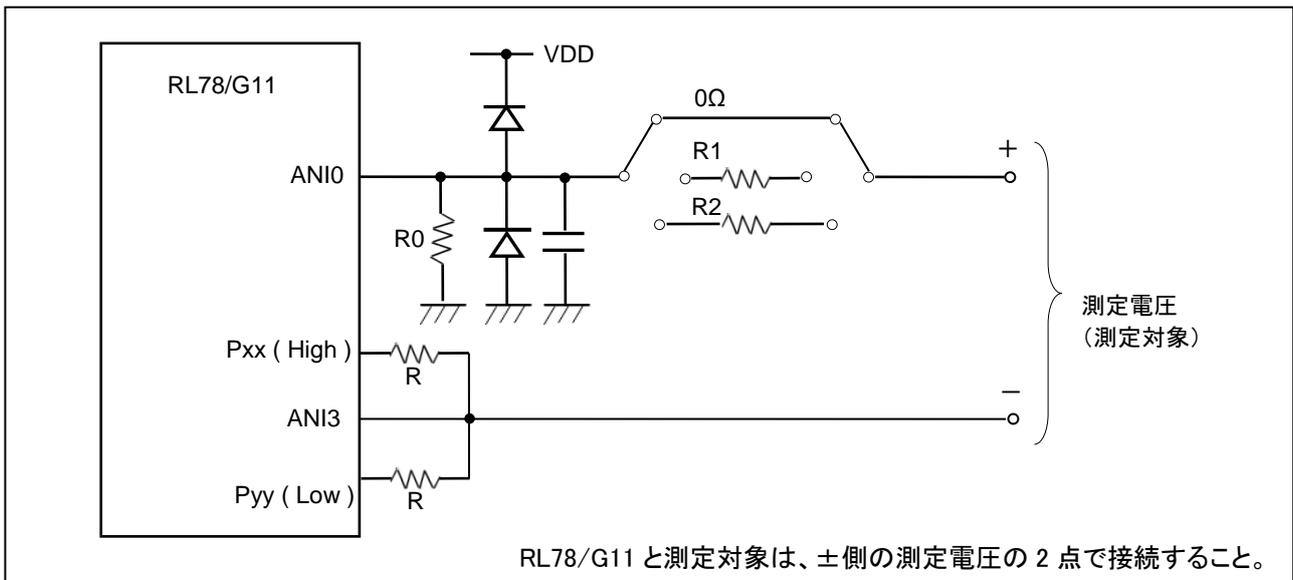


図 1.2 負電圧への対応

1.2 自動測定レンジへの対応

手動レンジ切り替えは、アナログ入力端子と測定電圧間の抵抗を切り替えましたが、自動測定レンジ切り替えでは、手動レンジ切り替えで GND 側に接続していた抵抗を切り替えます。

R1 の抵抗値を R0 の 9 分の 1、R2 の抵抗値を R0 の 99 分の 1 とします。Port 1 をハイ・インピーダンス、Port 2 をロウ・レベル出力にすると、R2 が選択され、ANI0 端子に印加される電圧は測定電圧の 1/100 となります。この状態で ANI0 端子に印加される電圧を測定します。その測定結果が、VDD/10~VDD/100 なら R1 を選択し、VDD/100~VDD/1000 なら、R1、R2 ともに非選択 (Port 1、Port 2 ともにハイ・インピーダンス) にします。再度、ANI0 端子に印加される電圧を測定することで、測定電圧を精度よく測ることができます。

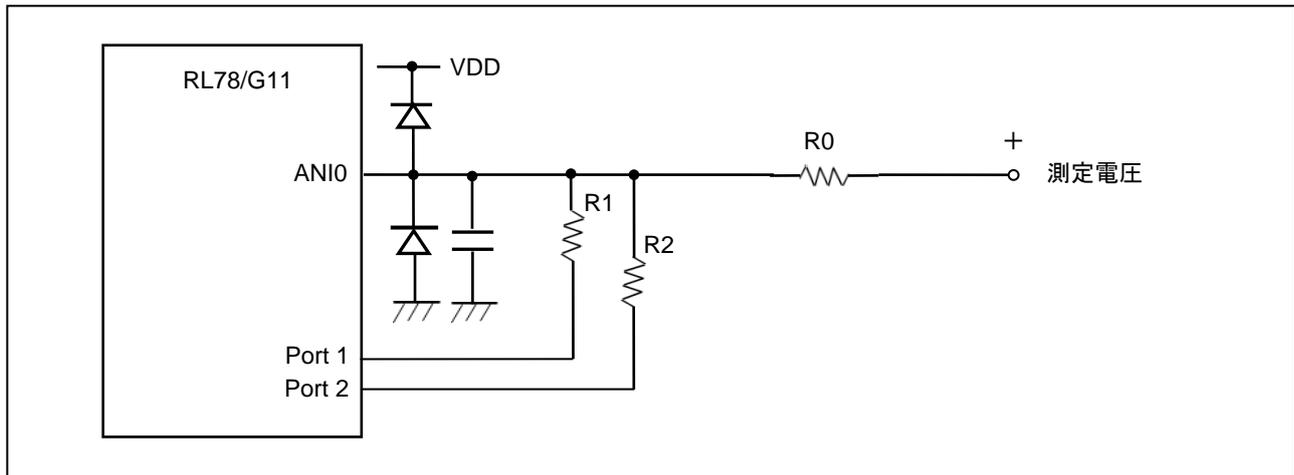


図 1.3 自動測定レンジ対応

1.3 抵抗値と測定レンジの検討

各レンジに抵抗分割用の GND レベルを生成します。GND レベルは、ポートのハイ・レベル出力とロウ・レベル出力を抵抗で分割して中間電位（RL78/G11 基準で $1/2 V_{DD}$ 程度）にします。

抵抗分割用の GND レベルを実際に測定し、その実測値を測定電圧の計算に使用します。

測定電圧範囲は最大 $\pm 100 V$ （交流の波高値は約 $\pm 150 V$ ）とします。また、入力インピーダンスを $1 M\Omega$ 程度として R_0 、 R_1 、 R_2 をそれぞれ $910 k\Omega$ 、 $82 k\Omega$ 、 $8.2 k\Omega$ とします。

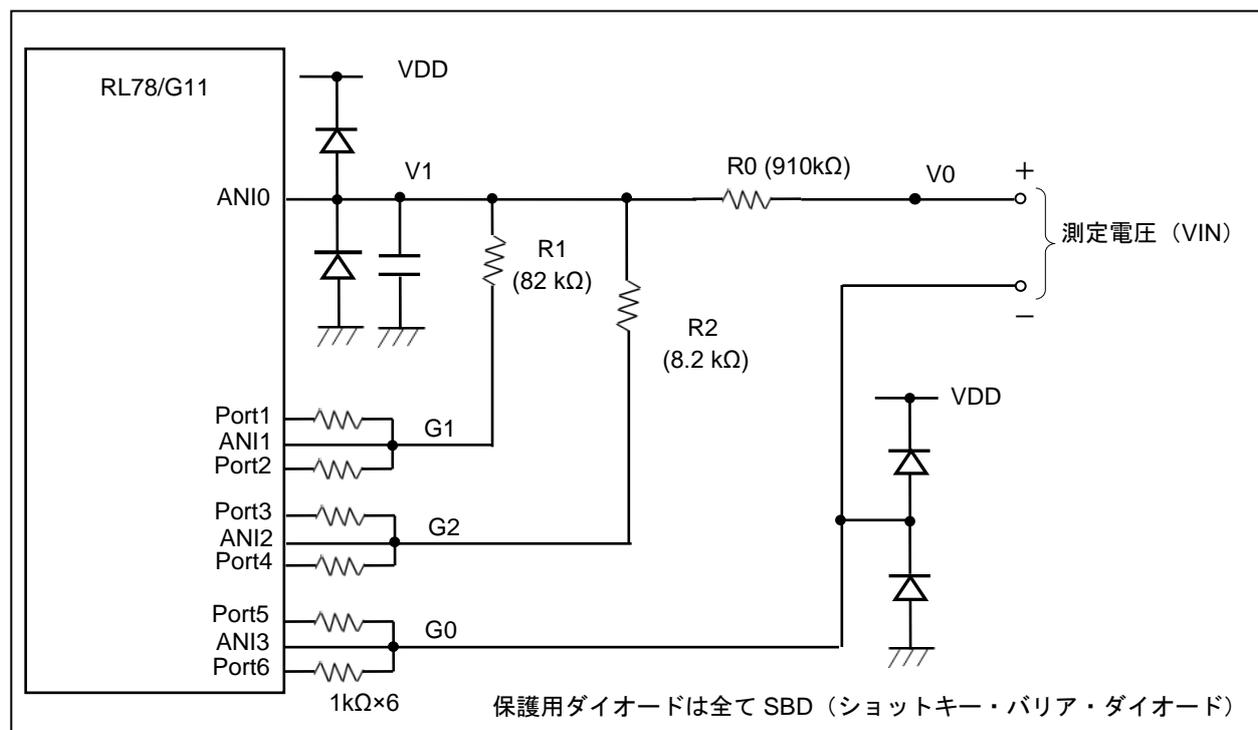


図 1.4 測定回路

1.4 測定結果の演算処理

電圧測定の演算処理で浮動小数点演算を利用すると、簡単にプログラムを作成できます。しかし、処理時間が長くなるばかりでなく、浮動小数点ライブラリを使用するためにプログラムサイズが増えます。本アプリケーションノートでは、浮動小数点演算を使用せず、データのビット数を意識した演算を行います。また、乗除算については、演算するビット長に合わせて RL78/G11 に内蔵された乗除積和算命令を利用します。

電圧分割抵抗 R_n (R_1 または R_2) が有効になっているとき、電圧 V_0 と実際に測定する電圧 (V_1) との関係はつぎのようになります。

$$\begin{aligned} V_1 &= (V_0 - G_n) \times R_n / (R_0 + R_n) + G_n \\ &= (V_0 \times R_n + G_n \times R_0) / (R_0 + R_n) \end{aligned}$$

この式を変形すると、

$$V_0 = \{ V_1 \times (R_0 + R_n) - G_n \times R_0 \} / R_n$$

よって、測定電圧 V_{IN} は V_0 から G_0 を引くことで得られるため、つぎのようになります。

$$\begin{aligned} V_{IN} &= V_0 - G_0 \\ &= \{ V_1 \times (R_0 + R_n) - G_n \times R_0 \} / R_n - G_0 \\ &= \{ V_1 \times (R_0 + R_n) - (G_n \times R_0 + G_0 \times R_n) \} / R_n \end{aligned}$$

上式の中括弧内の第 1 項と第 2 項を比較することで、測定電圧が正電圧か負電圧かを判断することができます。第 1 項が大きければ正電圧、第 1 項が小さければ負電圧になります。

1.5 内部基準電圧による補正処理

A/D コンバータの基準電圧は VDD を使用します。VDD 電源としてアルカリ電池を 3 本使用する場合、電池の出力電圧は約 4.8 V (1.6 V × 3) から約 3 V (1.0 V × 3) まで変化します。これでは、A/D コンバータの基準電圧が 30 % 以上変動することになり、A/D 変換精度が良くありません。そのため、RL78/G11 に内蔵された内部基準電圧 (TYP. 1.45 V) を A/D 変換することで測定結果の補正を行います。ただし、内部基準電圧は 5% 程度のばらつきがあります。

内部基準電圧を A/D 変換した結果 (SAR) は以下のようになります。

$$\text{SAR} = (1.45 / \text{VDD}) \times 1024$$

この式を変形すると、

$$\text{VDD} = (1.45 / \text{SAR}) \times 1024$$

見方を少し変えると、1 ビットの電圧は $1.45 / \text{SAR}$ と表すことができます。

よって、本アプリケーションノートでは、測定結果の電圧を次のように表します。

$$\text{VIN} = [\{ \text{V1} \times (\text{R0} + \text{Rn}) - (\text{Gn} \times \text{R0} + \text{G0} \times \text{Rn}) \} / \text{Rn}] \times 1.45 / \text{SAR}$$

この計算を行う場合は、浮動小数点演算ではなく整数演算を行うため、次のように補正処理します。

- (1) 抵抗値は 0.1 kΩ 単位で計算する。
- (2) 1.45 は“145 / 100”として考え、100 で割らずに 10 mV 単位として結果を求める。

測定電圧を 10 mV 単位で計算しても、測定電圧が交流 100 V であれば 16 ビットで処理可能です。

また、本アプリケーションノートでは、測定電圧 (実効値) の計算を 32 ビット演算で行うために、除算処理を行います。除算処理はシフト命令で実行するため、除数は、± 100 V レンジで 16、± 10 V レンジでは 4 とします。これにより、測定電圧の計算結果 (g_vin_data[32]) が 12 ビット以下となり、計算結果 (g_vin_data[32]) の二乗を 32 個加算 (実効値の計算処理) しても 32 ビット演算で実現できます。

1.6 測定レンジの切り替え処理

測定電圧の測定を開始するときは、必ず R2 (8.2 kΩ) で中間電位 (G2) に接続した状態 (± 100 V レンジ) から開始します。必ず、上記設定で、測定電圧を 32 回計測します。その測定結果の絶対値の最大値によって、測定レンジを決定します。

R2 選択時の測定電圧の計算結果 (g_vin_data[32]) が、9 以下 (±1.5 V 相当) であれば測定レンジを ±1 V レンジにします。10 ~ 93 (±15 V 相当) であれば測定レンジを ±10 V レンジにします。計算結果 (g_vin_data[32]) が 937 (±150 V 相当) を超える場合は、オーバーレンジを表示します。測定完了後は、安全のために R2 (8.2 kΩ) を選択します。

計算結果 (g_vin_data[32]) に測定レンジ毎の定数 (± 100 V レンジで 16、± 10 V レンジでは 4) を掛けて 10 mV 単位を 1 V 単位に変更すると、測定電圧の絶対値が得られます。

1.7 AC 電源への対応

本アプリケーションノートでは、周波数 50 Hz ~ 60 Hz 程度の正弦波に対応しています。

正弦波を 1 ms 周期で 32 回サンプリングして、極性が反転するタイミング（ゼロクロス）の間隔を測定します。実効値の計算では、その期間（間隔）の測定電圧のみを対象とします。もし、ゼロクロスがなければ、最後の 20 サンプリングを対象として、実効値を計算します。

なお、本アプリケーションノートでは、脈流の測定には対応していません。脈流を測定するためには、サンプリング数を増やして実効値を計算する必要があります。

1.8 表示周期

表示の切り替え周期は 1 秒とします。内部基準電圧の測定後、1 ms 周期のインターバル・タイマの割り込み毎に測定電圧を計算して測定電圧（実効値）を決定します。その後、1 ms 周期のインターバル・タイマを停止し、測定電圧（実効値）を表示してスタンバイ状態（STOP モード）にします。以降、1 秒毎に上記動作を繰り返します。

もし、測定電圧（実効値）が 20 回連続して ± 10 mV 以下ならば、表示を消し、1 秒周期のインターバル・タイマを停止してスタンバイ状態（STOP モード）にします。スイッチが押下（INTP0 発生）されると、再び、電圧測定と表示を開始します。

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/G11 (R5F1056)
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 : 16 MHz ● CPU/周辺ハードウェア・クロック : 16 MHz
動作電圧	3.0V~5.5V LVD 動作モード : リセット・モード、電圧 : VLVD2 = Min. 3.00 V
統合開発環境(CS+)	ルネサス エレクトロニクス製 CS+ V5.00.00
統合開発環境(e2studio)	ルネサス エレクトロニクス製 e2 studio 5.3.0.023
コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.04.00
使用ボード	RL78/G11 ターゲット・ボード+LCD

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 CC-RL (R01AN2575J) アプリケーションノート

RL78/G10 平方根プログラム CC-RL (R01AN3079J) 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。

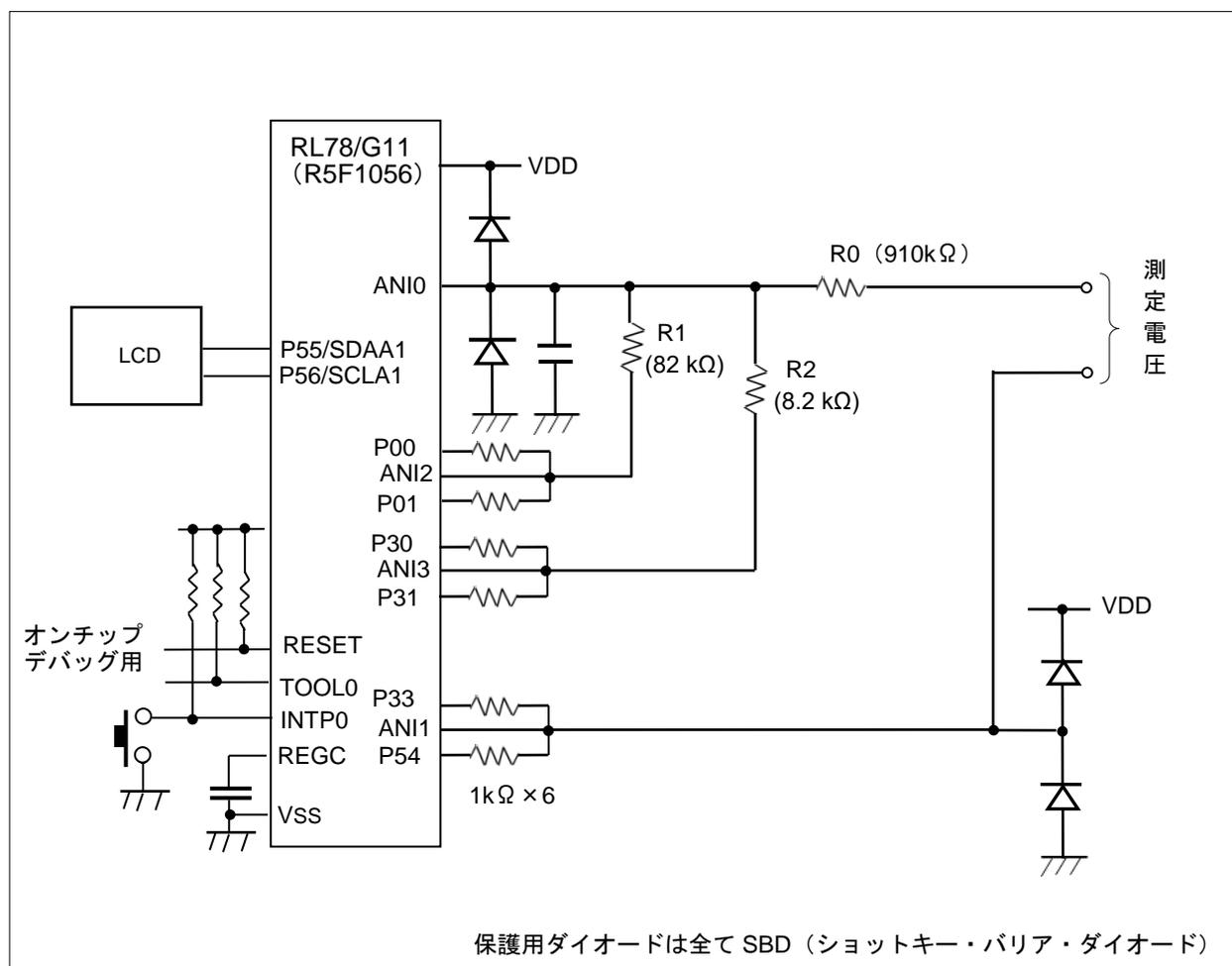


図 4.1 ハードウェア構成

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
ANI0~ANI3	入力	アナログ信号入力
P00, P01, P30, P31 P33, P54	出力	各測定レンジの中間電圧生成に使用
P55/SDAA1 P56/SCLA1	入出力	LCD への表示データ転送用の I2C バス制御用
INTP0/P137	入力	SW 入力

5. ソフトウェア説明

5.1 動作概要

本アプリケーションノートでは、RL78/G11 に搭載された A/D コンバータを利用して電圧計を実現します。

- (1) 初期設定完了後、LCD に “Renesas RL78/G11” を表示して、スタンバイ状態 (STOP モード) にします。
- (2) スイッチが押されると、INTP0 が発生してスタンバイ状態を解除します。
- (3) 1 秒タイマを起動します。
- (4) 内部基準電圧の測定を行います。1 ms 周期のインターバル・タイマの割り込み毎に測定電圧を測定します。アナログ入力電圧の測定は、ANI0 から ANI3 まで順に行います。A/D 変換結果は DTC で内蔵 RAM に待避させます。
- (5) 1 ms 周期で測定電圧を 32 回サンプリングして、測定電圧の絶対値の最大値と極性が反転するタイミング (ゼロクロス) の間隔を測定します。
- (6) 測定電圧の絶対値の最大値を判定して、必要に応じて、測定レンジを切り替え、再度 1ms 毎に 32 回電圧を測定します。また、極性が反転するタイミング (ゼロクロス) の間隔も測定します。
- (7) 32 回の電圧測定が完了したら、極性が反転したタイミング間の測定電圧 (実効値) を求めます。ゼロクロスがなければ、最後の 20 回分を対象として、実効値を計算します。
- (8) 1 ms 周期のインターバル・タイマを停止し、求めた測定電圧 (実効値) を LCD に表示し、スタンバイ状態 (STOP モード) にします。以降、1 秒毎に(4)から(8)を繰り返します。
ただし、測定電圧 (実効値) が 20 回連続して ± 10 mV 以下ならば、表示を消し、1 秒周期のインターバル・タイマを停止してスタンバイ状態 (STOP モード) にします。スイッチが押下 (INTP0 発生) されると、(2)から再開します。

