

## 要旨

本アプリケーションノートでは、RZ/T1 の各動作モードにおける初期設定を行うサンプルプログラムについて説明します。

初期設定サンプルプログラムの特長を以下に示します。

- 各動作モードに対応した 16 ビットバスブート版、SPI ブート版、および外付けフラッシュを必要とせず内蔵 RAM (TCM) のみ使用する RAM 実行版から構成されます。
- 本サンプルプログラムはローダプログラム部とユーザアプリケーションプログラム部で構成され、ブート起動後、順に RZ/T1 の初期設定を行います。
- ローダプログラムでは、RZ/T1 のブート処理後にクロック発生回路、バスステートコントローラなどの初期化やユーザアプリケーションプログラム自体のコピー処理などを行います。
- ユーザアプリケーションプログラムでは、ポート、ECM 機能、割り込み設定を行い、LED の点滅処理を行います。

## 対象デバイス

RZ/T1 グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

# 目次

1.	仕様	5
2.	動作環境	6
3.	周辺機能説明	7
4.	ハードウェア説明	8
4.1	ハードウェア構成例	8
4.2	使用端子一覧	10
5.	ソフトウェア説明	11
5.1	動作概要	11
5.1.1	プロジェクト設定	12
5.1.2	使用準備	13
5.2	メモリマップ	14
5.2.1	サンプルプログラムのセクション配置	15
5.2.2	MPU の設定	17
5.2.3	例外処理ベクタテーブル	18
5.3	使用割り込み一覧	19
5.4	固定幅整数一覧	19
5.5	定数／エラーコード一覧	20
5.6	構造体／共用体／列挙型一覧	22
5.7	大域変数一覧	23
5.8	関数一覧	24
5.9	API 関数仕様	25
5.9.1	loader_init1	25
5.9.2	loader_init2	25
5.9.3	reset_check	25
5.9.4	cpg_init	26
5.9.5	bsc_init	26
5.9.6	copy_to_atcm	26
5.9.7	cache_init	27
5.9.8	set_low_vec	27
5.9.9	main	27
5.9.10	port_init	27
5.9.11	ecm_init	28
5.9.12	icu_init	28
5.9.13	atcm_waitset	28
5.9.14	cpg_pll_wait	29
5.9.15	R_ECM_Init	29
5.9.16	R_ICU_Enable	29
5.9.17	R_ICU_Disable	30
5.9.18	R_ICU_ExtPinInit	30
5.9.19	R_ICU_Regist	31

5.9.20	Userdef_SFLASH_Set_Mode.....	31
5.9.21	Userdef_SFLASH_Write_Enable .....	32
5.9.22	Userdef_SFLASH_Busy_Wait.....	32
5.9.23	sflash_exmode_init.....	32
5.9.24	sflash_exmode_setting.....	33
5.9.25	sflash_wait_tend.....	33
5.9.26	sflash_set_config.....	34
5.9.27	SPIBSC_Exread_Mode_Config.....	34
5.9.28	Userdef_SPIBSC_Set_Config .....	34
5.9.29	R_IRQ9_isr.....	35
5.9.30	R_IRQ16_isr.....	35
5.10	フローチャート.....	36
5.10.1	ローダプログラム 1 処理.....	36
5.10.2	ローダプログラム 2 処理.....	37
5.10.3	メイン処理.....	38
5.10.4	リセットフラグ判定処理.....	39
5.10.5	クロック設定処理.....	41
5.10.6	バス設定処理.....	42
5.10.7	アプリケーションプログラム転送処理 .....	45
5.10.8	MPU、キャッシュ設定処理.....	45
5.10.9	ロウベクタ設定処理 .....	46
5.10.10	ポート設定処理 .....	46
5.10.11	ECM 設定処理 .....	46
5.10.12	ICU 設定処理 .....	47
5.10.13	ATCM アクセスウエイト設定処理 .....	48
5.10.14	PLL 安定待ち (100 $\mu$ s) 処理.....	49
5.10.15	ECM 初期設定処理 .....	50
5.10.16	割り込み許可設定処理.....	51
5.10.17	割り込み禁止設定処理.....	52
5.10.18	外部割り込み設定処理.....	53
5.10.19	割り込み登録設定処理.....	54
5.10.20	シリアルフラッシュメモリ内レジスタ設定処理 (ユーザ定義).....	55
5.10.21	シリアルフラッシュメモリライト許可設定処理 (ユーザ定義).....	55
5.10.22	シリアルフラッシュメモリレディー待ち処理 (ユーザ定義).....	55
5.10.23	SPIBSC 外部アドレスモード初期設定処理 .....	56
5.10.24	SPIBSC 初期設定処理.....	56
5.10.25	SPIBSC データ転送終了待ち処理.....	57
5.10.26	SPIBSC 外部アドレスリードモード設定処理.....	57
5.10.27	SPIBSC コンフィギュレーション処理.....	58
5.10.28	SPIBSC 外部アドレスリード設定処理 (ユーザ定義).....	59

5.10.29	IRQ9 割り込み (IRQ 端子割り込み 5) 処理.....	60
5.10.30	IRQ16 割り込み (IRQ 端子割り込み 12) 処理.....	61
6.	サンプルプログラム.....	62
7.	参考ドキュメント .....	63
付録 1.	各開発環境における補足内容 .....	64