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・バイト設定値

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 2.45V (2.40V~2.50V)
0x000C2	0b11101001	HS モード、HOCO : 16MHz
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
COMBYTE	0x00	LCD へのコマンド書き込みモード
DATABYTE	0x80	LCD へのデータ書き込みモード
CLRDISP	0x01	LCD クリア命令
HOMEPOSI	0x02	LCD ホームポジション命令
LCD_Mode	0b00111000	LCD 表示モード (2 ライン表示)
DISPON	0b00001111	LCD 表示オン
ENTRY_Mode	0b00000110	LCD 表示データを右にシフトしていく
SLAVEADDR	0xA0	LCD の I2C バス上でのスレーブ・アドレス
R0_OHM	10000	R0 の抵抗値 (1M Ω)
R1_OHM	820	R1 の抵抗値 (82k Ω)
R2_OHM	82	R2 の抵抗値 (8.2k Ω)
R0_R2	R0_OHM + R2_OHM	R0 と R2 の合成抵抗値
R0_R1	R0_OHM + R1_OHM	R0 と R1 の合成抵抗値
LIMIT100	937	100V レンジの最大値
LIMIT10	93	10V レンジの最大値
LIMIT1	8	1V レンジの最大値
SLEEPTIME	20	休止するまでの無入力回数

5.4 変数一覧

表 5.3 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

Type	Variable Name	Contents	Function Used
uint16_t	ad_buff[4]	A/D 変換結果バッファ	R_ADC_Init(), r_adc_interrupt()
uint16_t	g_vin_data[32]	測定結果バッファ	R_ADC_Init(), r_adc_interrupt()
uint16_t	g_vref	内部基準電圧測定結果	R_ADC_Init(), r_adc_interrupt(), main()
uint16_t	g_vmax	測定した最大電圧	R_ADC_Init(), r_adc_interrupt(), main()
uint8_t	g_range	測定レンジ	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_times	測定回数	main(),R_MAIN_UserInit(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_vin_sign	測定結果の符号	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_sign_chng1	ゼロクロス点 1	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_sign_chng2	ゼロクロス点 2	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint8_t	g_sign_chng3	ゼロクロス点 3	main(), r_adc_interrupt(), R_ADC_inter_ref()
uint16_t	g_voltage	測定結果 (10mV 単位)	main()
uint8_t	g_period	有効測定データ数	main()
uint8_t	g_start	有効データ開始位置	main()
uint8_t	g_stpcnt	電圧未検出カウンタ	main(),R_MAIN_UserInit()
int8_t	g_2ndline	表示データバッファ	main()
st_dtc_data_t	dtc_controldata_0	DTC のパラメータ	R_DTC_Create(), R_ADC_DTC_Init(), r_adc_interrupt(), main()
uint8_t	g_iica_status	IICA1 のステータス	R_IICA_bus_check, r_iica_interrupt(), R_IICA_wait_comend
uint8_t*	gp_iica_tx_address	IICA1 送信ポインタ	R_IICA_Master_Send(), r_iica_interrupt()
uint8_t	g_iica_tx_cnt	IICA1 送信カウンタ	R_IICA_Master_Send(), r_iica_interrupt()
uint8_t	g_write_commmand[2]	LCD コマンドバッファ	set_command()
uint8_t	g_write_data[2]	LCD データバッファ	set_data()

5.5 関数一覧

表 5.4 に使用する関数一覧を示します。

表 5.4 関数一覧

関数名	概要
R_MAIN_UserInit()	各種機能の起動等
InitialiseDisplay	LCD 表示の初期化
wait_time	60 μ s 単位での時間待ち合わせ処理
wait_60us	60 μ s 待ち合わせ処理
set_command	LCD コマンド設定
set_data	LCD 表示データ設定
DisplayString	LCD への ASCII 文字列の表示
R_IICA_Master_Send	IICA での LCD への送信処理
R_IICA_wait_comend	IICA の転送完了待ち
R_IICA_StopCondition	IICA からのストップ・コンディション発行
R_IICA_bus_check	IIC バスの状態を確認し、スタート・コンディション発行
r_iica_interrupt	IICA 割り込み処理
r_intp0_interrupt	INTP0 割り込み処理
r_it_interrupt	INTIT00 割り込み処理
R_ADC_inter_ref	内部基準電圧計測処理
R_ADC_DTC_Init	A/D と DTC の設定処理
r_tm01_start	TM01 (1ms インターバル・タイマ) 起動処理
delay_us	1us 待ち
r_adc_interrupt	A/D 変換結果の DTC 転送完了処理
__ssqrt	32 ビットデータの平方根 (詳細は R01AN3079 参照)
__r_mul32	16 ビット \times 16 ビット
__r_div32	32 ビット \div 16 ビット
__r_div16	16 ビット \div 8 ビット
__r_mod16	16 ビット \div 8 ビットの剰余演算
__r_rms16	16 ビットデータの自乗平均平方根

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_MAIN_UserInit

概要	使用する各機能の初期化
ヘッダ	r_cg_macrodriver.h、r_cg_userdefine.h
宣言	void R_MAIN_UserInit (void);
説明	使用する変数等を初期化します。
引数	なし
リターン値	なし
備考	なし

[関数名] InitialiseDisplay

概要	LCD 表示の初期化処理
ヘッダ	lcd.h
宣言	void InitialiseDisplay (void);
説明	表示器を初期化し、LCD 表示を初期化します。
引数	なし
リターン値	なし
備考	なし

[関数名] wait_time

概要	60us 単位での時間待ち
ヘッダ	r_cg_macrodriver.h
宣言	void wait_time(uint16_t time);
説明	60us × 引数で示された回数だけ時間待ち処理を行います。
引数	60us 単位の待ち時間
リターン値	なし
備考	なし

[関数名] wait_60us

概要	TM03 を用いた 60us の時間待ち
ヘッダ	r_cg_macrodriver.h
宣言	void wait_60us(void);
説明	TM03（ディレイカウンタ）を用いて 60us の時間待ち処理を行います。
引数	なし
リターン値	なし
備考	なし

[関数名] set_command

概要	I2C を介して、LCD にコマンドを送信します。
ヘッダ	r_cg_macrodriver.h、lcd.h、r_iica_user.h
宣言	void set_command(uint8_t comcode);
説明	I2C を介して引数で示されたコマンドを LCD コントローラに送信します。
引数	LCD へのコマンド
リターン値	なし
備考	なし

[関数名] set_data

概要	I2C を介して、LCD にコマンドを送信します。	
ヘッダ	r_cg_macrodriver.h、lcd.h、r_iica_user.h	
宣言	void set_data(uint8_t datacode);	
説明	I2C を介して引数で示された表示データを LCD コントローラに送信します。	
引数	LCD への表示データ	
リターン値	なし	
備考	なし	

[関数名] DisplayString

概要	LCD への ASCII 文字列の表示。	
ヘッダ	r_cg_macrodriver.h、lcd.h、r_iica_user.h	
宣言	void DisplayString(uint8_t position, __far int8_t * string);	
説明	I2C を介して、第 1 引数で示された LCD の位置に第 2 引数で示された ASCII 文字列を表示させます。	
引数	第 1 引数	LCD での表示開始位置
	第 2 引数	表示文字列のポインタ
リターン値	なし	
備考	なし	

[関数名] R_IICA_Master_Send

概要	I2C のスレーブにデータを送信開始	
ヘッダ	r_cg_macrodriver.h、r_iica_user.h	
宣言	MD_STATUS R_IICA_Master_Send(uint8_t adr, uint8_t * const tx_buf, uint16_t tx_num);	
説明	IICA1 を使用して、引数で指定されたスレーブに対して指定されたデータの送信を開始します。I2C バスが使用中ならエラー・ステータスを戻し、I2C バスの使用権を確保できたならスレーブ・アドレスを送信開始します。	
引数	第 1 引数	スレーブ・アドレス
	第 2 引数	送信データのポインタ
	第 3 引数	送信データ数
リターン値	送信ステータス :	
	MD_OK	正常開始
	MD_ERROR1	I2C バスが使用中
備考	なし	

[関数名] R_IICA_wait_comend

概要	IICA の転送完了待ち	
ヘッダ	r_cg_macrodriver.h、r_iica_user.h	
宣言	MD_STATUS R_IICA_wait_comend(void);	
説明	IICA1 のステータスを確認して、転送（通信）完了を待ちます。	
引数	なし	
リターン値	通信結果ステータス :	
	MD_OK	通信を正常に完了
	MD_ERROR3	スレーブから ACK 応答なし
備考	なし	

[関数名] R_IICA_StopCondition

概要	IICA から I2C バスにストップ・コンディションを発行
ヘッダ	r_cg_macrodriver.h、r_iica_user.h
宣言	void R_IICA_StopCondition(void);
説明	IICA から I2C バスにストップ・コンディションを発行し、ストップ・コンディションが検出されるのを待ちます。
引数	なし
リターン値	なし
備考	なし

[関数名] R_IICA_bus_check

概要	I2C バスの状態確認
ヘッダ	r_cg_macrodriver.h、r_iica_user.h
宣言	MD_STATUS R_IICA_bus_check(void);
説明	I2C バスの状態を確認し、バスが使用可能であれば、スタート・コンディションを発行し、100 回確認するまでにスタート・コンディションが検出できれば、正常終了します。
引数	なし
リターン値	送信ステータス : MD_OK 正常開始 MD_ERROR1 I2C バスが使用中
備考	なし

[関数名] r_iica_interrupt

概要	IICA1 の転送完了割り込み処理
ヘッダ	r_cg_macrodriver.h、r_iica_user.h
宣言	#pragma interrupt r_iica_interrupt(vect=INTIICA1,BANK=rb1) static void r_iica_interrupt(void)
説明	IICA1 の転送完了割り込み処理を行います。指定されたデータの転送が完了したなら、ステータスを正常終了にします。
引数	なし
リターン値	なし
備考	なし

[関数名] r_intp0_interrupt

概要	INTP0 割り込み処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h
宣言	#pragma interrupt r_intp0_interrupt(vect=INTP0) static void __near r_intp0_interrupt(void)
説明	SW の押下で起動され、8 ビット・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_it_interrupt

概要	インターバル・タイマ割り込み処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h
宣言	#pragma interrupt r_it_interrupt(vect=INTIT00) static void __near r_it_interrupt(void)
説明	1s ごとの INTIT00 で起動され、A/D で内部基準電圧を測定し、DTC と TM01 を設定し、1ms ごとの A/D のハードウェア・トリガを起動します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_ADC_inter_ref

概要	内部基準電圧の計測処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h、r_cg_adc.h
宣言	void R_ADC_inter_ref(void);
説明	内部基準電圧を計測して変数 g_vref に格納し、測定レンジを 2 (100V) に設定し、測定用パラメータを初期化します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_ADC_DTC_Init

概要	A/D と DTC の初期設定処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h、r_cg_adc.h、r_cg_dtc.h
宣言	void R_ADC_DTC_Init(void);
説明	ADC をハードウェア・トリガ、ANI3-0 のシングル・スキャンに設定し、DTC の転送用パラメータを再設定します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_tm01_start

概要	TM01 起動処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h
宣言	void r_tm01_start(void);
説明	A/D 変換のトリガ用の TM01 を割り込み禁止で起動します。
引数	なし
リターン値	なし
備考	なし

[関数名] delay_1u

概要	1us ウェイト処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h
宣言	void delay_1us(void);
説明	ADCE 設定から ADCS 設定までの 1us の時間を待ちます。
引数	なし
リターン値	なし
備考	なし

[関数名] r_adc_interrupt

概要	INTAD 割り込み処理
ヘッダ	r_cg_macrodriver.h、r_timer_user.h
宣言	#pragma interrupt r_adc_interrupt(vect=INTAD) static void __near r_adc_interrupt(void)
説明	4 チャンネル分の DTC 転送完了による割り込みで起動されます。 A/D 変換されたデータから測定レンジに応じた演算を行い、結果の最大値及び符号の変化点を記録します。32 回の測定が完了したら、TM01 を停止して A/D 変換を完了します。32 回未満なら、DTC の転送用パラメータを再設定します。
引数	なし
リターン値	なし
備考	なし

[関数名] __ssqrt

概要	32 ビットの平方根演算処理
ヘッダ	arith_lib.h
宣言	uint16_t __sqrt(uint32_t);
説明	引数で渡された 32 ビットのデータの平方根を求めます。
引数	32 ビットのデータ BC:上位 16 ビット, AX: 下位 16 ビット
リターン値	平方根 AX (=BC)
備考	詳細は R01AN3079 参照

[関数名] __r_mul32

概要	16 ビット×16 ビット
ヘッダ	r_cg_macrodriver.h、arith_lib.h
宣言	uint32_t __r_mul32(uint16_t multiplicand, uint16_t multiplier);
説明	16 ビットの 2 つの引数の乗算を行います。
引数	第 1 引数 被乗数 (AX) 第 2 引数 乗数 (BC)
リターン値	積 (BC-AX)
備考	なし

[関数名] __r_div32

概要	32 ビット÷16 ビット
ヘッダ	r_cg_macrodriver.h、arith_lib.h
宣言	uint32_t __r_div32(uint32_t dividend, uint16_t divisor);
説明	第 1 引数 (32 ビット) を第 2 引数 (16 ビット) で割ります。
引数	第 1 引数 被除数 (BC-AX) 第 2 引数 除数 (DE)
リターン値	商 (BC-AX)
備考	なし

[関数名] __r_div16

概要	16 ビット ÷ 8 ビット	
ヘッダ	r_cg_macrodriver.h、arith_lib.h	
宣言	uint16_t __r_div16(uint16_t dividend, uint8_t divisor);	
説明	第 1 引数 (16 ビット) を第 2 引数 (8 ビット) で割ります。	
引数	第 1 引数	被除数 (AX)
	第 2 引数	除数 (C)
リターン値	商	AX
備考	なし	

[関数名] __r_mod16

概要	16 ビット ÷ 8 ビットの剰余演算	
ヘッダ	r_cg_macrodriver.h、arith_lib.h	
宣言	uint8_t __r_mod16(uint16_t dividend, uint8_t divisor);	
説明	第 1 引数 (16 ビット) を第 2 引数 (8 ビット) で割った剰余を求めます。	
引数	第 1 引数	被除数 (AX)
	第 2 引数	除数 (C)
リターン値	剰余	A
備考	なし	

[関数名] __r_rms16

概要	16 ビットデータの自乗平均平方根演算処理	
ヘッダ	r_cg_macrodriver.h、arith_lib.h	
宣言	uint16_t __r_rms16(uint8_t start, uint18_t data_no);	
説明	想定結果 (_g_vin_data[]) の第 1 引数で示されるデータから第 2 引数で示される個数のデータの自乗平均平方根を求めます。	
引数	第 1 引数	開始データ (A)
	第 2 引数	データ数 (X)
リターン値	自乗平均平方根	AX
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。

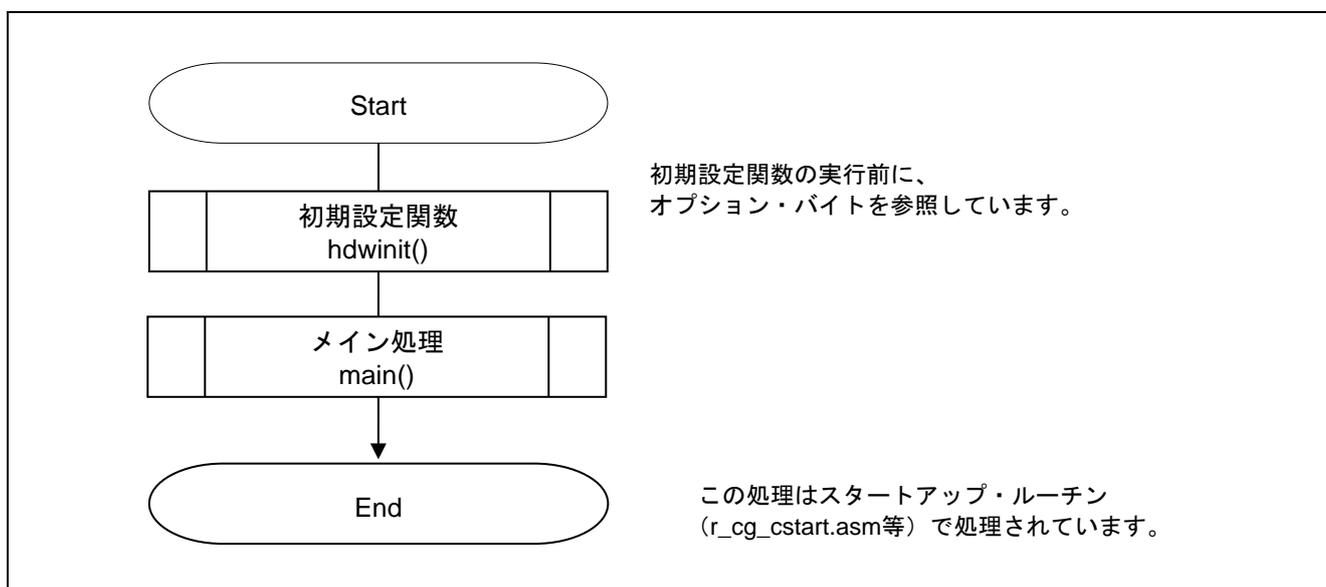


図 5.1 全体フロー

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。

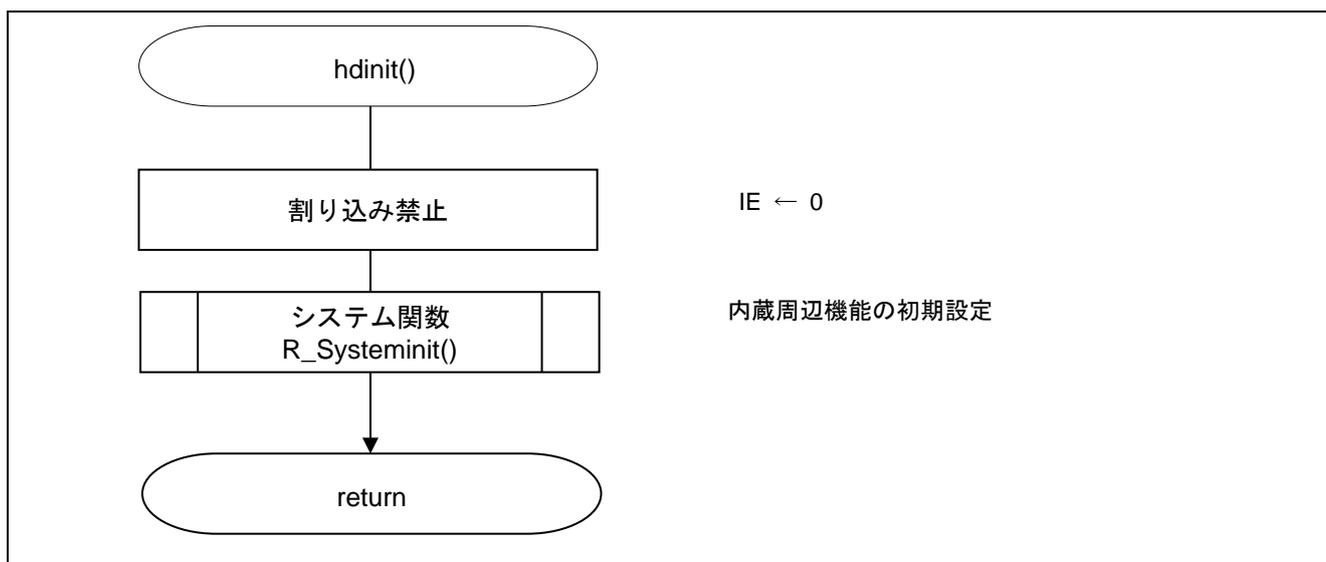


図 5.2 初期設定関数

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。

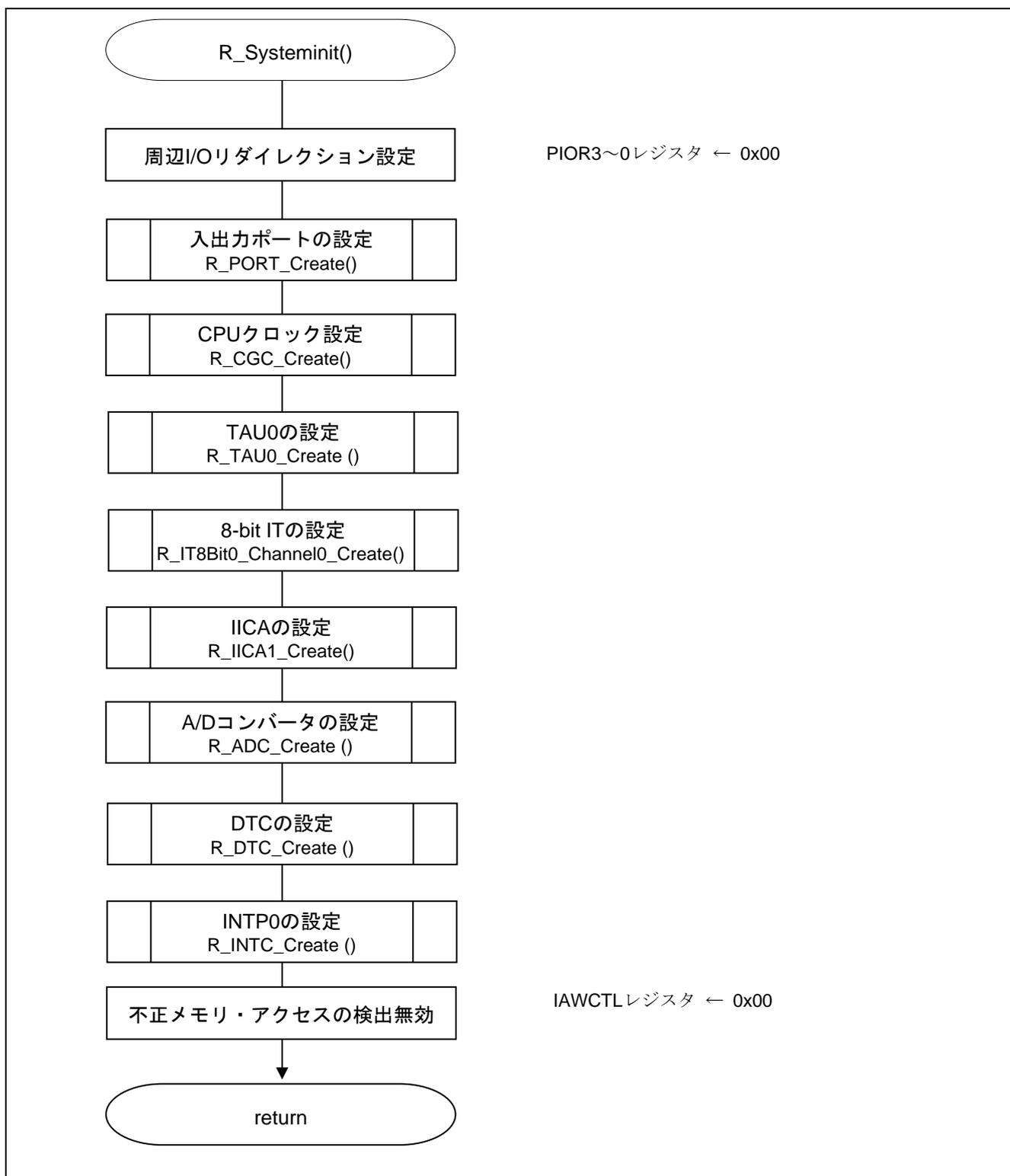


図 5.3 システム関数

5.7.3 入出力ポートの設定

図 5.4 に入出力ポート設定のフローチャートを示します。

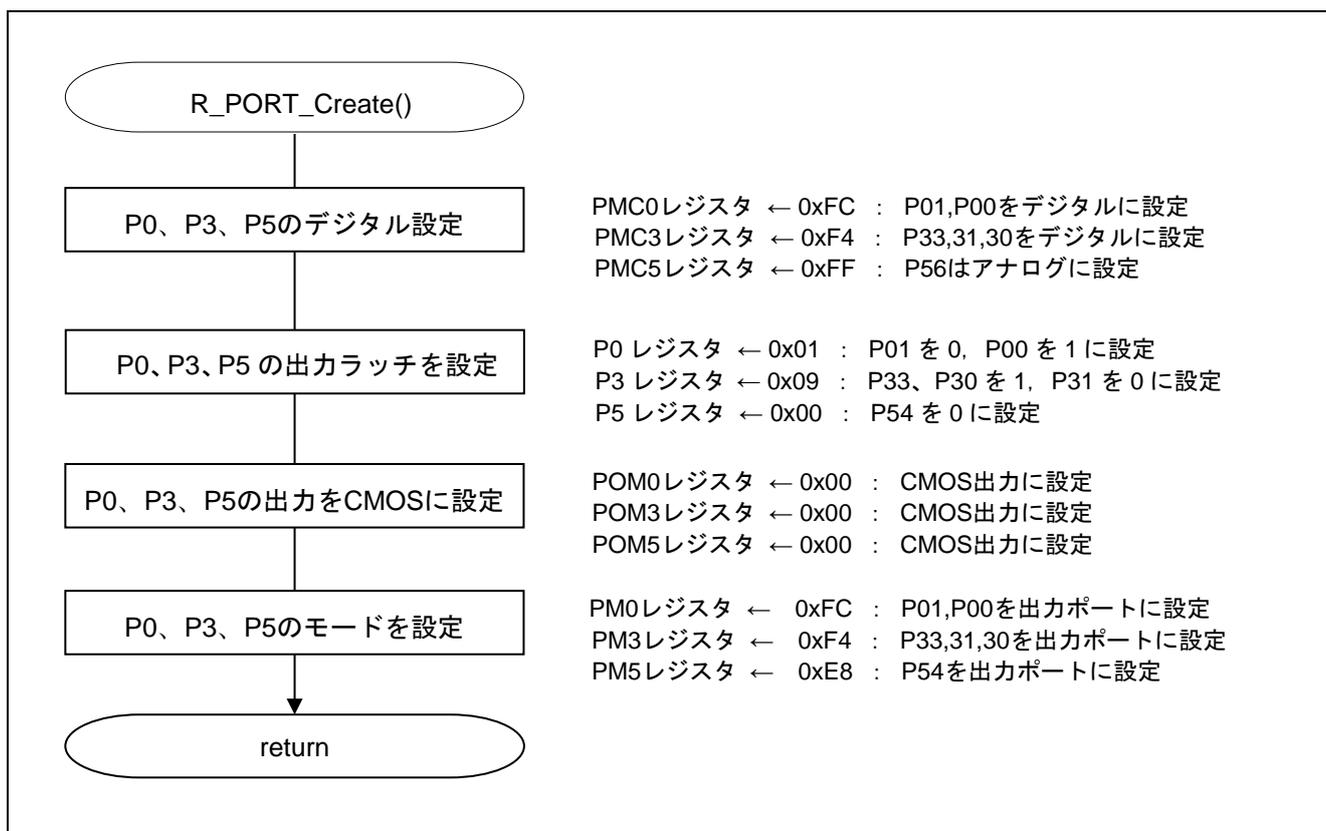


図 5.4 入出力ポート設定

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。

5.7.4 CPU クロック設定

図 5.5 に CPU クロック設定のフローチャートを示します。

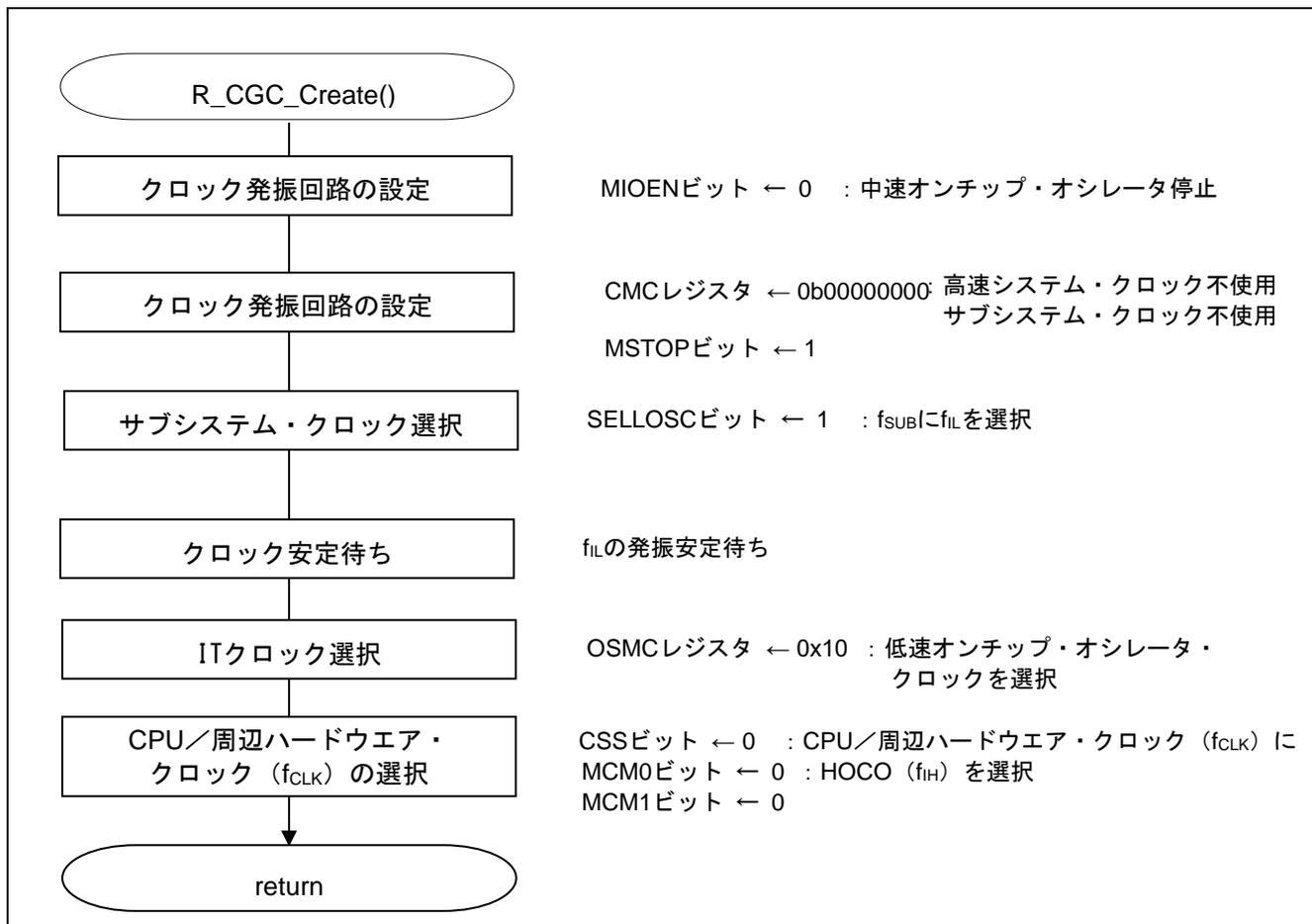


図 5.5 CPU クロックの設定

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。

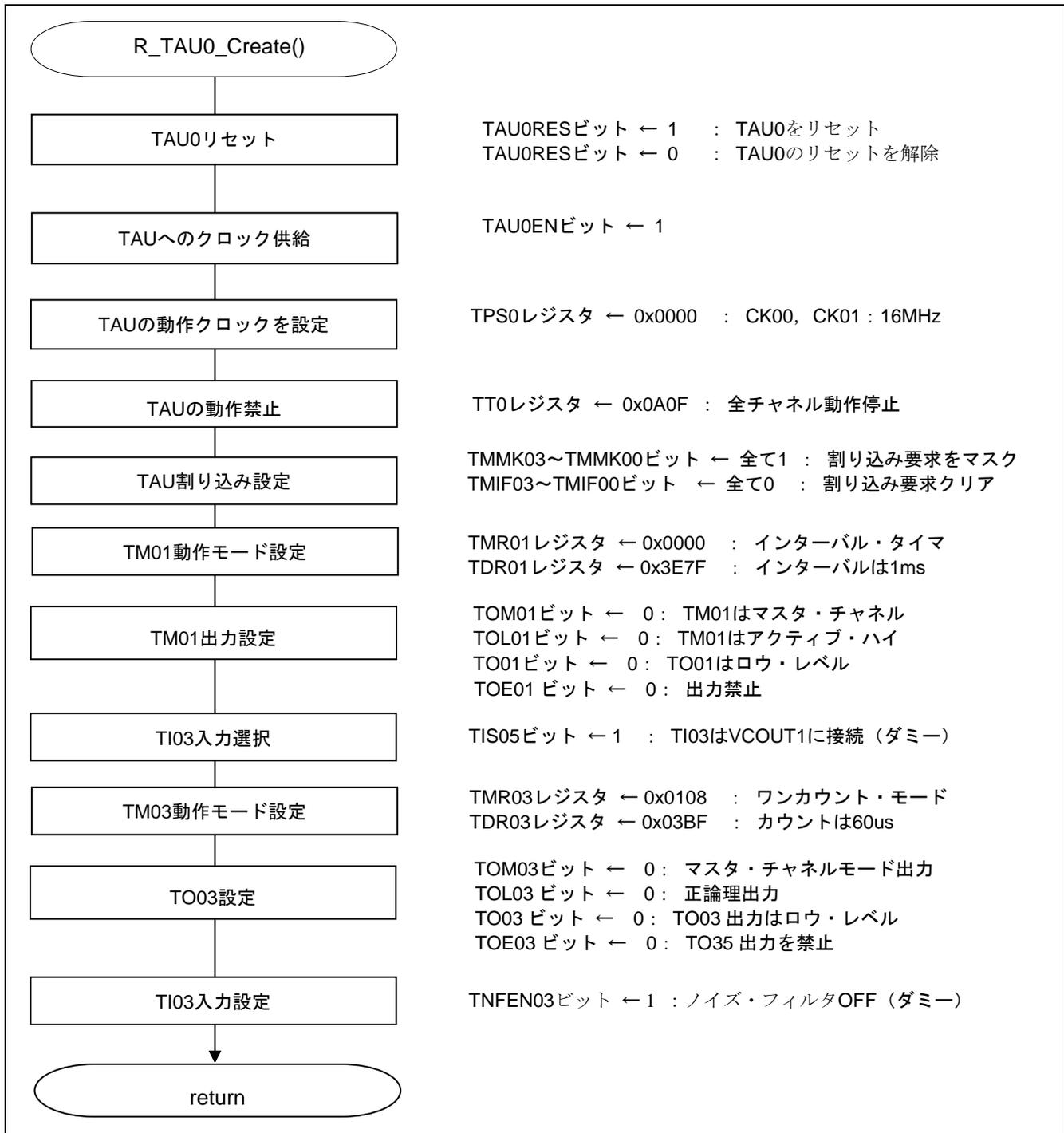


図 5.6 タイマ・アレイ・ユニットの設定

5.7.6 8ビット・インターバル・タイマの設定