# 1. 仕様

表 1.1 に使用する周辺機能と用途を、図 1.1 に動作環境を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
クロック発生回路 (CPG)	CPUクロックおよび低速オンチップオシレータで使用
割り込みコントローラ (ICUA)	外部割り込み入力端子 (IRQ5、IRQ12) で使用
バーステートコントローラ (BSC)	CS0、CS1空間にNORフラッシュメモリを接続、および、CS2、CS3空間でSDRAMを接続するために使用
SPIマルチI/Oバスコントローラ (SPIBSC)	SPIマルチI/O空間にシリアルフラッシュメモリを接続するために使用
エラーコントロールモジュール (ECM)	ERROROUT#端子の初期化および拡張疑似エラー 35を使用
リセット	リセットステータスフラグを用いたリセット判定で使用
汎用入出力ポート	LEDの点灯および消灯のための端子制御に使用

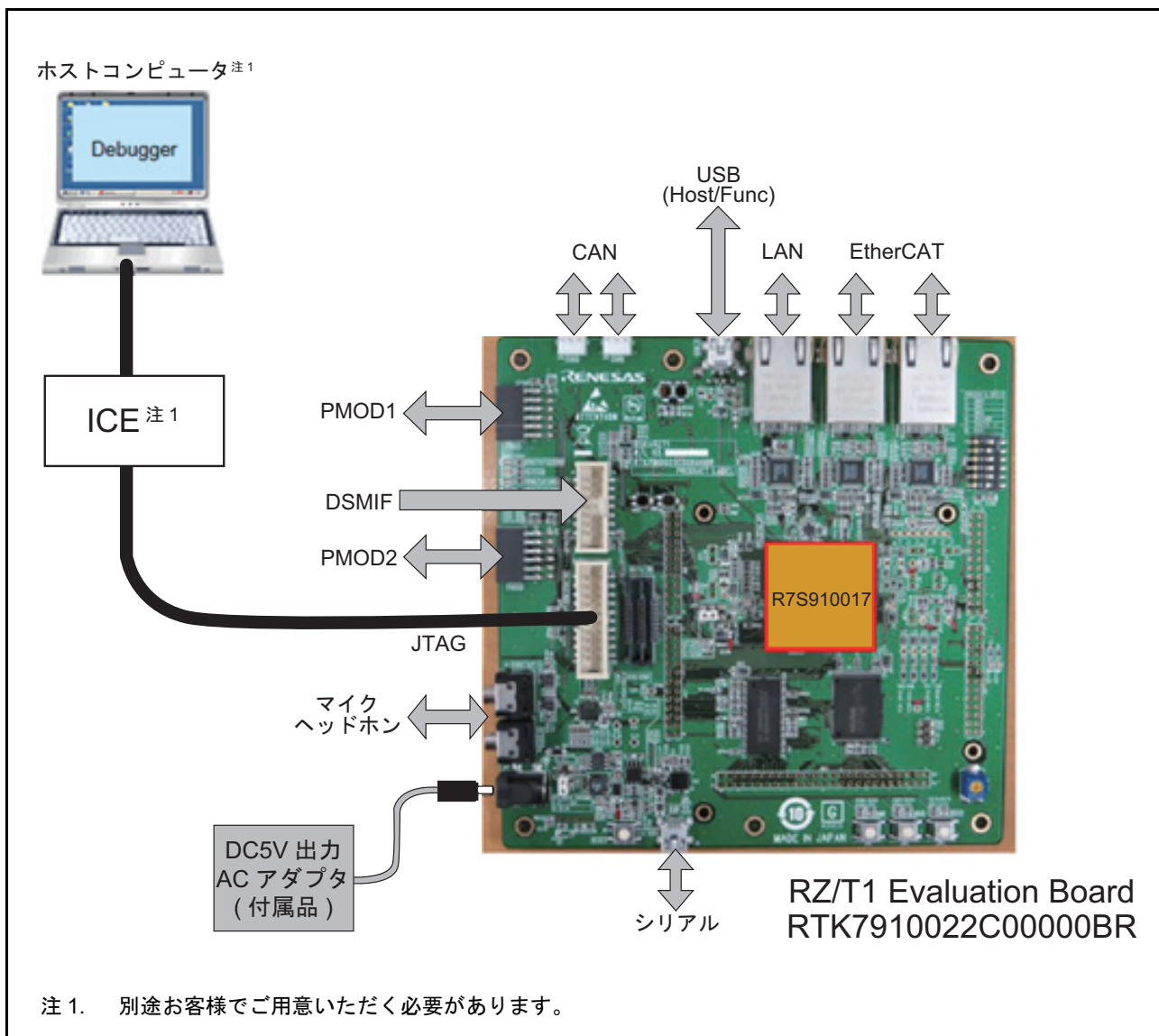


図 1.1 動作環境

## 2. 動作環境

本アプリケーションノートのサンプルプログラムは、下記の環境を想定しています。

表2.1 動作環境

項目	内容
使用マイコン	RZ/T1グループ
動作周波数	CPUCLK = 450MHz
動作電圧	3.3V
統合開発環境	IARシステムズ製 Embedded Workbench® for Arm Version 8.20.2 Arm製 DS-5™ 5.26.2 RENESAS製 e2studio 6.1.0
動作モード	SPIブートモード 16ビットバスブートモード
使用ボード	RZ/T1 Evaluation Board (RTK7910022C00000BR)
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none"><li>NORフラッシュメモリ (CS0、CS1空間に接続) メーカー名 : Macronix International Co..、型名 : MX29GL512FLT2I-10Q</li><li>SDRAM (CS2、CS3空間に接続) メーカー名 : Integrated Silicon Solution Inc、型名 : IS42S16320D-7TL</li><li>シリアルフラッシュメモリ メーカー名 : Macronix International Co..、型名 : MX25L51245G</li></ul>

### 3. 周辺機能説明

動作モード、クロック発生回路 (CPG)、割り込みコントローラ (ICUA)、バスステートコントローラ (BSC)、SPI マルチ I/O コントローラ (SPIBSC)、エラーコントロールモジュール (ECM)、リセット、汎用入出力ポートについての基本的な内容は、RZ/T1 グループ・ユーザズマニュアルハードウェア編を参照してください。

## 4. ハードウェア説明

### 4.1 ハードウェア構成例

図 4.1 に 16 ビットバスブートモード時のハードウェア構成例を、図 4.2 に SPI ブートモード時のハードウェア構成例を示します。

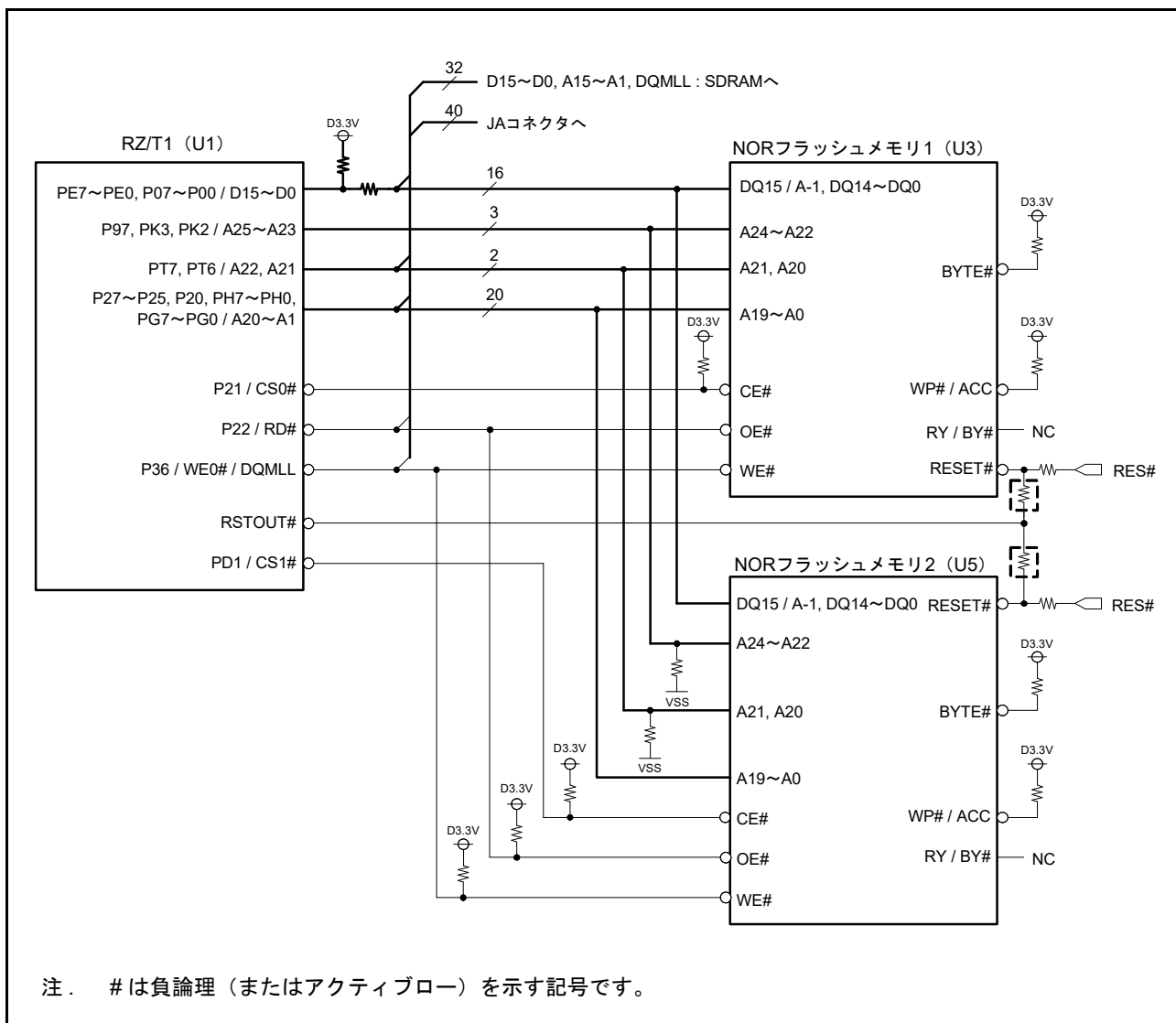


図 4.1 16 ビットバスブートモード時のハードウェア構成例



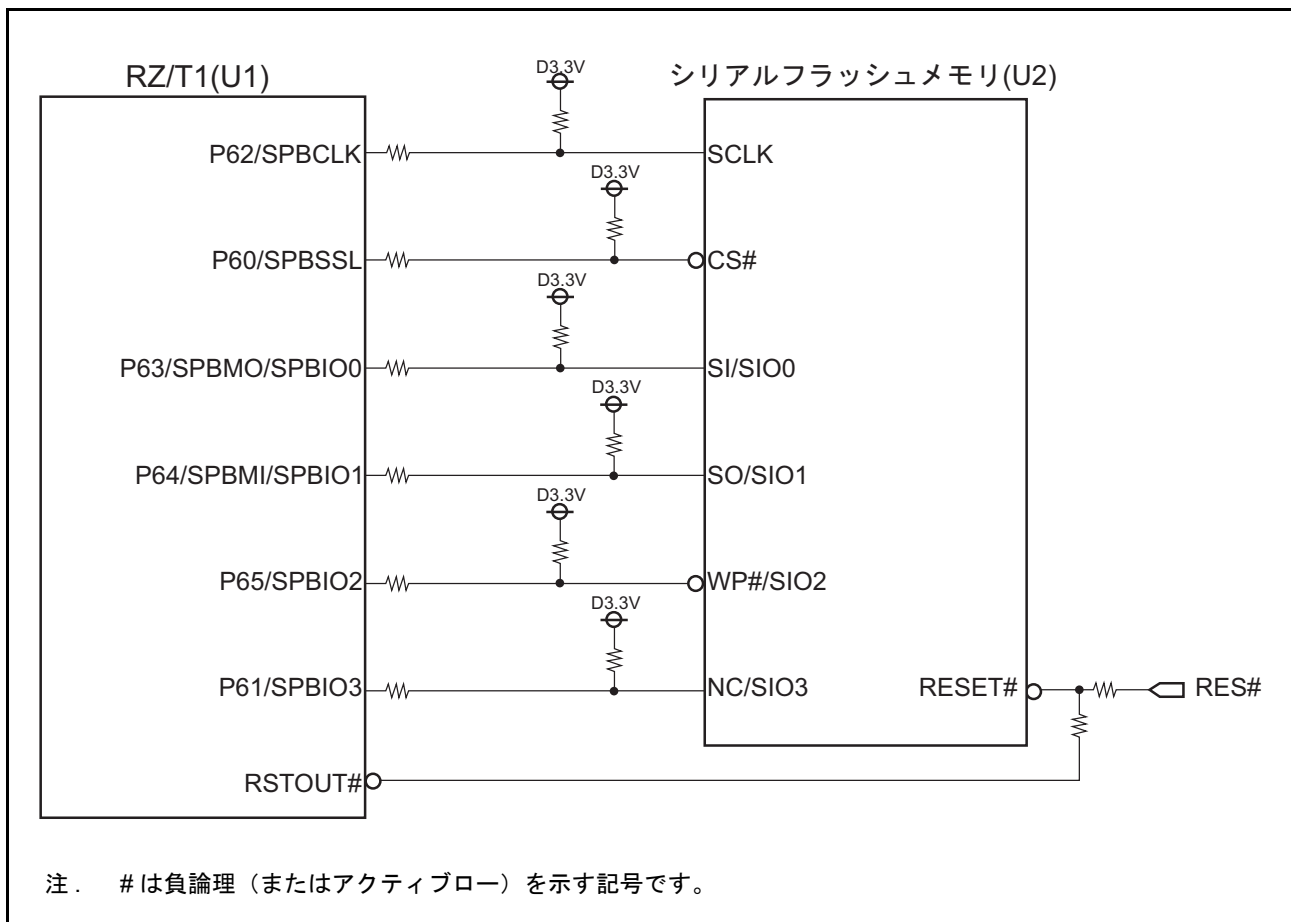


図 4.2 SPI ブートモード時のハードウェア構成例

## 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
A0～A20	出力	NORフラッシュメモリ、SDRAMへのアドレス信号出力
D0～D15	入出力	NORフラッシュメモリ、SDRAMのデータ信号入出力
CS0#	出力	CS0空間に接続されたNORフラッシュメモリへのデバイス選択信号出力
CS1#	出力	CS1空間に接続されたNORフラッシュメモリへのデバイス選択信号出力
CS2#	出力	CS2空間に接続されたSDRAMへのデバイス選択信号出力
CS3#	出力	CS3空間に接続されたSDRAMへのデバイス選択信号出力
RAS#	出力	SDRAMへのRAS#制御信号出力