図 5.7 に 8 ビット・インターバル・タイマの設定のフローチャートを示します。

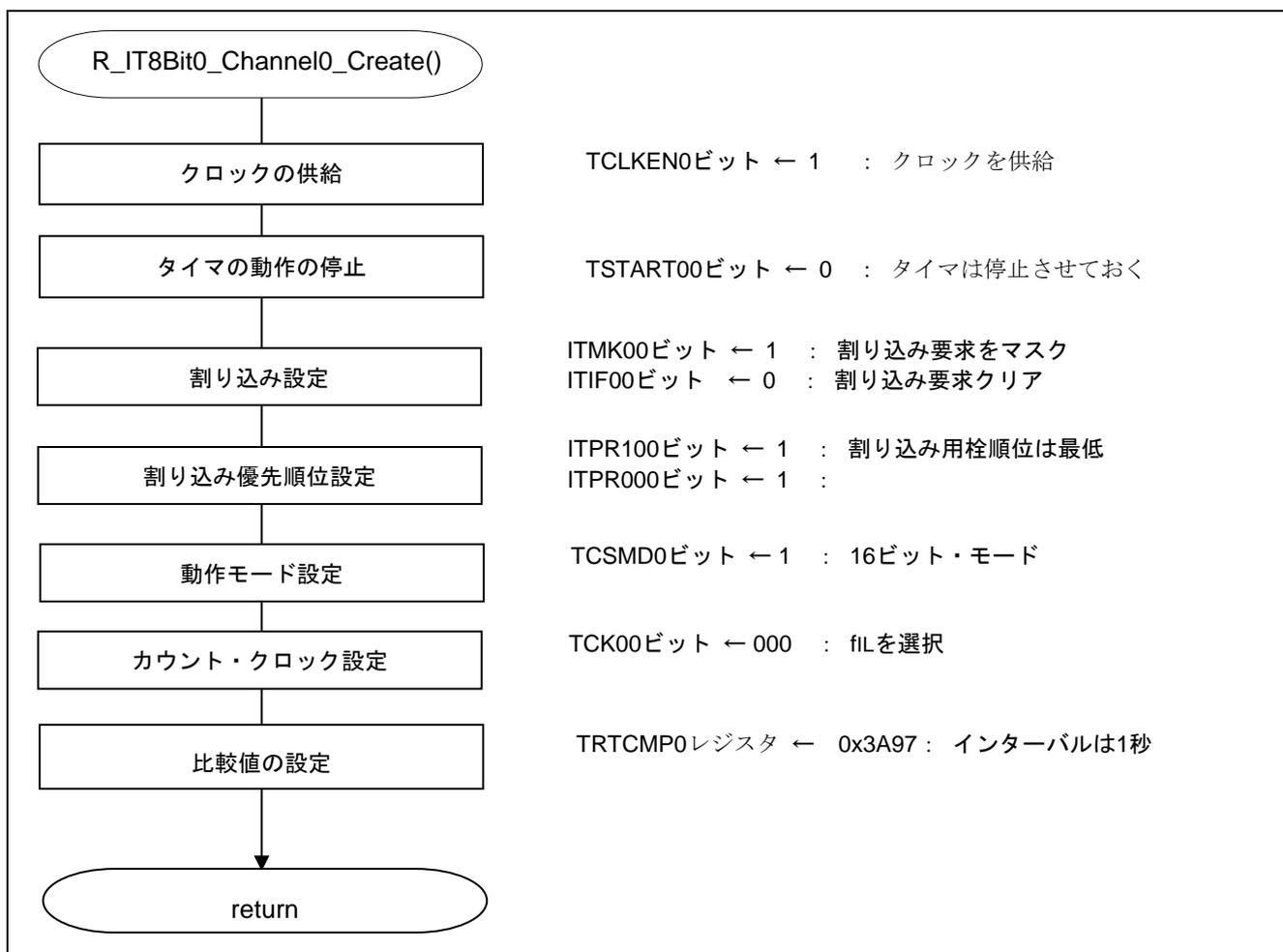


図 5.7 8ビット・インターバル・タイマの設定

5.7.7 IICA の設定

図 5.8 に IICA の設定のフローチャートを示します。

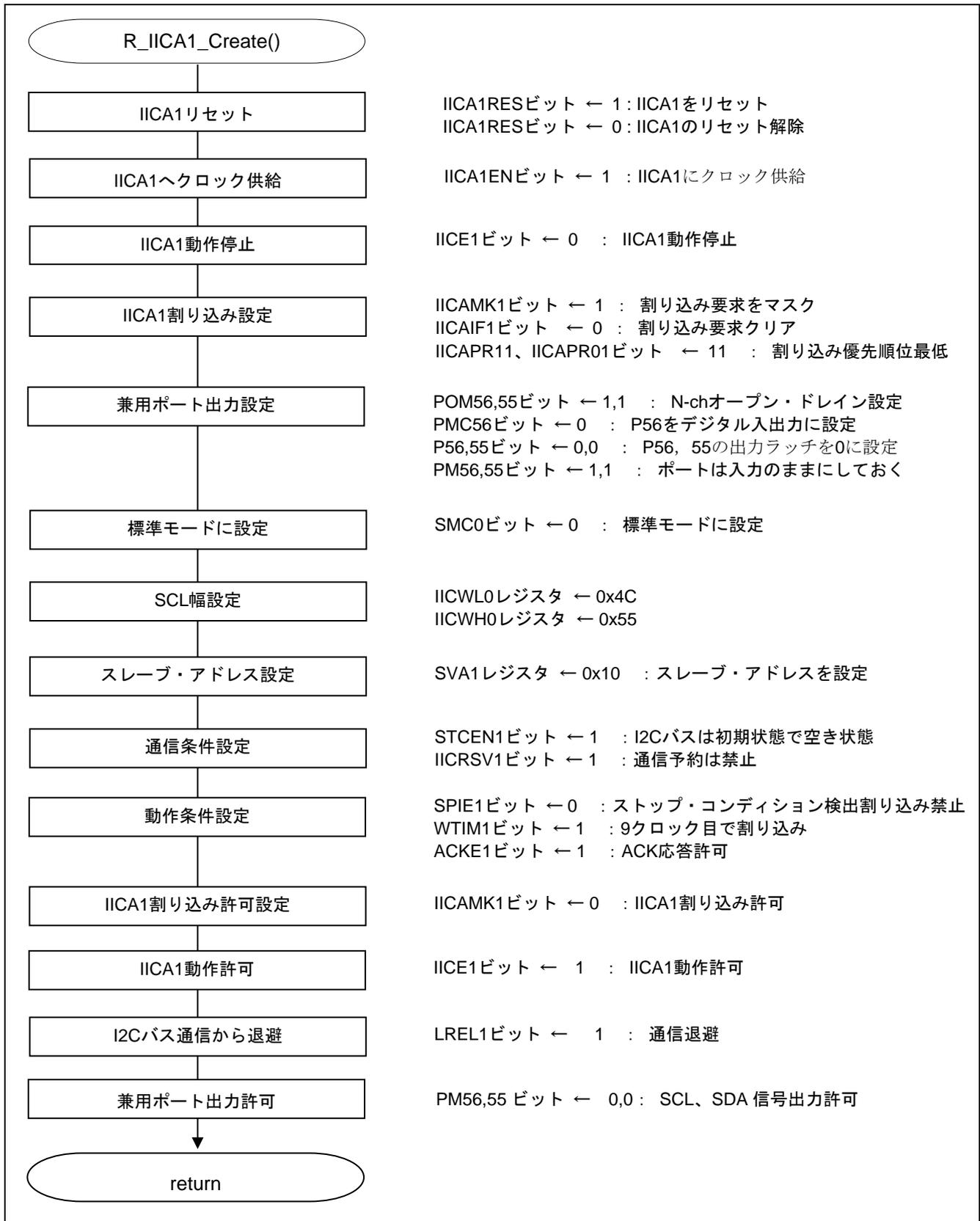


図 5.8 IICA の設定

5.7.8 A/D コンバータの設定

図 5.9 に A/D コンバータの設定のフローチャートを示します。

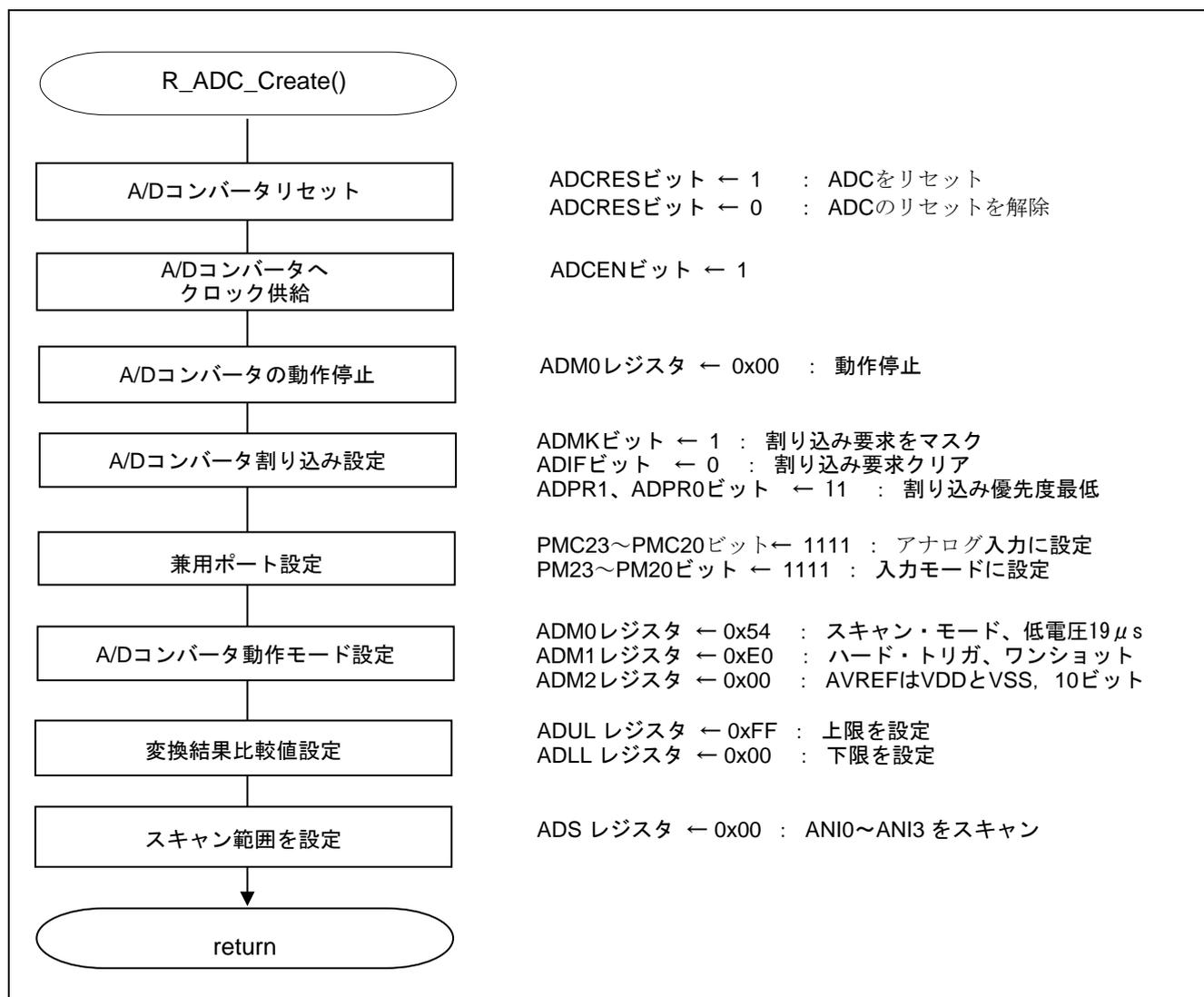


図 5.9 A/D コンバータの設定

5.7.9 DTC の設定

図 5.10 に DTC の設定のフローチャートを示します。

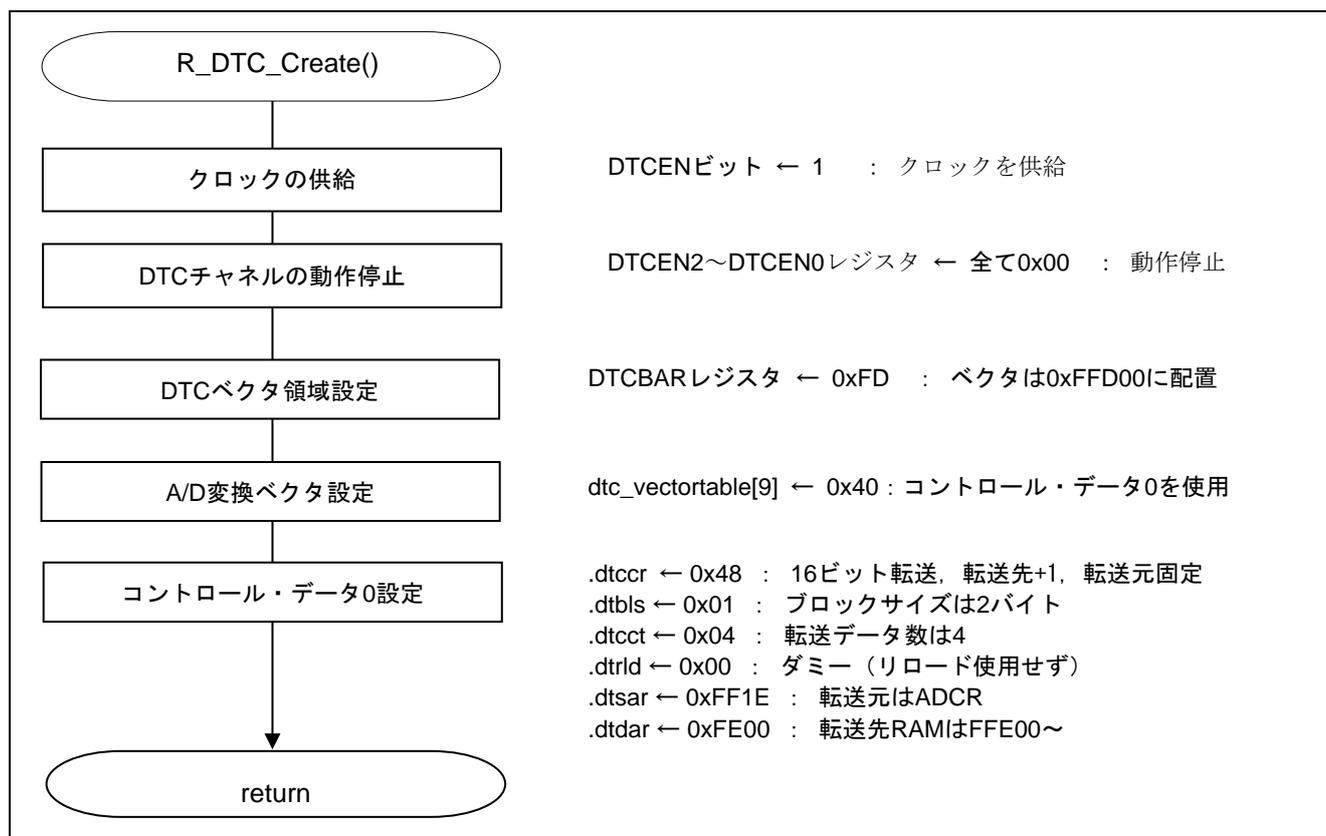


図 5.10 DTC の設定

5.7.10 INTP0 の設定

図 5.11 に INTP0 の設定のフローチャートを示します。

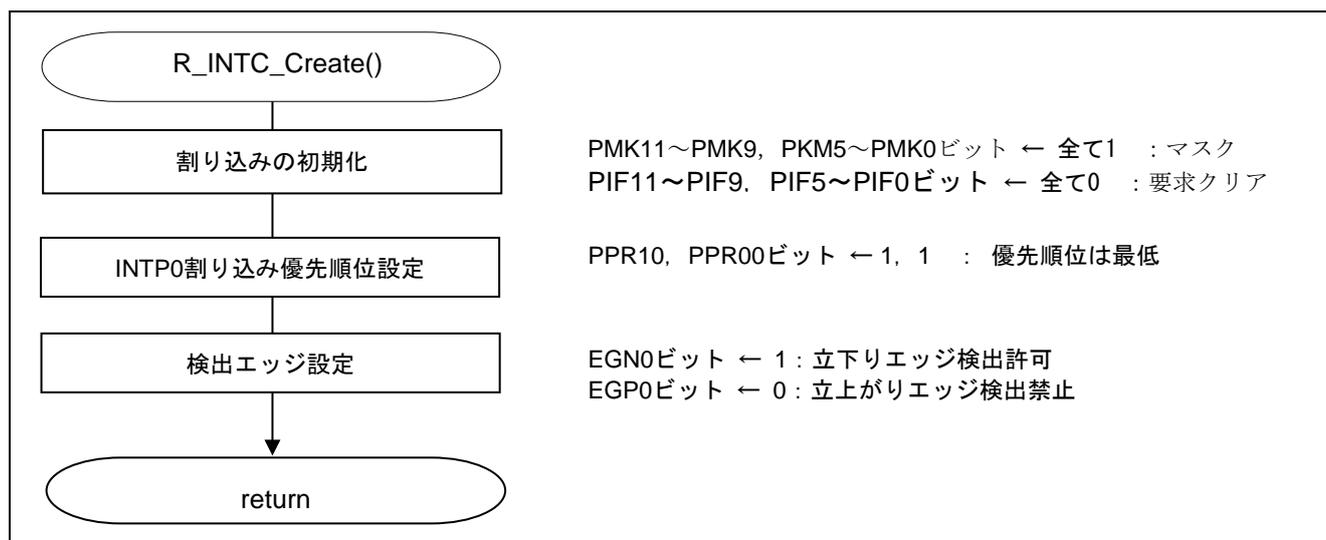


図 5.11 INTP0 の設定

5.7.11 メイン処理

図 5.12～図 5.14 にメイン処理のフローチャートを示します。

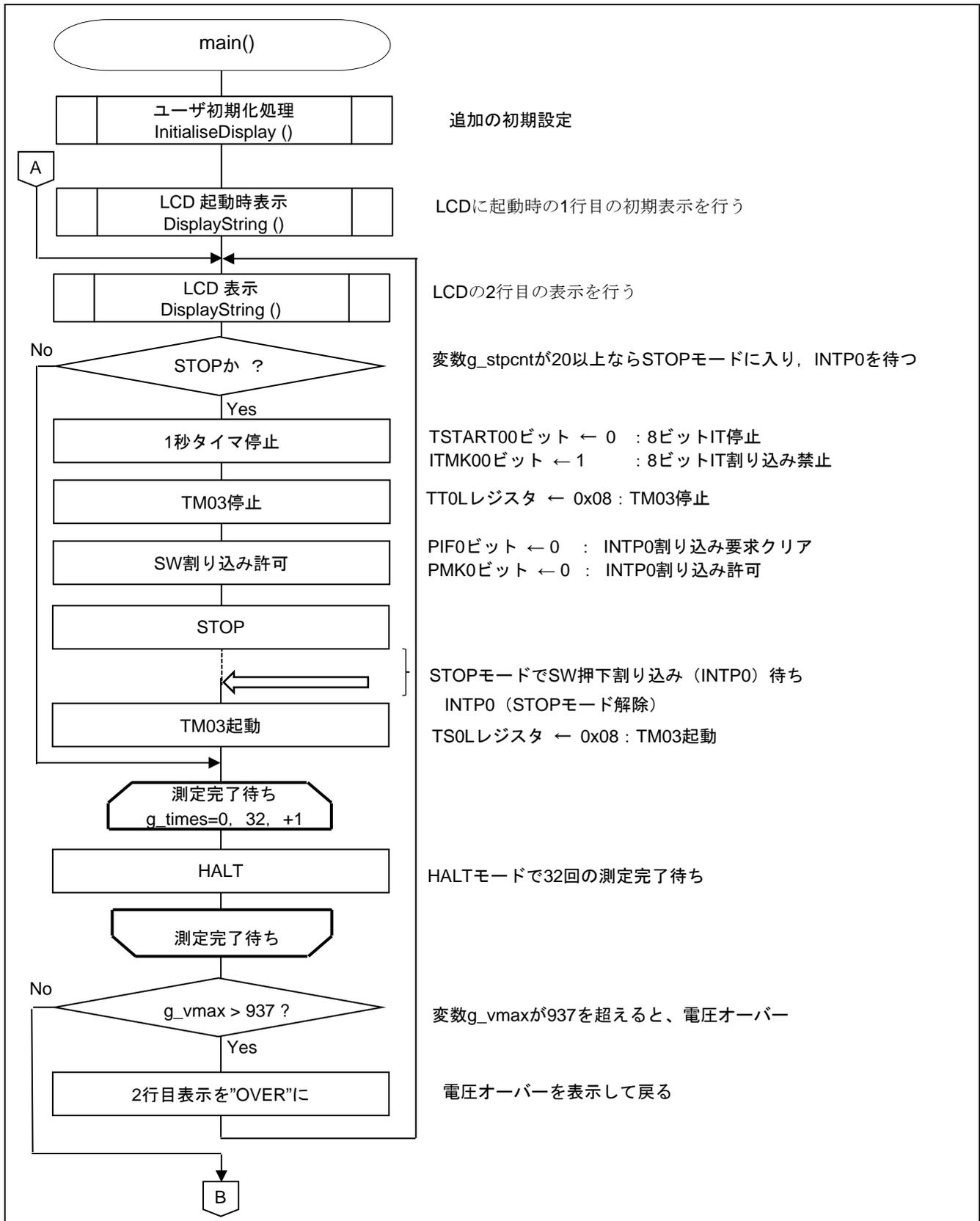


図 5.12 メイン処理 (1/3)

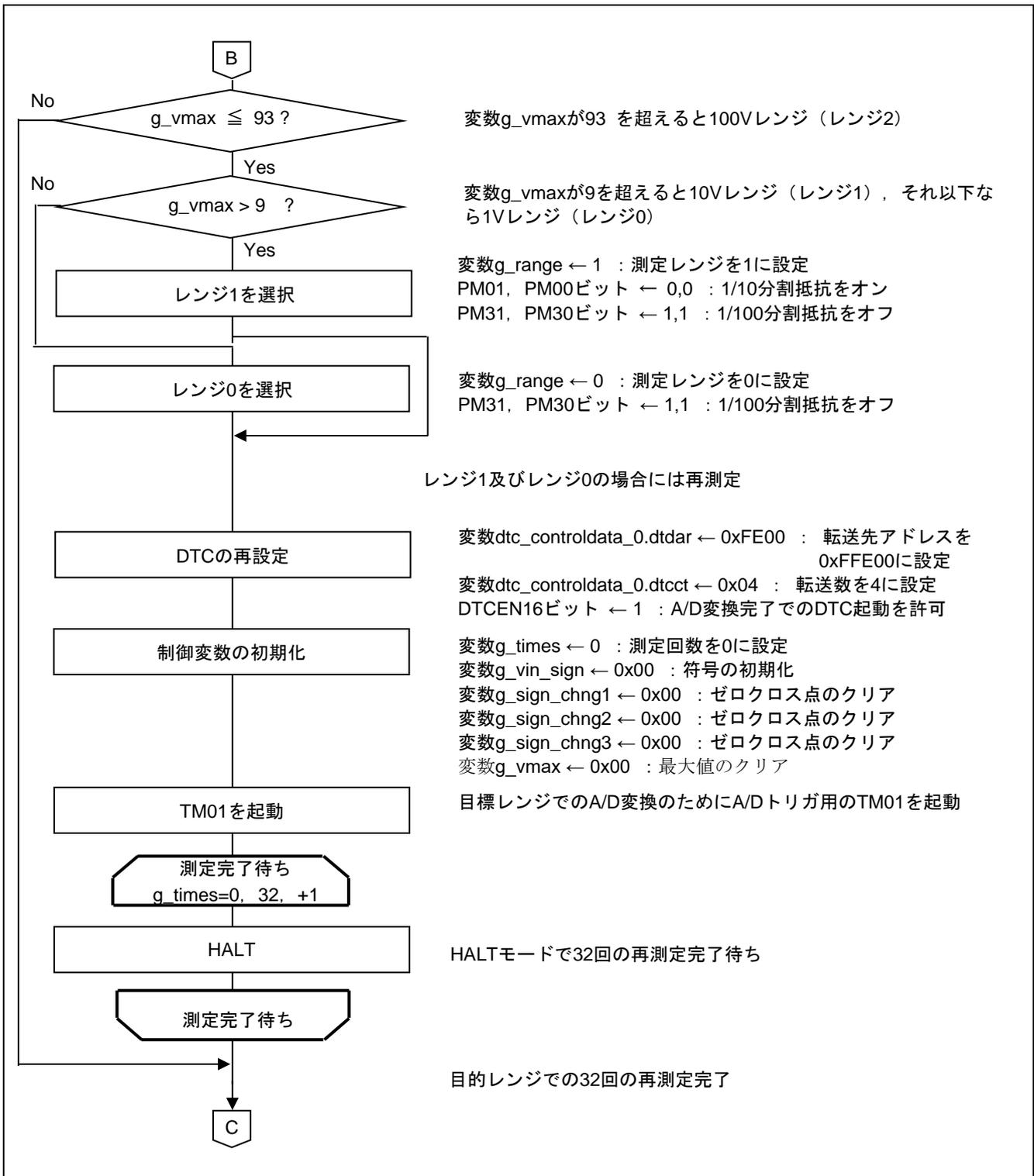


図 5.13 メイン処理 (2/3)