CAS#	出力	SDRAMへのCAS#制御信号出力
RD/WR#	出力	SDRAMへのリード制御信号またはライト制御信号出力
CKE	出力	SDRAMへのクロックイネーブル制御信号出力
RD#	出力	リード中を示すストロープ信号出力
WE0#/DQMLL	出力	D15～D8に対するライトストロープ信号出力
WE1#/DQMLU	出力	D7～D0に対するライトストロープ信号出力
SPBSSL	出力	スレーブセレクト
SPBCLK	出力	クロック出力
SPBMO/SPBIO0	入出力	マスタ送出データ/データ0
SPBMI/SPBIO1	入出力	マスタ入力データ/データ1
SPBIO2	入出力	データ2
SPBIO3	入出力	データ3
MD0	入力	動作モードの選択 MD0 = "L"、MD1 = "L"、MD2 = "L" (SPIブートモード) MD0 = "L"、MD1 = "H"、MD2 = "L" (16ビットバスブートモード)
MD1	入力	
MD2	入力	
IRQ5	入力	SW2 (IRQ端子割り込み)
IRQ12	入力	SW3 (IRQ端子割り込み)
PF7	出力	LED0の点灯および消灯
P56	出力	LED1の点灯および消灯
P77	出力	LED2の点灯および消灯
P53	出力	LED5の点灯および消灯

注. #は負論理（またはアクティブロー）を示す記号です。

## 5. ソフトウェア説明

以降、特に明記しない場合はEWARM（IAR社製）を使用した場合について説明を行います。

### 5.1 動作概要

RZ/T1はリセット解除後のブート処理で、各動作モード（16ビットバスブート／SPIブート）に対応する外付けフラッシュ（NOR／シリアル）に格納されたローダプログラムを内蔵RAM（BTCM）へコピーします。