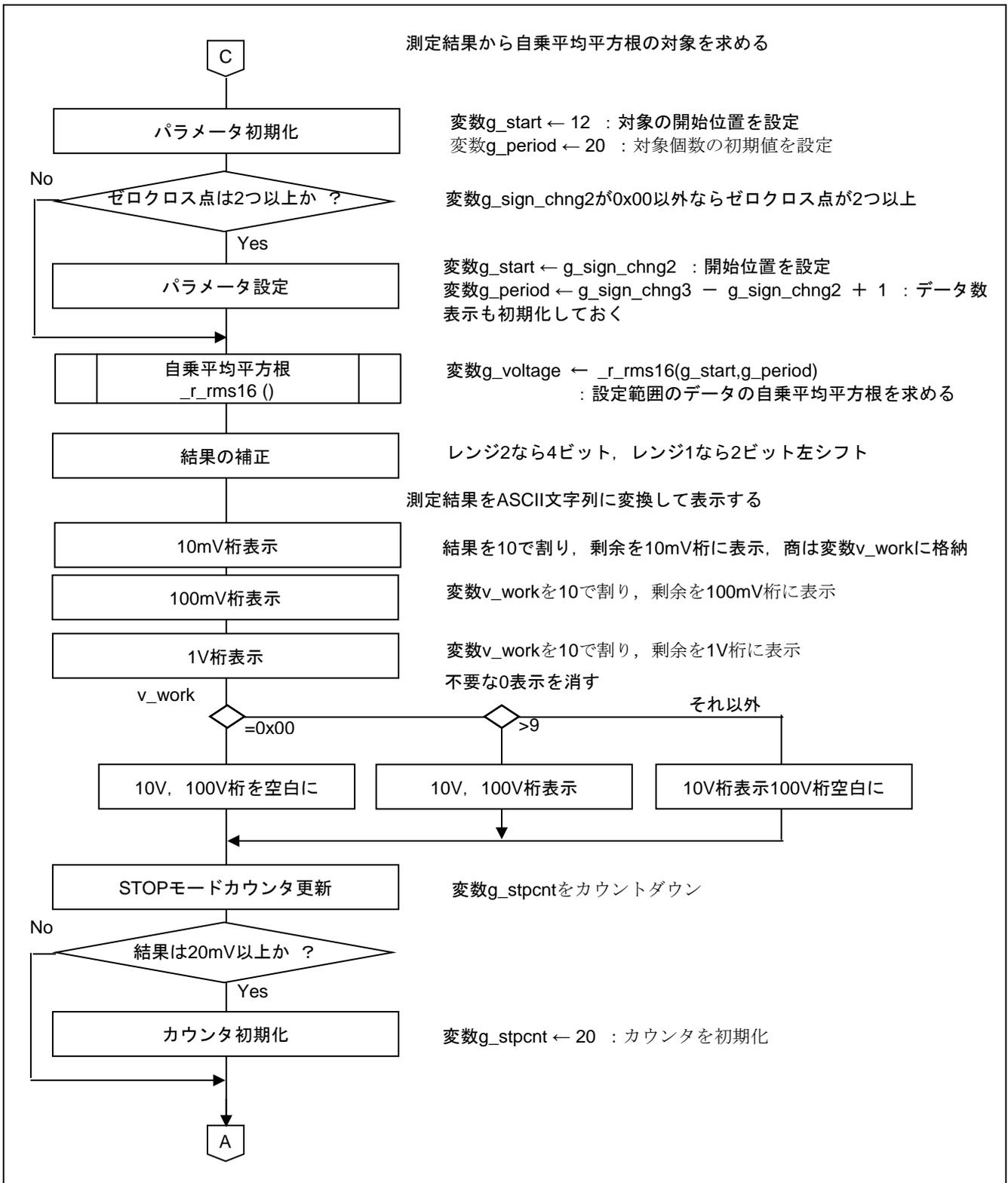


図 5.14 メイン処理 (3/3)

5.7.12 R_MAIN_UserInit 処理

図 5.15 に R_MAIN_UserInit 処理のフローチャートを示します。

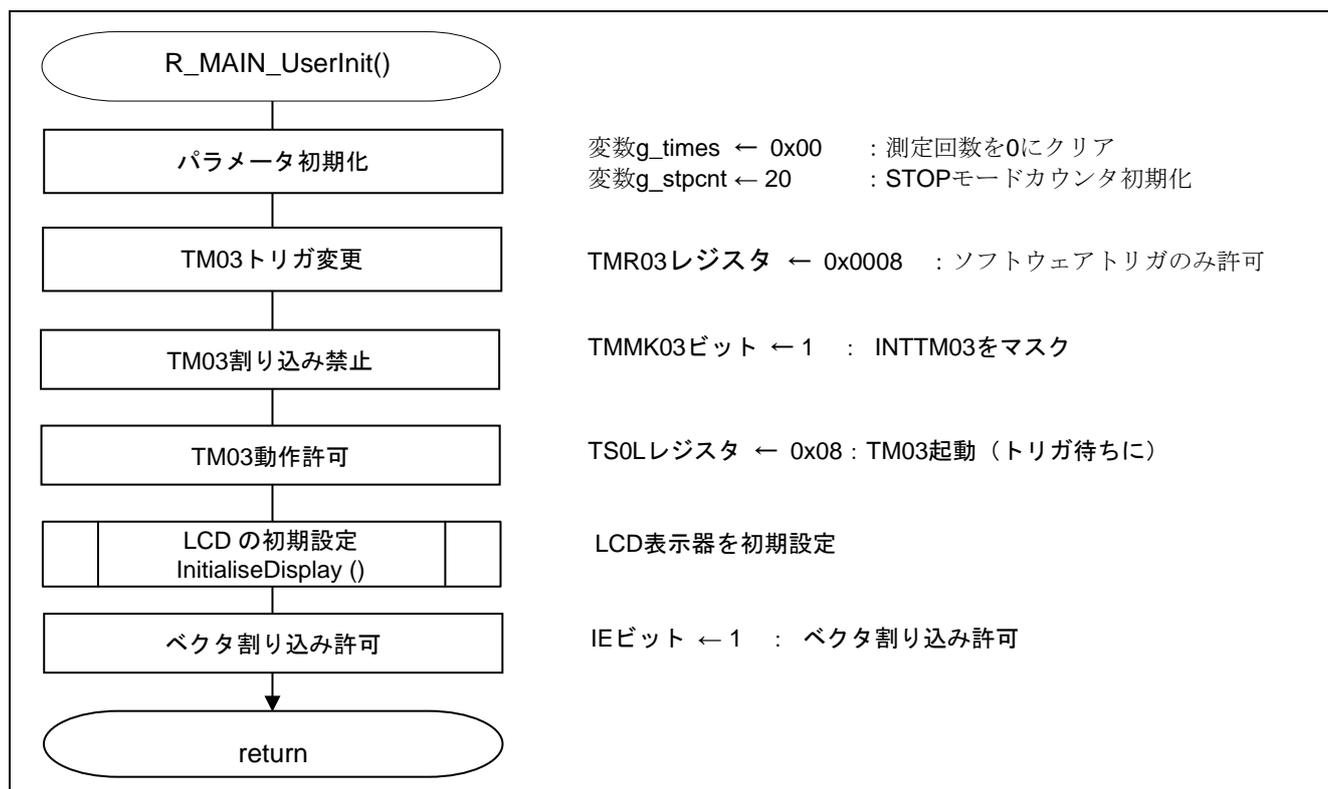


図 5.15 R_MAIN_UserInit 処理

5.7.13 LCD の初期設定

図 5.16 に LCD の初期設定処理のフローチャートを示します。

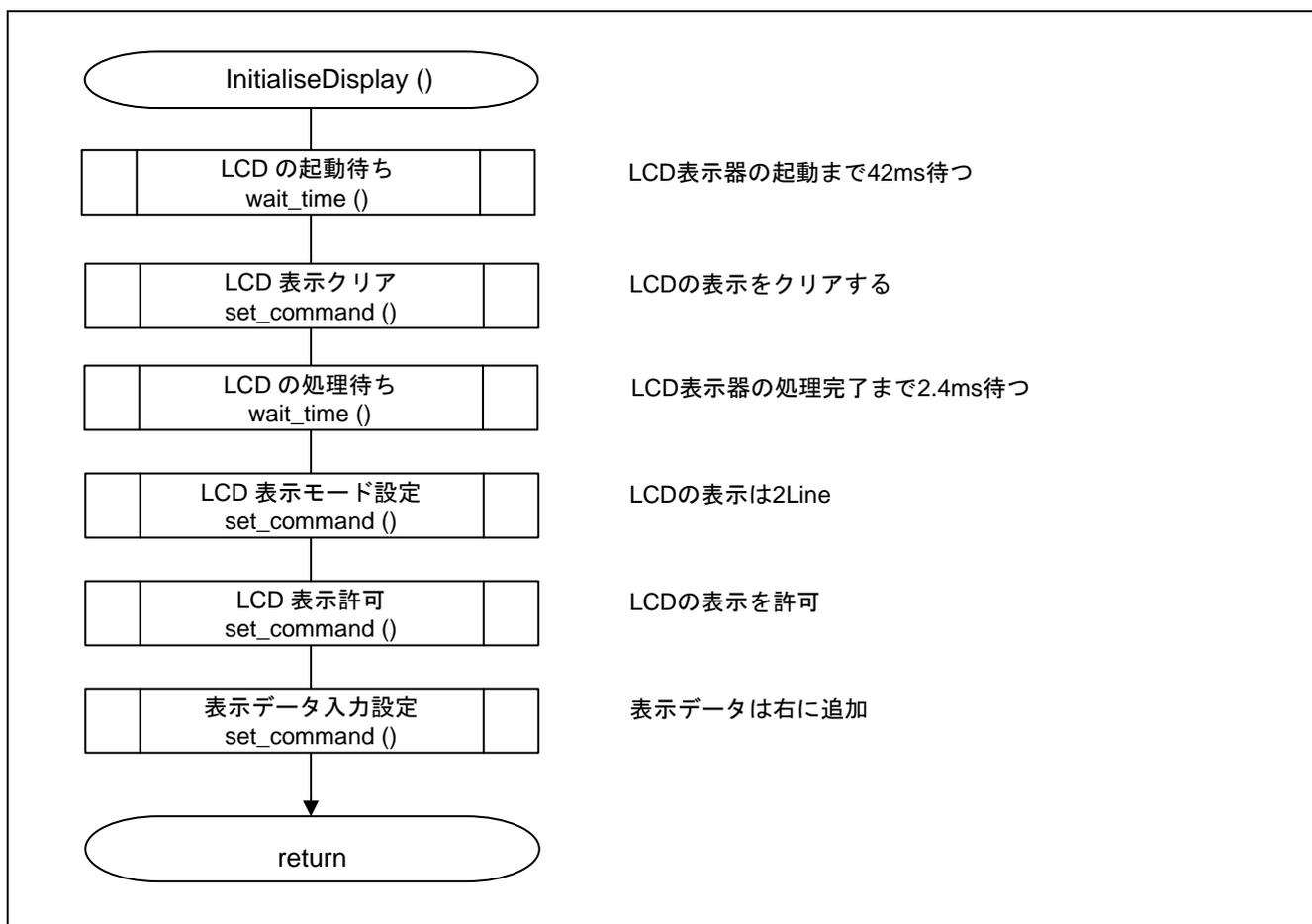


図 5.16 LCD の初期設定処理

5.7.14 LCD コマンド設定処理

図 5.17 に LCD コマンド設定処理のフローチャートを示します。

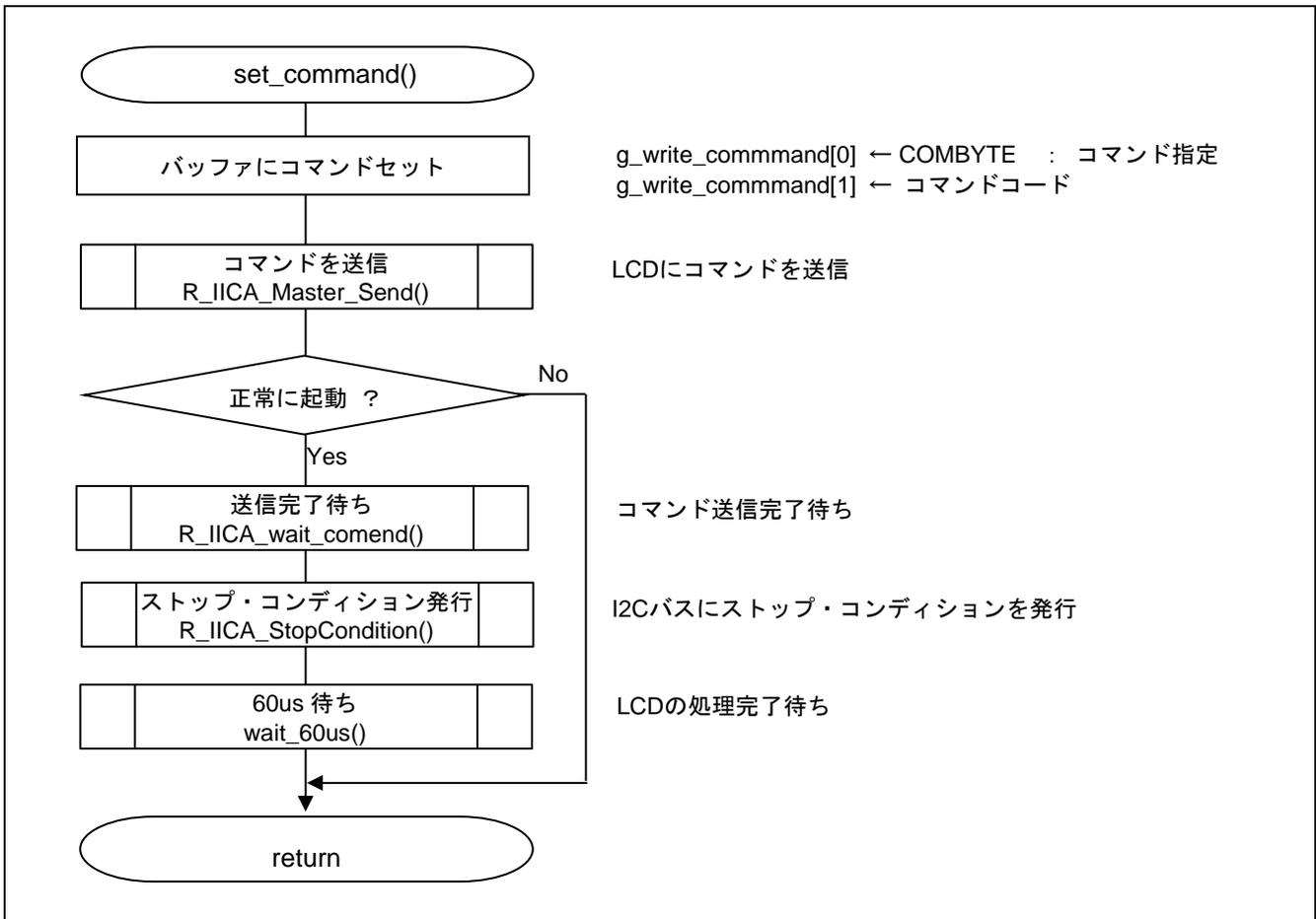


図 5.17 LCD コマンド設定処理

5.7.15 時間待ち処理

図 5.18 に時間待ち処理のフローチャートを示します。

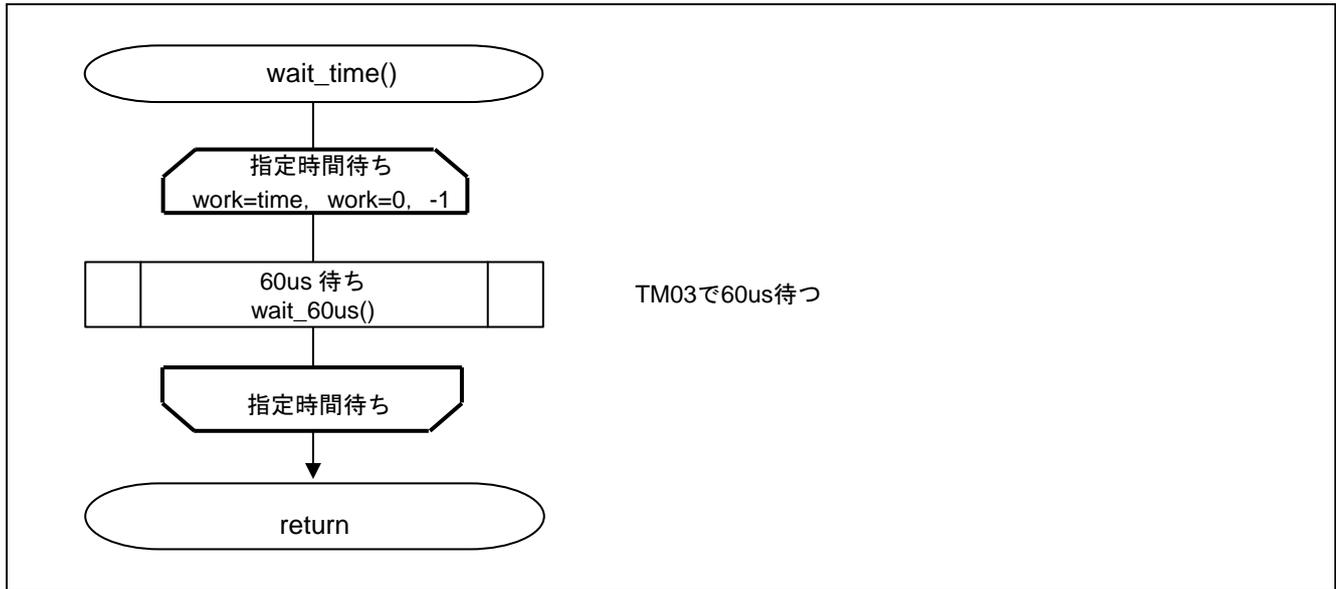


図 5.18 時間待ち処理

5.7.16 60us 待ち処理

図 5.19 に 60us 待ち処理のフローチャートを示します。

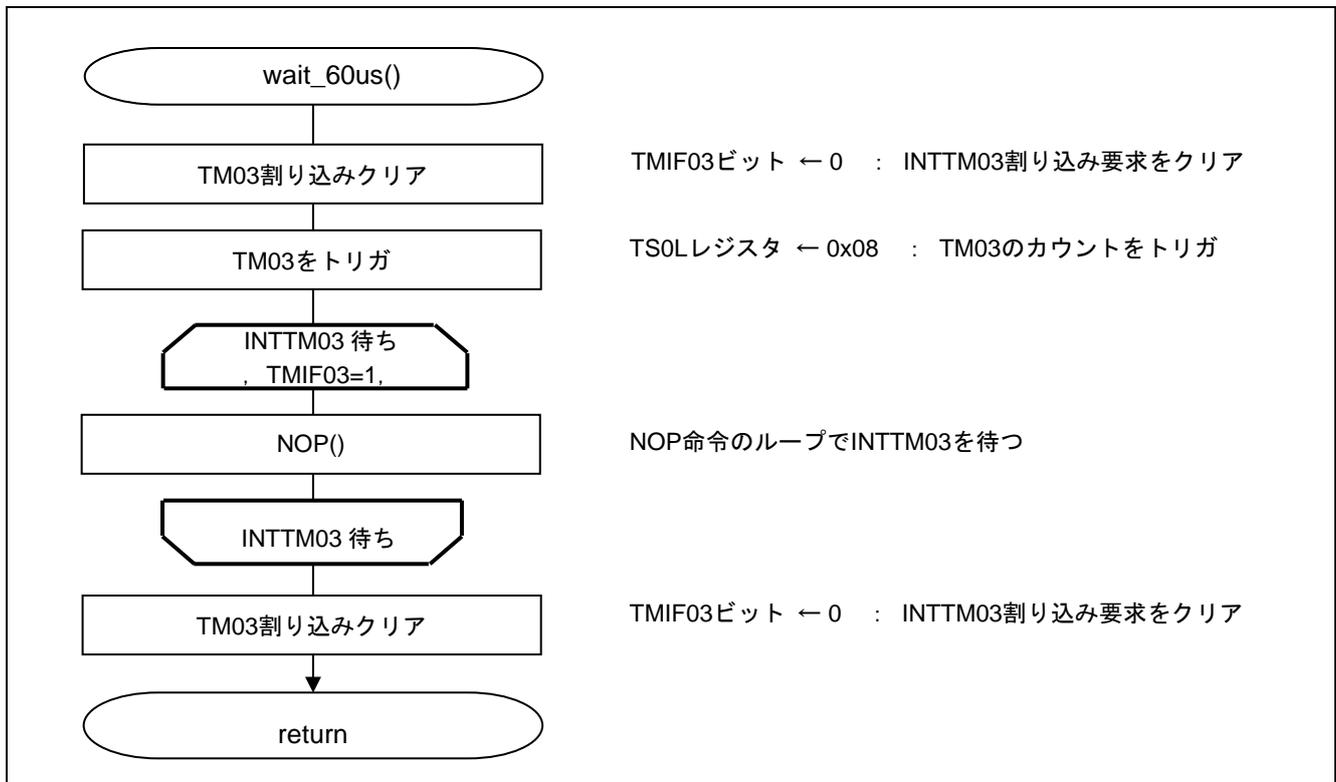


図 5.19 60us 待ち処理

5.7.17 LCD 文字列表示処理

図 5.20 に LCD 文字列表示処理のフローチャートを示します。

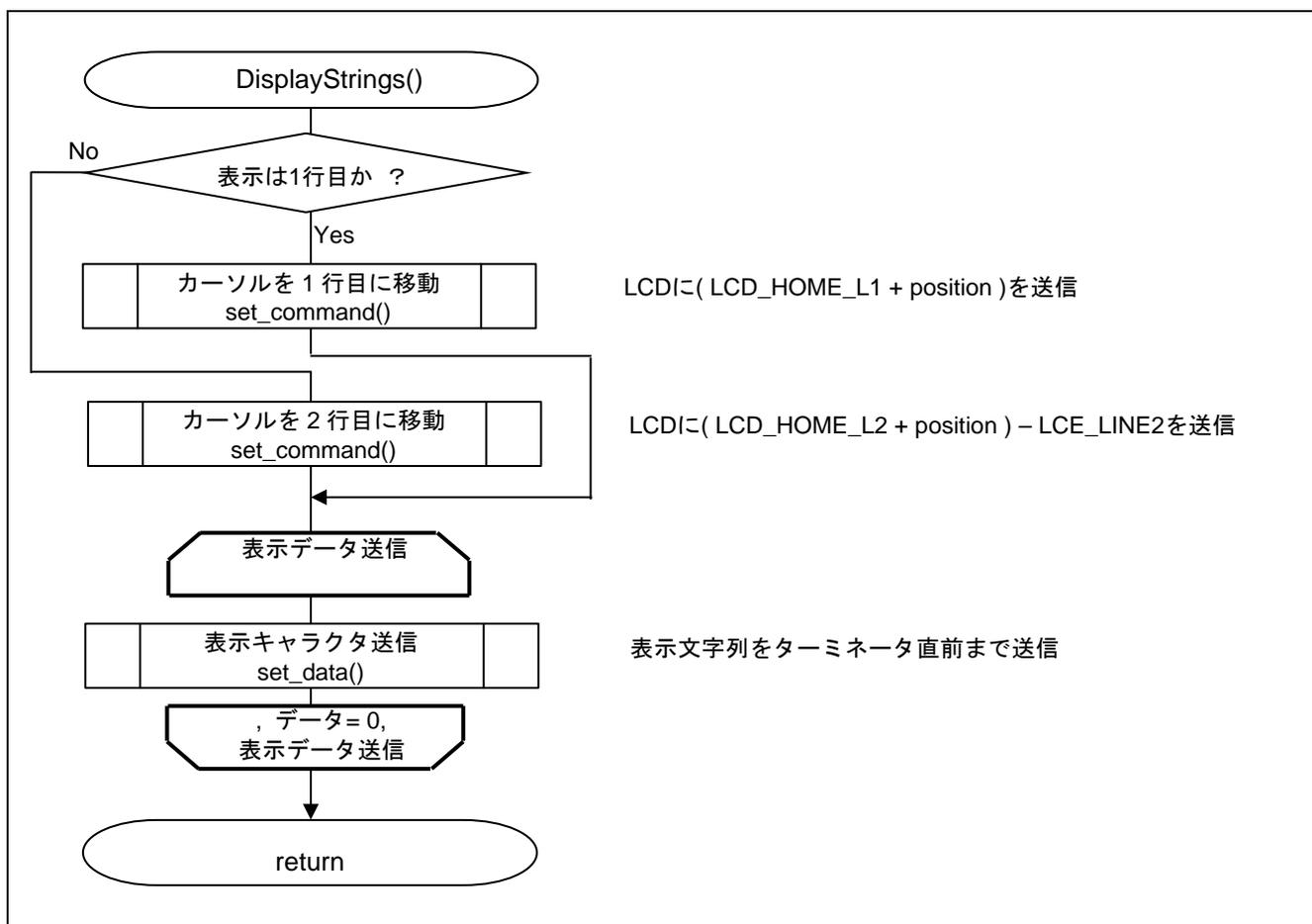


図 5.20 LCD 文字列表示処理

5.7.18 表示データ設定処理

図 5.21 に表示データ設定処理のフローチャートを示します。

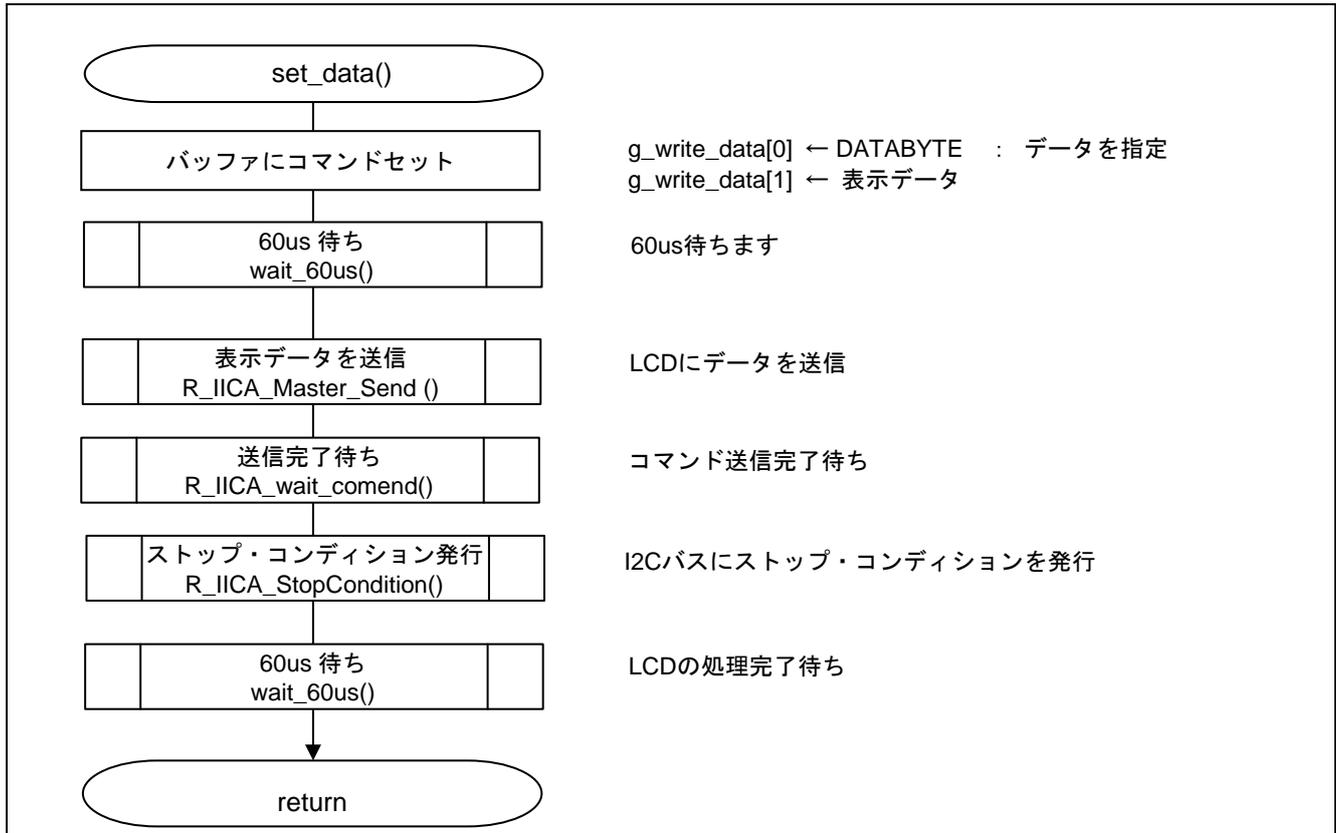


図 5.21 表示データ設定処理

5.7.19 I2C 送信起動処理

図 5.22 に I2C 送信起動処理のフローチャートを示します。

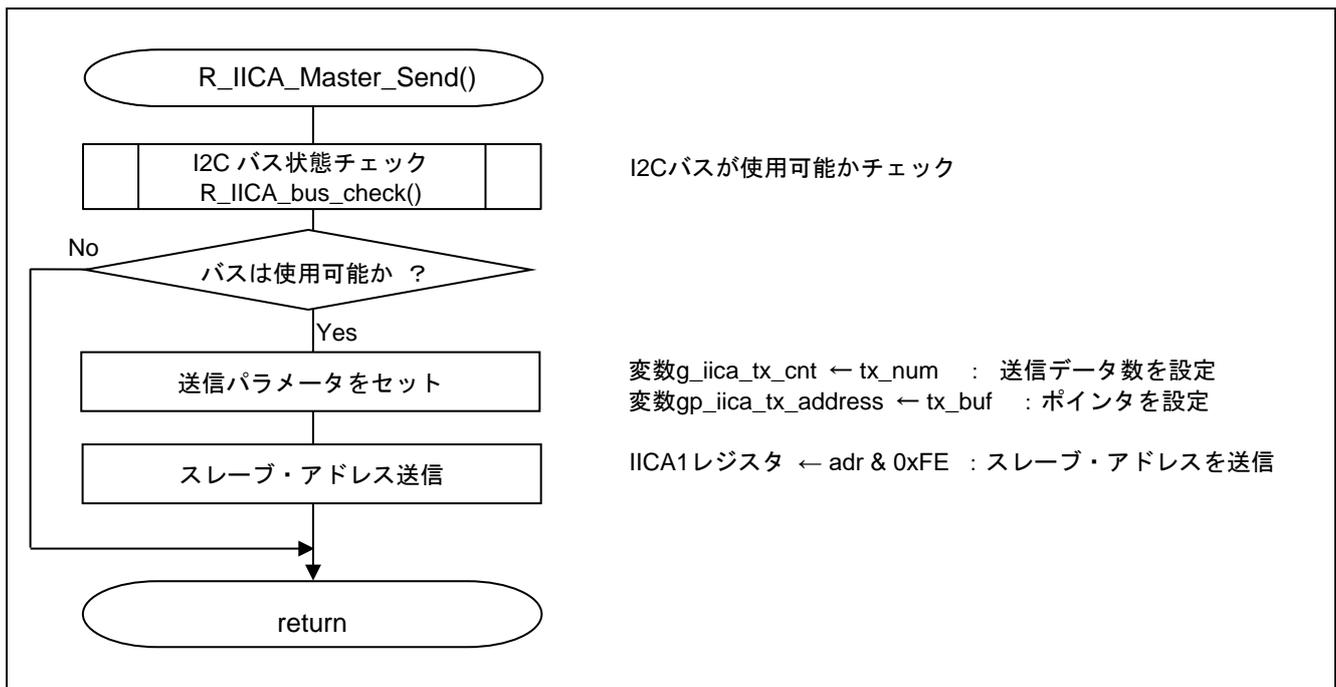


図 5.22 I2C 送信起動処理

5.7.20 I2C 通信完了待ち処理

図 5.23 に I2C 通信完了待ち処理のフローチャートを示します。

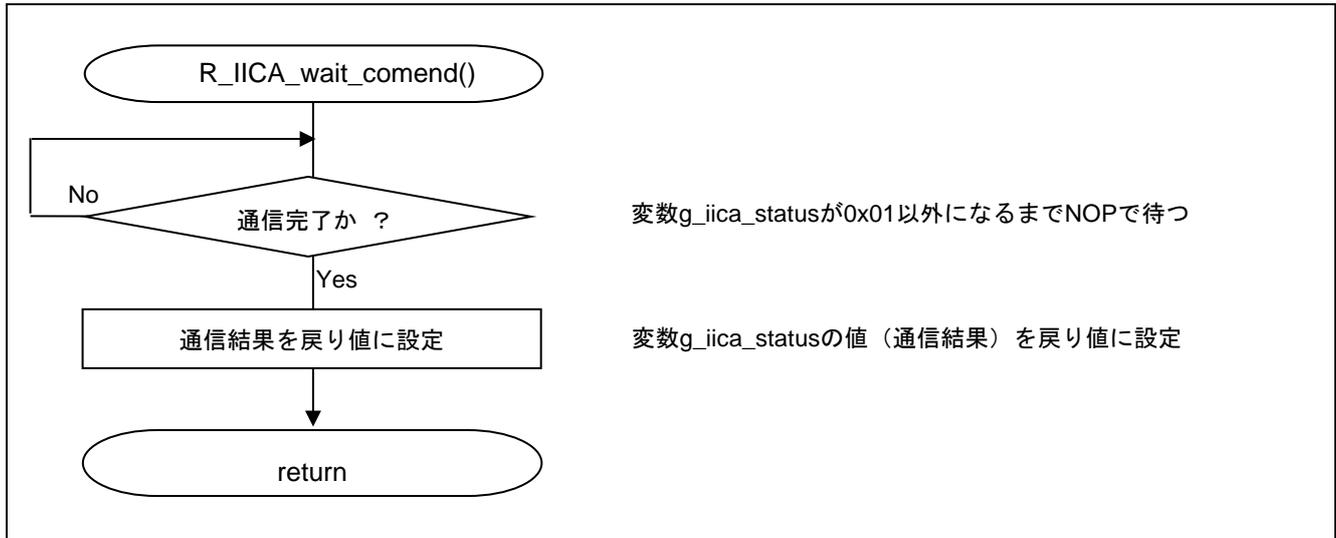


図 5.23 I2C 通信完了待ち処理

5.7.21 ストップ・コンディション発行処理

図 5.24 にストップ・コンディション発行処理のフローチャートを示します。

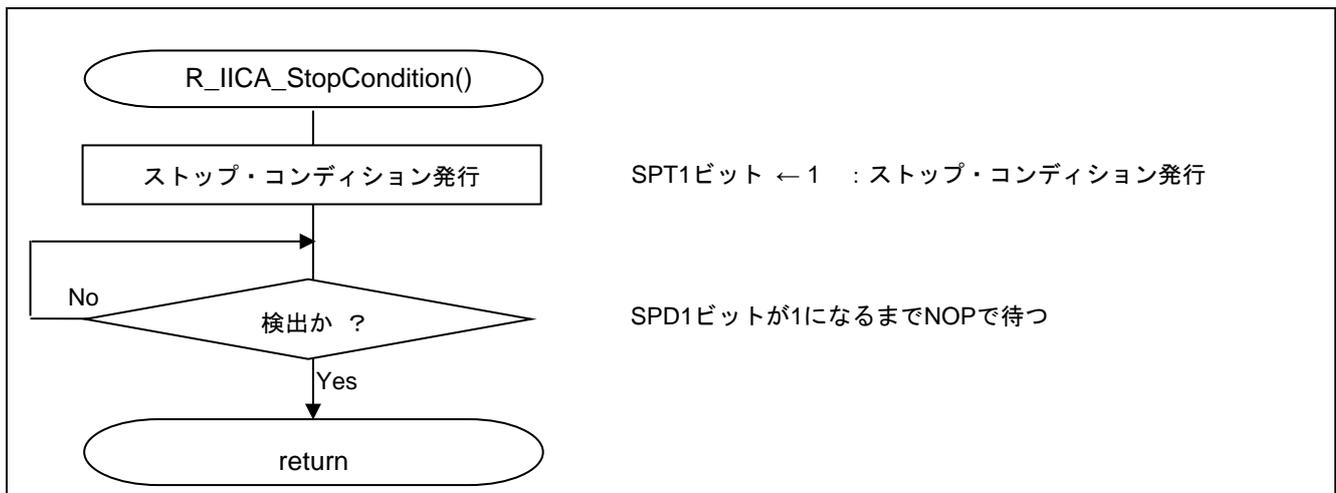