ローダプログラムでは、ブート処理後にリセット判定、クロック設定、バス設定などを行った後、外付けフラッシュ（NOR／シリアル）に格納されたユーザアプリケーションプログラムを内蔵RAM（ATCM）にコピーします。次にMPU設定、キャッシュ設定を行い、例外ベクタをロウベクタ状態に切り替え、ユーザアプリケーションプログラムの先頭に分岐します。

ユーザアプリケーションプログラムでは、ポート設定、ECM機能設定、外部端子の割り込み設定を行い、LED0の点滅処理を行います。

SW2が押下されると外部端子割り込み5が発生し、LED1が約5秒の間点灯します。また本処理中にSW3を押下することで多重割り込みが可能です。

SW3を押下すると外部端子割り込み12が発生し、LED1が点灯状態ならばSW3の押下ごとにLED5が点灯／消灯を繰り返します。LED1が消灯状態のときは、ECMリセットが発生します。リセット解除後はリセット判定でECMリセット検出をしてLED2を点灯させます。

図5.1にブート処理後の概要動作を示します。

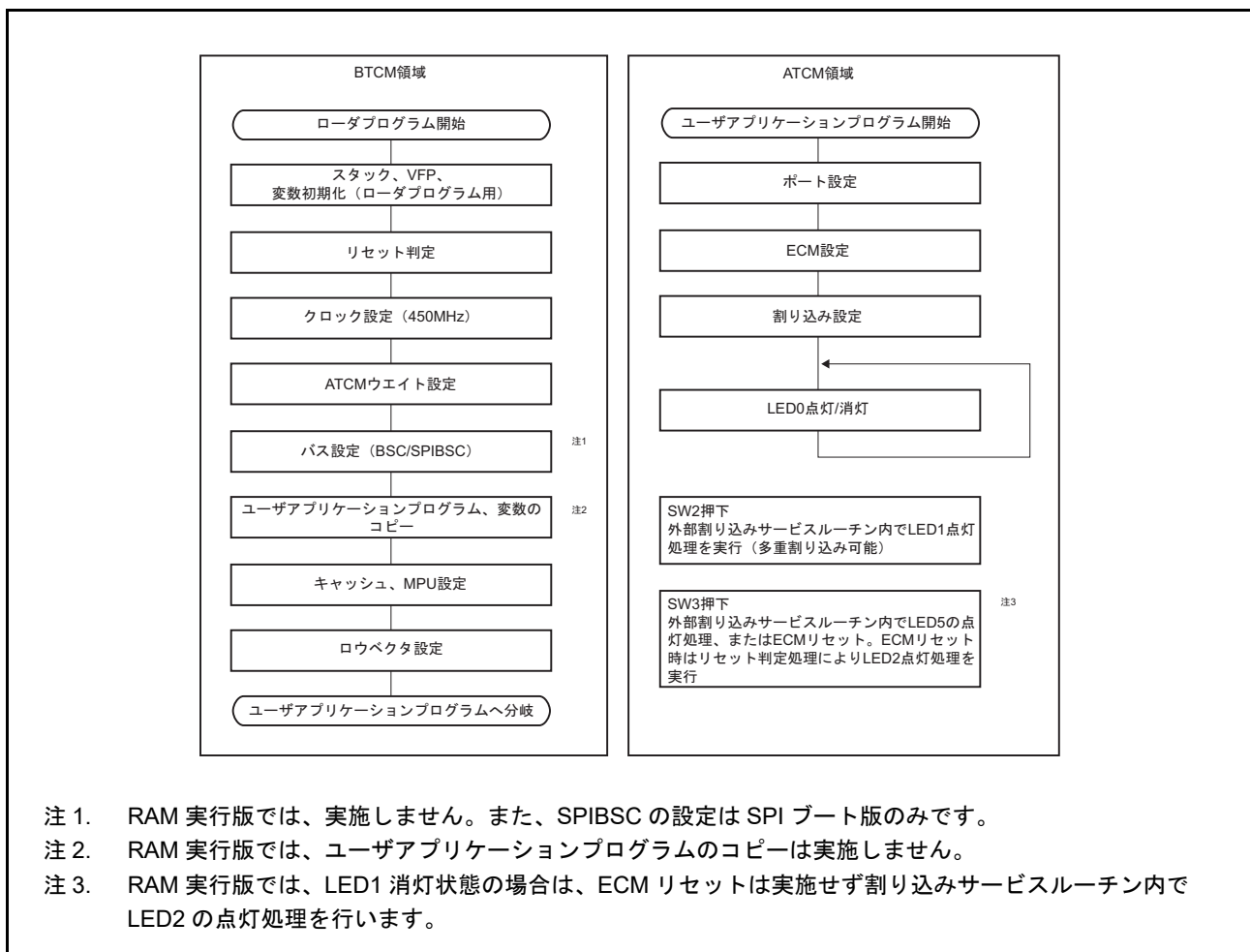


図 5.1 ブート処理後の概要動作



表5.1 サンプルプログラムのプロジェクト設定 (2 / 2)

デバッグ 注1		
設定	ドライバ	I-jet/JTAGjet
	セットアップマクロ 注2	マクロファイルの使用：チェックする 16ビットバスブート版/SPIブート版： \$PROJ_DIR\$\RZT1_init_boot.mac  RAM実行版： \$PROJ_DIR\$\RZT1_init_RAM.mac
ダウンロード	プログラムにアタッチする	チェックしない
	ベリファイする	チェックする
	ダウンロードしない	チェックしない
	フラッシュローダを使用する	チェックする デフォルトのboardファイルのオーバライド：チェックする \$TOOLKIT_DIR\$\config\flashloader\Renesas\FlashRZT1.board

注1. 本サンプルプログラムではデバッグのデフォルト設定ドライバとしてI-jet/JTAGjetを選択しています。使用するドライバに合わせて設定を変更してください。

注2. デバッグ接続時やフラッシュローダ接続時に実行されるRZ/T1用の初期化マクロです。

RZT1\_init\_boot.mac：ダウンロードしてデバッグを開始する時に、フラッシュローダを格納するTCMの初期化や、各動作モードに対応したバス設定（BSC、SPIBSC）などの初期化をします。

RZT1\_init\_RAM.mac：RAM実行版ではRZ/T1のブート処理を行わずにEARMから直接TCM上にプログラムをダウンロードします。このため、TCMの初期化やブート処理で設定されるノンマスク割り込み（FIQ）の有効設定、VICによるIRQ例外ベクタアドレス供給設定などを行います。

マクロの詳細についてはIAR社C-SPYデバッグガイドの「C-SPYマクロ」を参照してください。RZ/T1のブート処理の詳細についてはユーザーズマニュアル ハードウェア編の「動作モード」を参照してください。

### 5.1.2 使用準備

使用するプロジェクトによって、RZ/T1 Evaluation Board（RTK7910022C00000BR）上にあるSW4の設定が異なります。表5.2にSW4の設定を示します。SW4の各設定につきましては、RZ/T1 Evaluation Board RTK7910022C00000BR ユーザーズマニュアルに記載しています。詳細は、「7. 参考ドキュメント」を参照してください。

表5.2 SW4の設定

サンプルプログラム	SW4-1	SW4-2	SW4-3	SW4-4	SW4-5	SW4-6
16ビットバスブートモード版	ON	OFF	ON	ON	ON	OFF
SPIブートモード版	ON	ON	ON	ON	ON	OFF
RAM実行版	上記いずれかのSW4設定					

また各開発環境におけるサンプルプログラムの使用手順については、付録1. 各開発環境における補足内容を参照してください。

## 5.2 メモリマップ

図 5.2 に RZ/T1 グループのアドレス空間と RZ/T1 Evaluation Board のメモリマップを示します。

	RZ/T1 グループの アドレス空間	RZ/T1 Evaluation Board 上の メモリマップ
0000 0000h	ATCM (512KB)	ATCM (512KB)
0008 0000h	予約領域	予約領域
0080 0000h	BTCM (32KB)	BTCM (32KB)
0080 8000h	予約領域	予約領域
0400 0000h	拡張内蔵 SRAM (512KB) I ※対応品のみ	拡張内蔵 SRAM (512KB) I ※対応品のみ
0400 8000h	予約領域	予約領域
0800 0000h	Buffer RAM (128MB)	Buffer RAM (128MB)
1000 0000h	SPI マルチ I/O バス空間 (64MB)	シリアル・フラッシュメモリ (64MB) <sup>注2</sup>
1400 0000h	予約領域	予約領域
2000 0000h	拡張内蔵 SRAM (512KB) D ※対応品のみ	拡張内蔵 SRAM (512KB) D
2008 0000h	予約領域	予約領域
2200 0000h	拡張内蔵 SRAM (512KB) D ※対応品のみ	拡張内蔵 SRAM (512KB) D <sup>注1</sup>
2208 0000h	予約領域	予約領域 <sup>注1</sup>
2400 0000h	拡張内蔵 SRAM (512KB) D ※対応品のみ	拡張内蔵 SRAM (512KB) D <sup>注1</sup>
2408 0000h	予約領域	予約領域 <sup>注1</sup>
3000 0000h	SPI マルチ I/O バス空間 (64MB)	シリアル・フラッシュメモリ (64MB) <sup>注1, 注2</sup>
3400 0000h	予約領域	予約領域 <sup>注1</sup>
4000 0000h	外部アドレス空間 (CS0) (64MB)	NOR フラッシュメモリ 1 (64MB) <sup>注1</sup>
4400 0000h	外部アドレス空間 (CS1) (64MB)	NOR フラッシュメモリ 2 (64MB) <sup>注1</sup>
4800 0000h	外部アドレス空間 (CS2) (64MB)	SDRAM1 (64MB) <sup>注1</sup>
4C00 0000h	外部アドレス空間 (CS3) (64MB)	SDRAM2 (64MB) <sup>注1</sup>
5000 0000h	外部アドレス空間 (CS4) (64MB)	(未使用) <sup>注1</sup>
5400 0000h	外部アドレス空間 (CS5) (64MB)	
5800 0000h	予約領域	予約領域 <sup>注1</sup>
6000 0000h	外部アドレス空間 (CS0) (64MB)	NOR フラッシュメモリ 1 (64MB)
6400 0000h	外部アドレス空間 (CS1) (64MB)	NOR フラッシュメモリ 2 (64MB)
6800 0000h	外部アドレス空間 (CS2) (64MB)	SDRAM1 (64MB)
6C00 0000h	外部アドレス空間 (CS3) (64MB)	SDRAM2 (64MB)
7000 0000h	外部アドレス空間 (CS4) (64MB)	(未使用)
7400 0000h	外部アドレス空間 (CS5) (64MB)	
7800 0000h	予約領域	予約領域
A000 0000h	周辺 I/O レジスタ (1MB)	周辺 I/O レジスタ (1MB)
A010 0000h	予約領域	予約領域
E800 0000h	デバッグ用領域 (64KB)	デバッグ用領域 (64KB)
E801 0000h	予約領域	予約領域
FFFF 0000h	ブート専用領域 (32KB)	ブート専用領域 (32KB)
FFFF 8000h	予約領域	予約領域