図 5.24 I2C ストップ・コンディション発行処理

5.7.22 I2C バス状態チェック処理

図 5.25 に I2C バス状態チェック処理のフローチャートを示します。

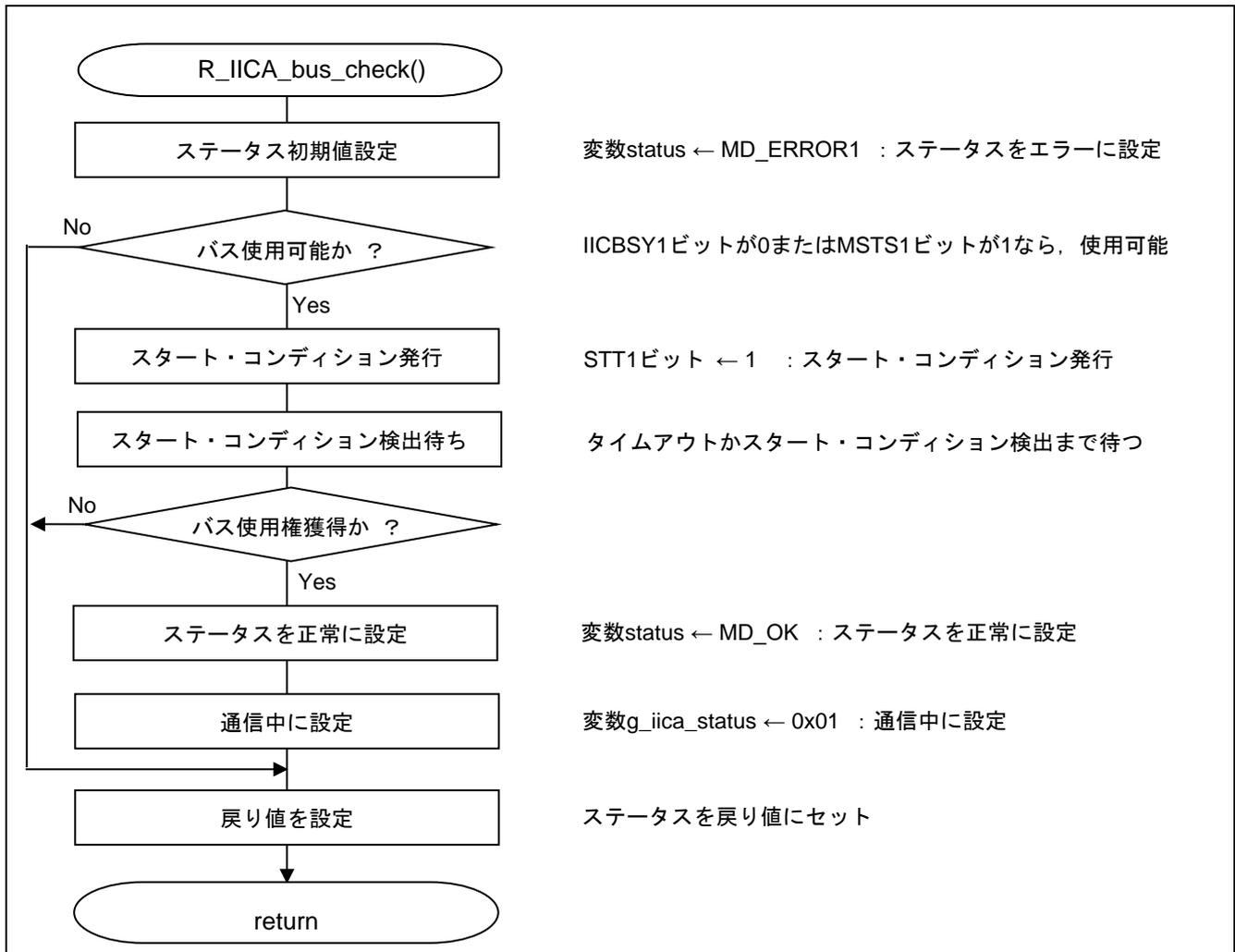


図 5.25 I2C バス状態チェック処理

5.7.23 INTP0 割り込み処理

図 5.26 に INTP0 割り込み処理のフローチャートを示します。



図 5.26 INTP0 割り込み処理

5.7.24 INTIT00 割り込み処理

図 5.27 に INTIT00 割り込み処理のフローチャートを示します。

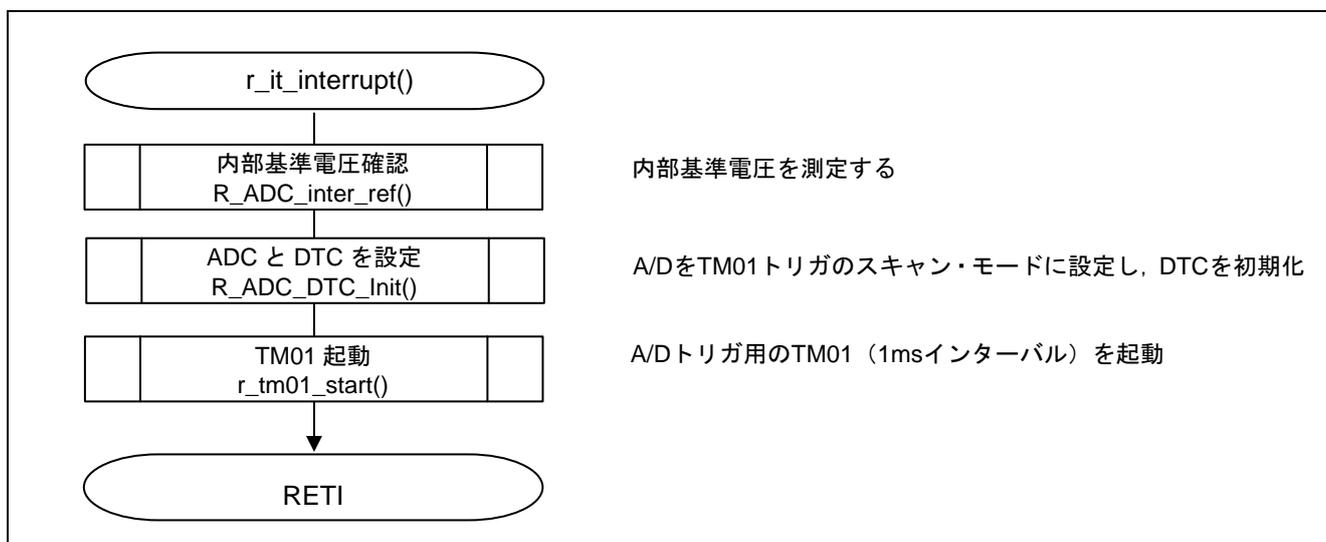


図 5.27 INTIT00 割り込み処理

5.7.25 内部基準電圧確認処理

図 5.28、図 5.29 に内部基準電圧確認処理のフローチャートを示します。

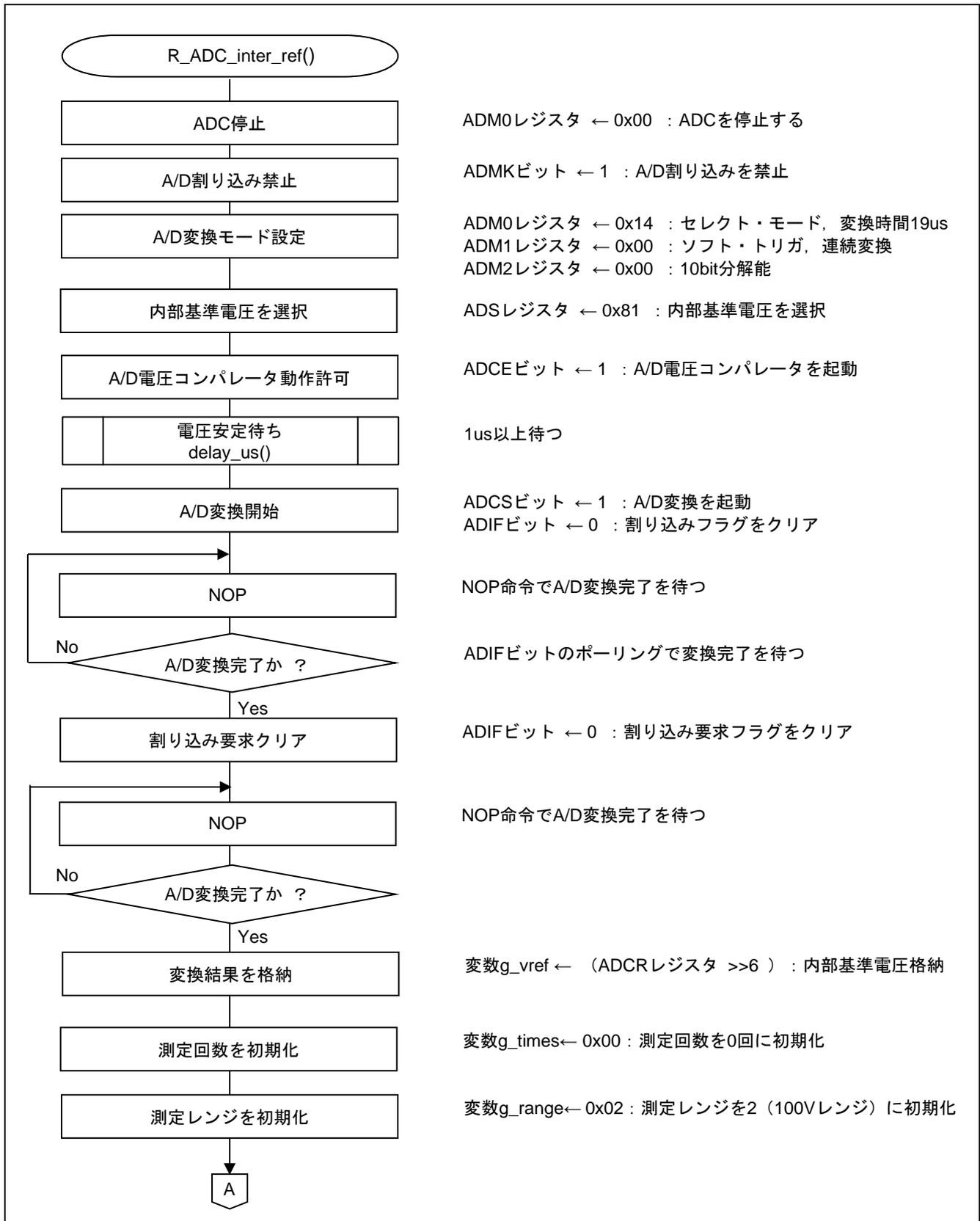


図 5.28 内部基準電圧確認処理 (1/2)

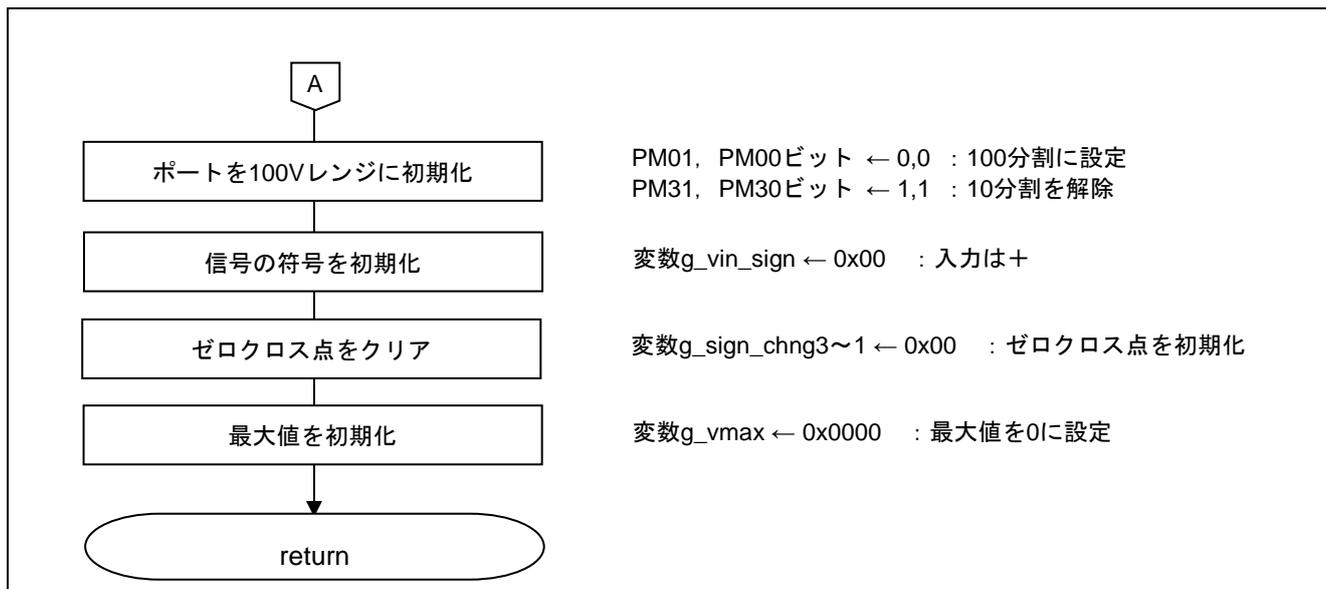


図 5.29 内部基準電圧確認処理 (2/2)

5.7.26 1us 待ち処理

図 5.30 に 1us 待ち処理のフローチャートを示します。

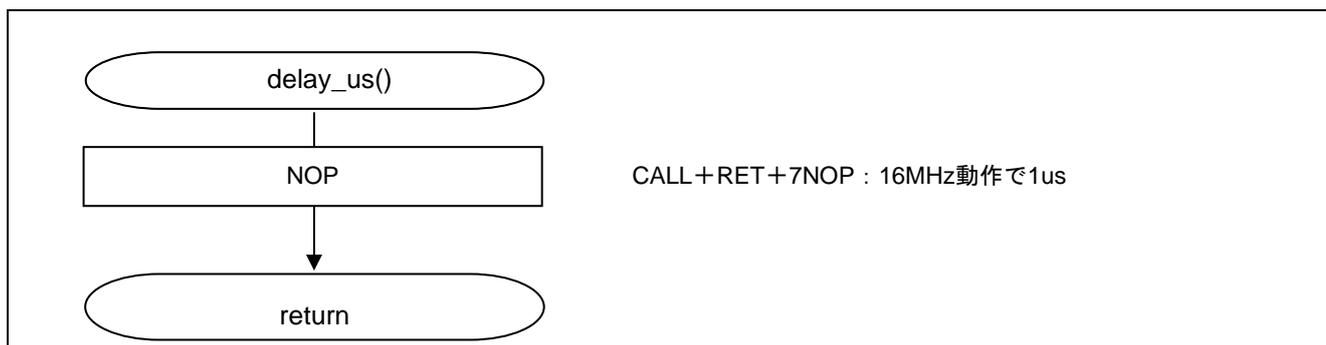


図 5.30 1us 待ち処理

5.7.27 ADC と DTC 設定処理

図 5.31 に ADC と DTC 設定処理のフローチャートを示します。

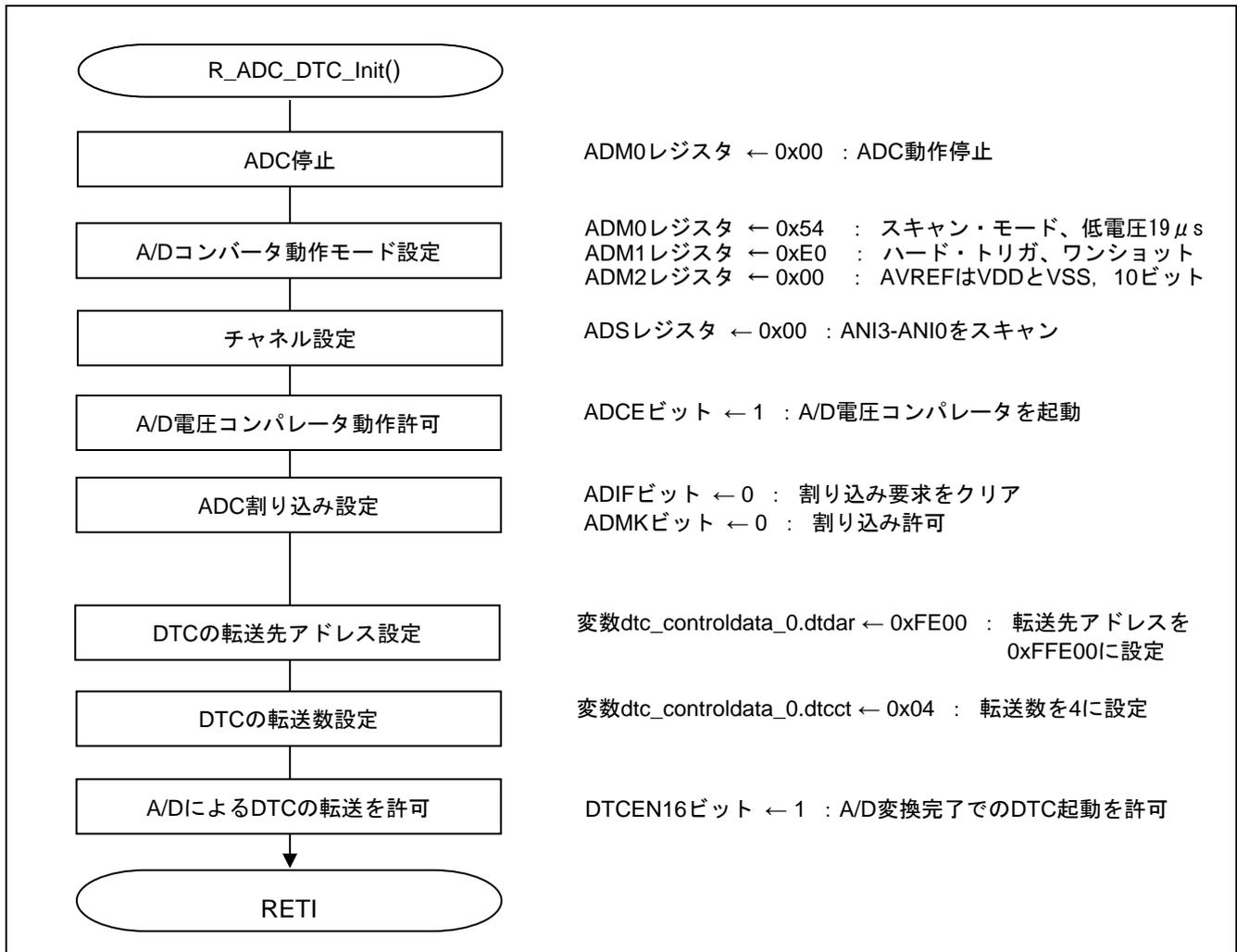


図 5.31 ADC と DTC 設定処理処理

5.7.28 TM01 起動処理

図 5.32 に ADC トリガ用の 1ms インターバルの TM01 起動処理のフローチャートを示します。

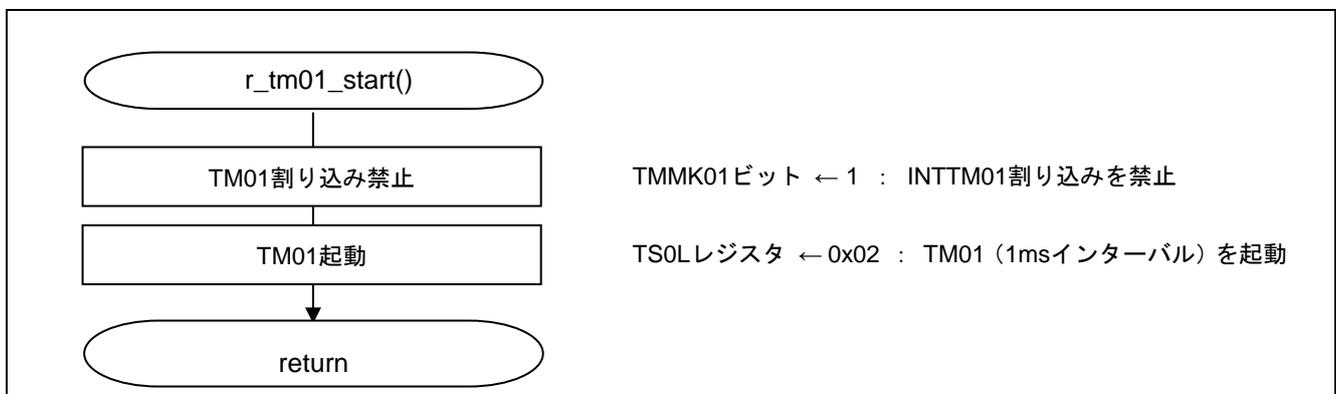


図 5.32 TM01 起動処理

5.7.29 A/D 変換完了割り込み処理

図 5.33、図 5.34 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。

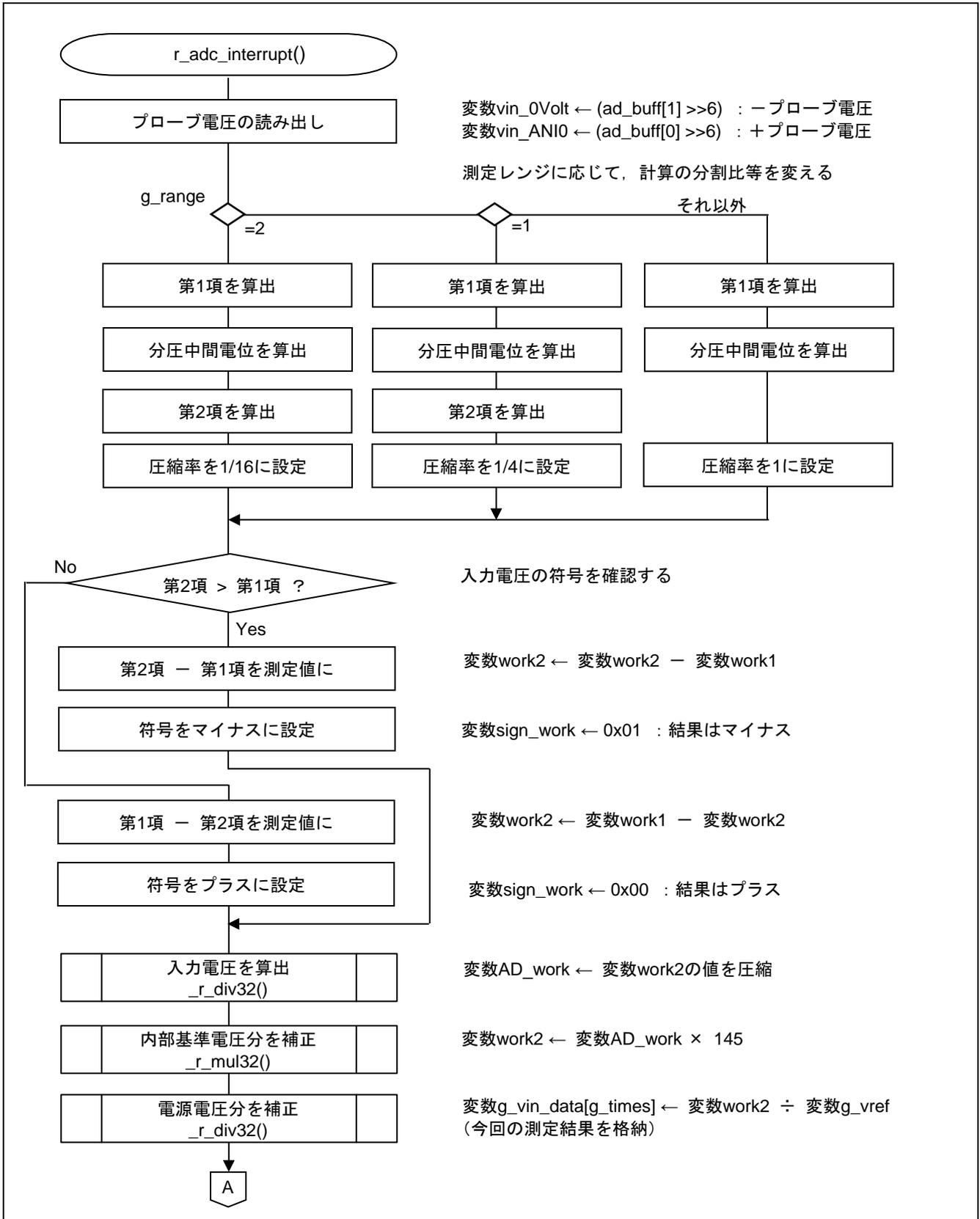


図 5.33 A/D 変換完了割り込み処理 (1/2)

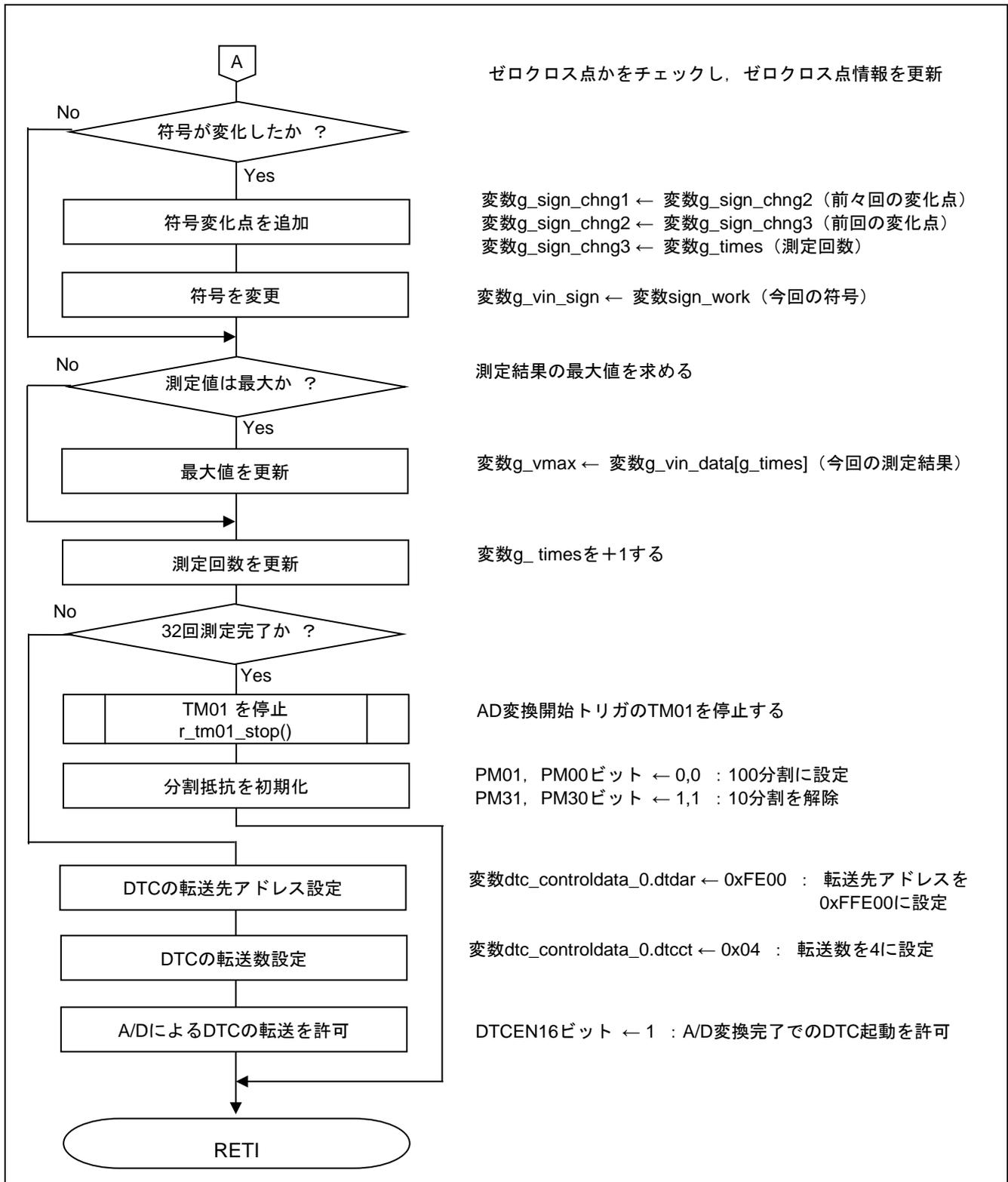


図 5.34 A/D 変換完了割り込み処理 (2/2)