注 1. ミラー領域になります。  
注 2. SPI ブートモード版のサンプルプログラムでのみ使用します。

図 5.2 メモリマップ

## 5.2.1 サンプルプログラムのセクション配置

表 5.3 に使用するセクション、図 5.3 にセクション配置（16 ビットバスブート版の例）を示します。

表5.3 使用するセクション

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_WBLOCK	リセット、例外ベクタテーブル	Code	—	ATCM
USER_PRG_WBLOCK	ユーザアプリケーションプログラム領域（実行用）	Code	—	ATCM
USER_DATA_WBLOCK	ユーザアプリケーションプログラム用変数領域（実行用）	Data	—	ATCM
CSTACK	スタック領域	Data	—	ATCM
SVC_STACK	スーパーバイザ（SVC）モードのスタック領域	Data	—	ATCM
IRQ_STACK	IRQモードのスタック領域	Data	—	ATCM
FIQ_STACK	FIQモードのスタック領域	Data	—	ATCM
UND_STACK	未定義（UND）モードのスタック領域	Data	—	ATCM
ABT_STACK	アボート（ABT）モードのスタック領域	Data	—	ATCM
LDR_DATA_WBLOCK <sup>注3</sup>	ローダプログラム用変数領域（実行用）	Data	—	BTCM
LDR_PRG_WBLOCK <sup>注3</sup>	ローダプログラム領域（実行用）	Code	—	BTCM
ldr_param	ローダ用パラメータ <sup>注1</sup>	Data	FLASH <sup>注2</sup>	—
LDR_PRG_RBLOCK <sup>注3</sup>	ローダプログラム領域（格納用） <sup>注1</sup>	Code	FLASH <sup>注2</sup>	—
LDR_DATA_RBLOCK <sup>注3</sup>	ローダプログラム用変数領域（格納用） <sup>注1</sup>	Data	FLASH <sup>注2</sup>	—
VECTOR_RBLOCK	リセット、例外ベクタテーブル領域（格納用） <sup>注1</sup>	Code	FLASH <sup>注2</sup>	—
USER_PRG_RBLOCK	ユーザアプリケーションプログラム領域（格納用） <sup>注1</sup>	Code	FLASH <sup>注2</sup>	—
USER_DATA_RBLOCK	ユーザアプリケーションプログラム用変数領域（格納用） <sup>注1</sup>	Data	FLASH <sup>注2</sup>	—

注1. RAM実行版では、存在しません。

注2. 16ビットバスブートモード版では、NORフラッシュメモリ、SPIブートモード版では、シリアル・フラッシュメモリとなります。

注3. 本サンプルプログラムでは、ローダプログラムで使用する関数（オブジェクト）はローダプログラム用セクション（LDR\_PRG\_xxx、LDR\_DATA\_xxx）で指定する必要があります。ローダプログラムで使用する関数を追加する場合は、リンク設定ファイル（RZ\_T1\_init\_xxx.icf）内でオブジェクトの指定をしてください。

例）ローダプログラムで使用する関数r\_func()（r\_func.c）を追加する場合、例えば以下のように該当セクションで指定します。

```
define block LDR_PRG_RBLOCK with fixed order
{
    ro code object loader_init.o,
    ro code object loader_init2.o,
    ro code object r_atcm_init.o,
    ro code object r_cpg.o,
    ro code object r_ram_init.o,
    ro code object r_mpc.o,
    ro code object bus_init_nor_boot.o,
    ro code object r_reset.o,
    ro code object r_func.o                // 追加した行
}
```