5.7.30 I2C 通信完了割り込み処理

図 5.35、図 5.36 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。

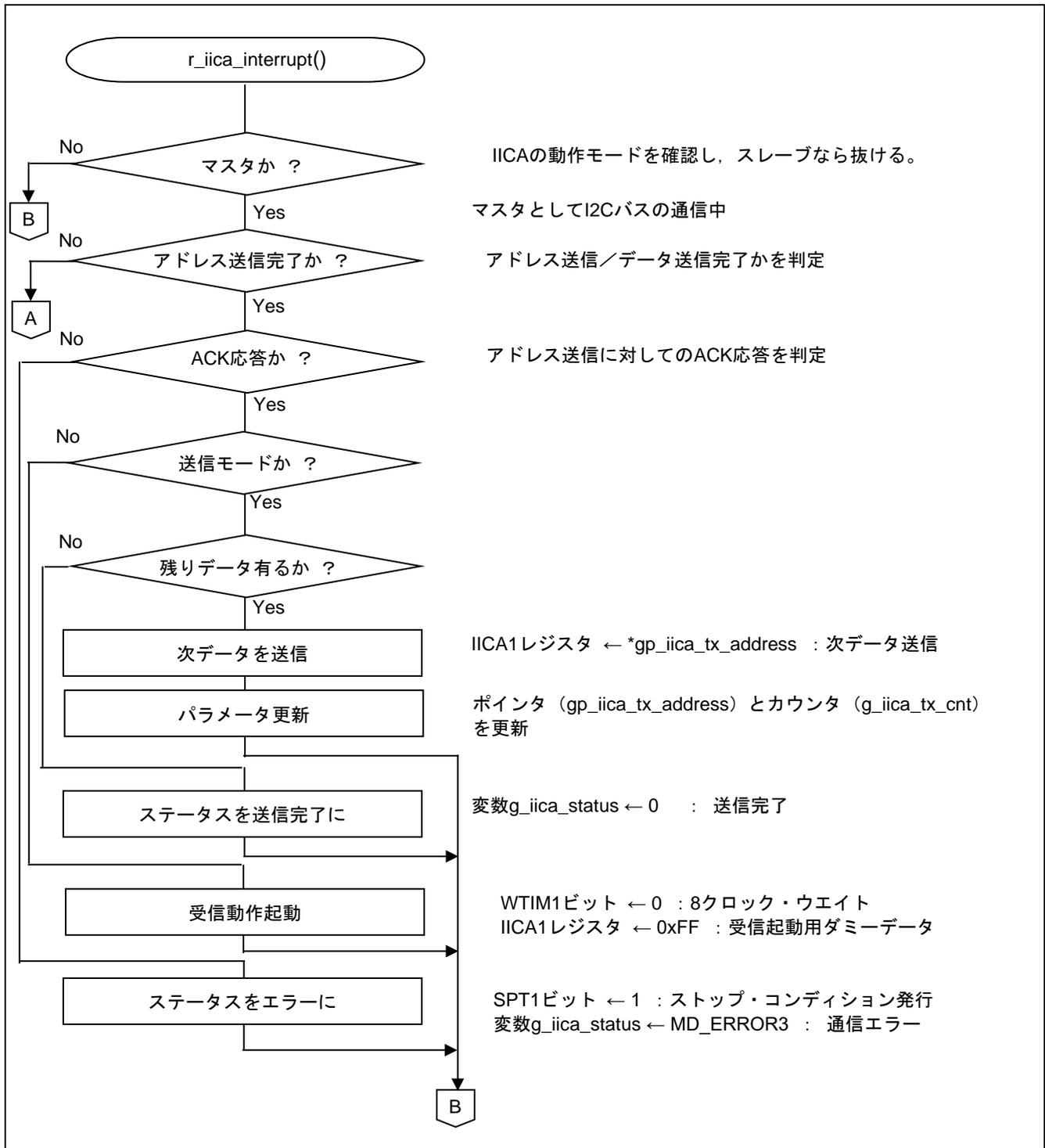


図 5.35 I2C 通信完了割り込み処理 (1/2)

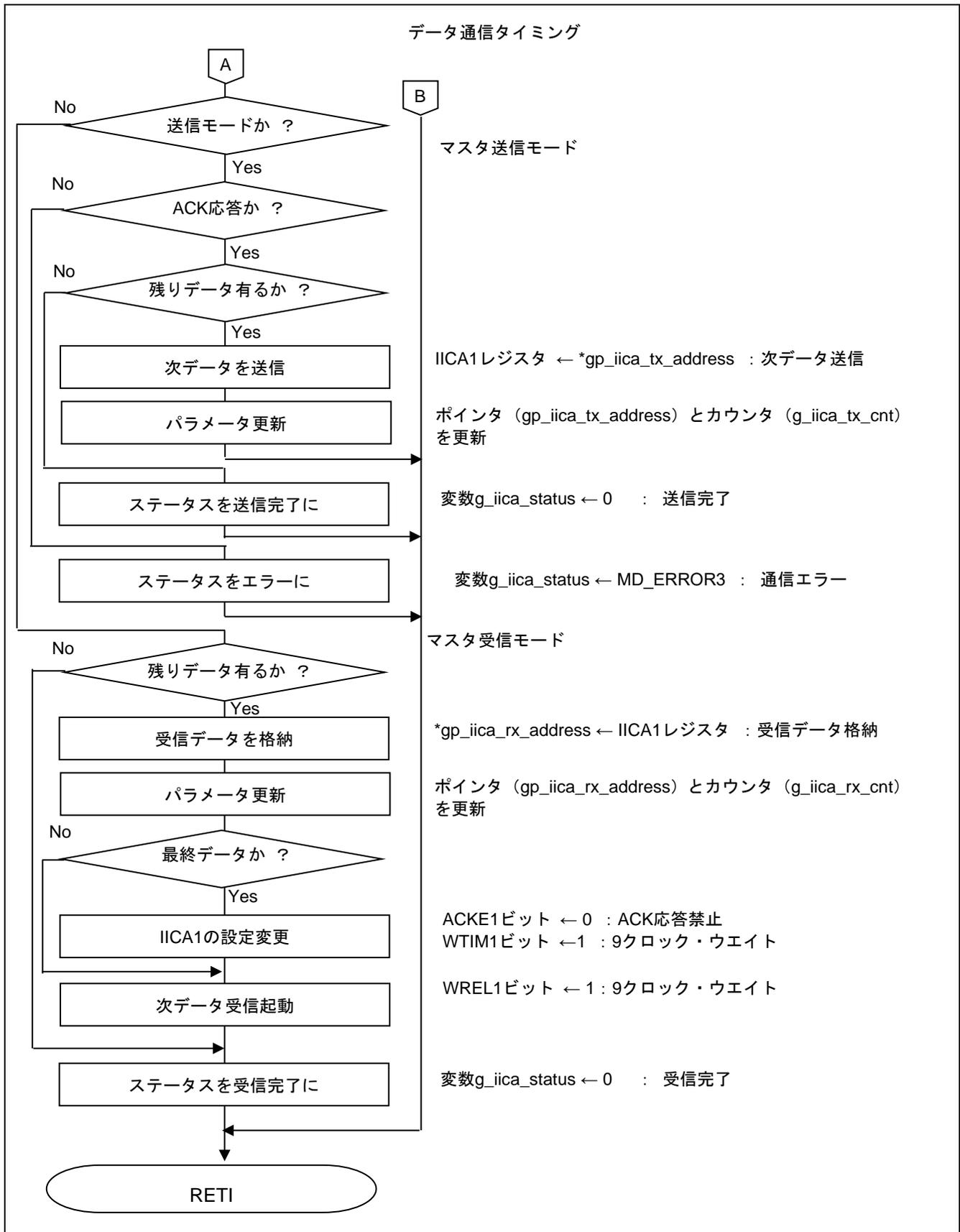


図 5.36 I2C 通信完了割り込み処理 (2/2)

5.7.31 16ビット乗算処理

図 5.37 に 16 ビット乗算処理のフローチャートを示します。

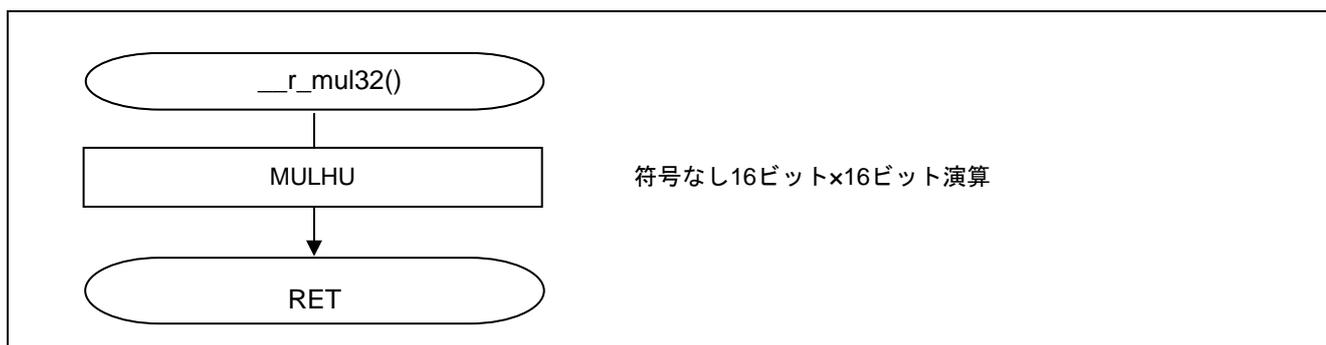


図 5.37 16 ビット乗算処理

5.7.32 32ビット除算処理

図 5.38 に 32 ビット除算処理のフローチャートを示します。

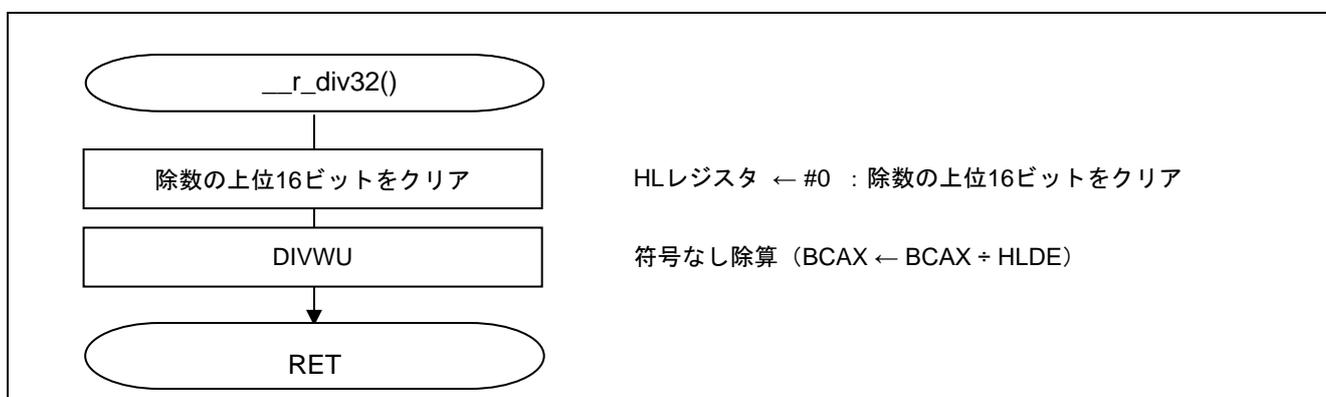


図 5.38 32 ビット除算処理

5.7.33 16ビット除算処理

図 5.39 に 16 ビット除算処理のフローチャートを示します。

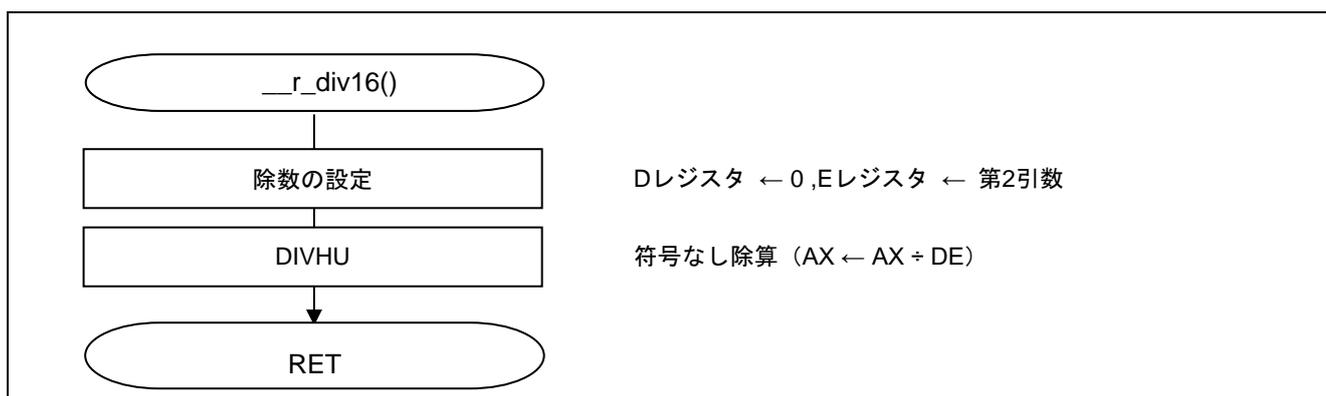


図 5.39 16 ビット除算処理

5.7.34 16ビット剰余処理

図 5.40 に 16 ビット剰余処理のフローチャートを示します。

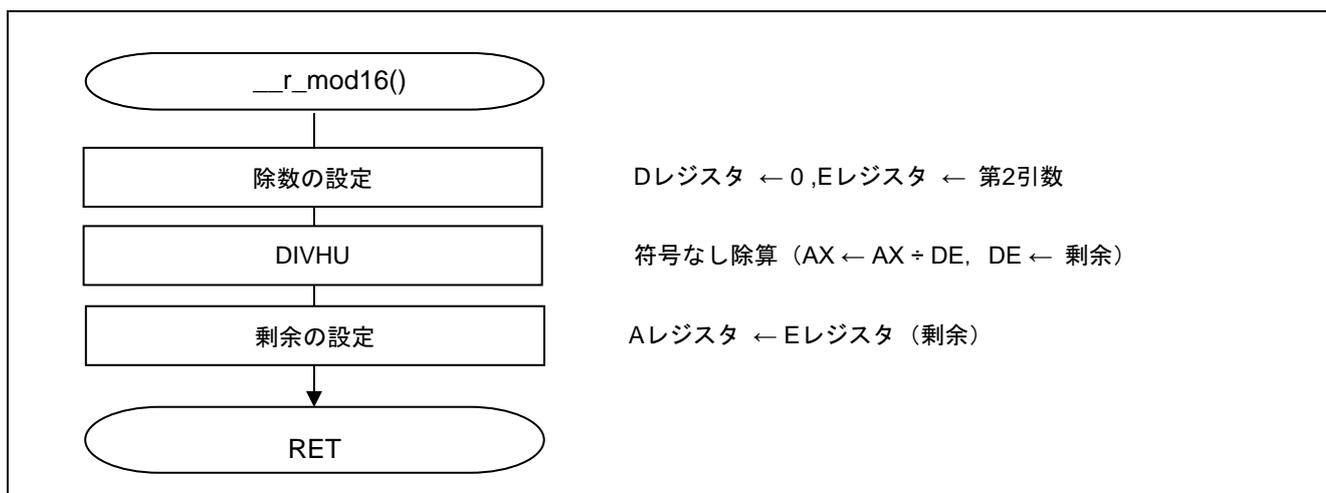


図 5.40 16ビット剰余処理

5.7.35 16ビット自乗平均平方根処理

図 5.41 に 16 ビット自乗平均平方根のフローチャートを示します。

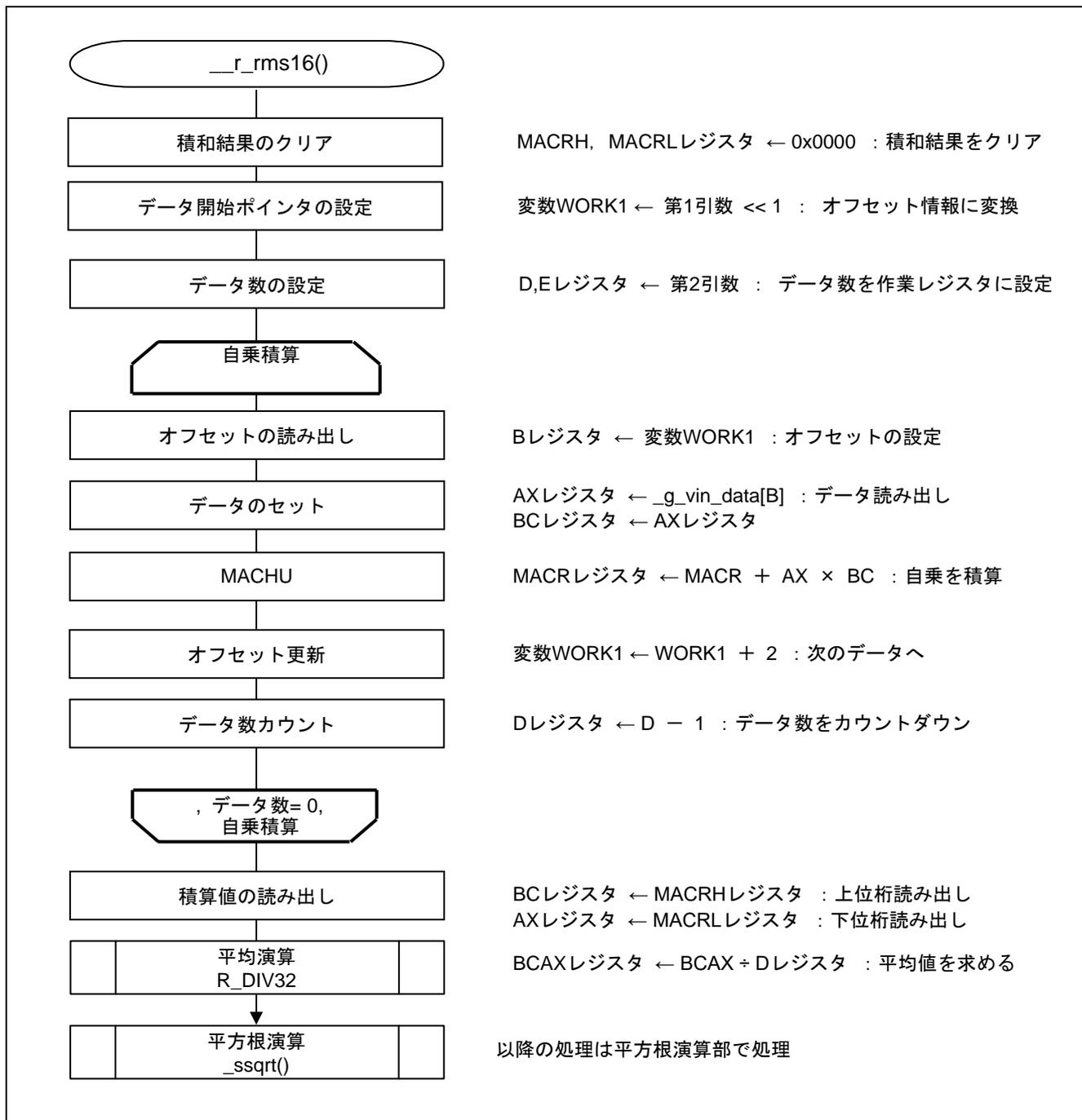


図 5.41 16 ビット自乗平均平方根処理

6. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

7. 参考ドキュメント

RL78/G11 ユーザーズマニュアル ハードウェア編 (R01UH0637J)

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

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/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

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	2018.1.19	—	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うと、内部ROM、レイアウトパターンの相違などにより、電气的特性の範囲で、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事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、お客様にて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/>