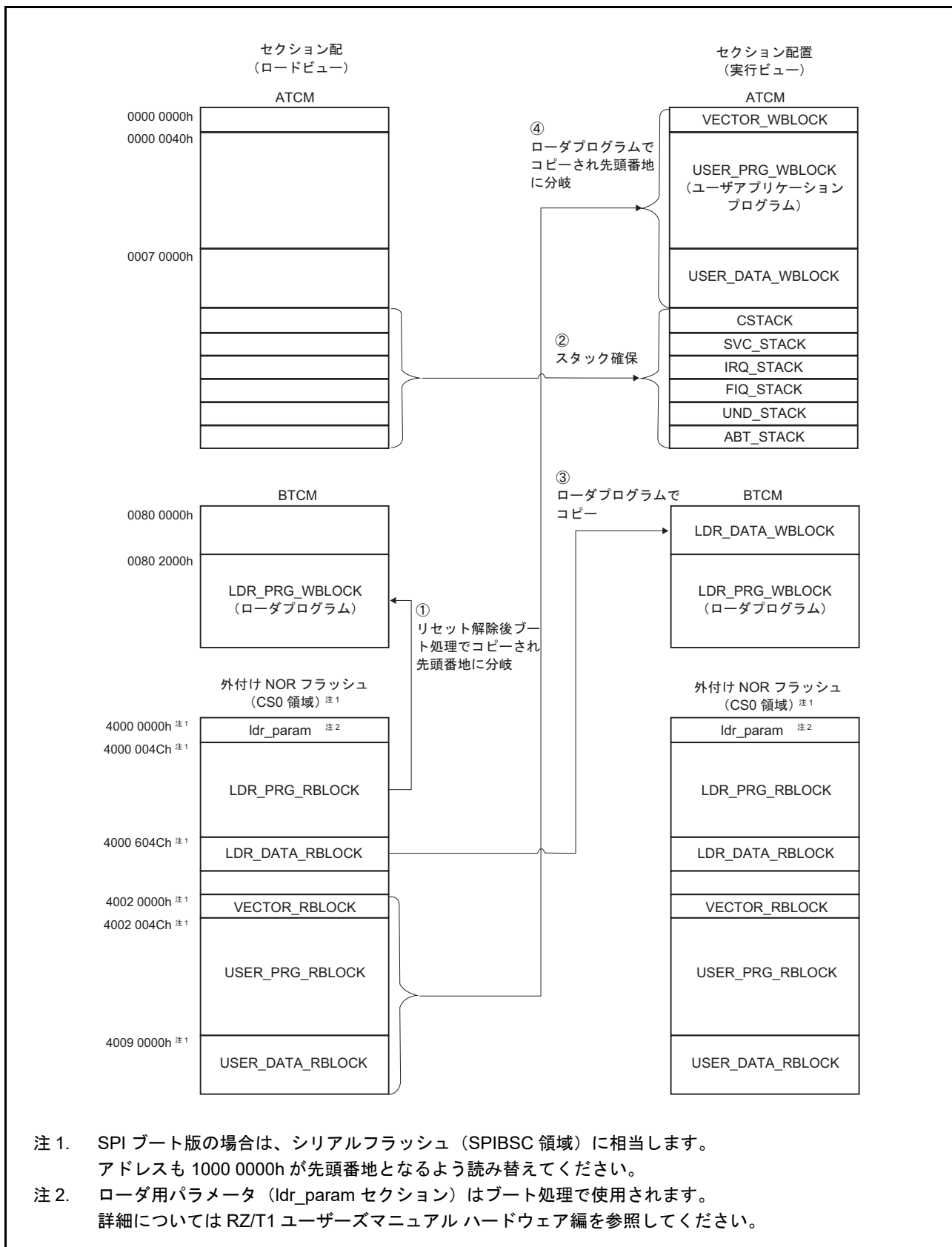


図 5.3 セクション配置（16 ビットバスブート版の例）



## 5.2.2 MPU の設定

表 5.4 に MPU の設定を示します。

表 5.4 MPU の設定

内容	アドレス	サイズ	メモリタイプ
拡張内蔵SRAM I	0400 0000H ~ 0407 FFFFH	512KB	領域0 ノーマル、キャッシュ不可、共有
SPIマルチI/Oバス空間	1000 0000H ~ 13FF FFFFH	64MB	領域1 ノーマル、キャッシュ不可、共有
拡張内蔵SRAM D, I ミラー	2000 0000H ~ 2800 0000H	128MB	領域2 ノーマル、キャッシュ不可、共有
SPIマルチI/Oバス空間ミラー	3000 0000H ~ 33FF FFFFH	64MB	領域3 ノーマル、キャッシュ不可、共有
CS0、CS1空間ミラー	4000 0000H ~ 47FF FFFFH	128MB	領域4 ノーマル、キャッシュ不可、共有
CS2、CS3空間ミラー	4800 0000H ~ 4FFF FFFFH	128MB	領域5 ノーマル、キャッシュ不可、共有
CS4、CS5空間ミラー	5000 0000H ~ 5FFF FFFFH	128MB	領域6 デバイス、共有、命令フェッチ不可
CS0、CS1空間 (NORフラッシュメモリ)	6000 0000H ~ 67FF FFFFH	128MB	領域7 ノーマル、キャッシュ不可、共有
CS2、CS3空間 (SDRAM)	6800 0000H ~ 6FFF FFFFH	128MB	領域8 ノーマル、キャッシュ不可、共有
CS4、CS5空間	7000 0000H ~ 77FF FFFFH	128MB	領域9 デバイス、共有、命令フェッチ不可
周辺I/Oレジスタ デバッグ用領域 ブート専用領域	8000 0000H ~ FFFF FFFFH	2GB	領域10 デバイス、共有、命令フェッチ不可



### 5.3 使用割り込み一覧

表 5.6 にサンプルプログラムで使用する割り込みを示します。

表5.6 サンプルプログラムで使用する割り込み

割り込み (要因ID)	優先度	処理概要
IRQ 端子割り込み5 (IRQ9)	15	SW2が押されると、LED1が約5秒の間点灯します。また多重割り込みを許可しています。
IRQ 端子割り込み12 (IRQ16)	10	SW3が押されると、LED1の状態により2通りの処理をします。LED1が点灯状態の場合、SW3の押下の度にLED5の点灯/消灯を繰り返します。LED1が消灯状態の場合、拡張疑似エラーによるECMリセットが発生します。リセット解除後のリセット判定でLED2が点灯します。

### 5.4 固定幅整数一覧

表 5.7 にサンプルプログラムで使用する固定幅整数を示します。

表5.7 サンプルプログラムで使用する固定幅整数

シンボル	内容
int8_t	8ビット整数、符号あり (標準ライブラリにて定義)
int16_t	16ビット整数、符号あり (標準ライブラリにて定義)
int32_t	32ビット整数、符号あり (標準ライブラリにて定義)
int64_t	64ビット整数、符号あり (標準ライブラリにて定義)
uint8_t	8ビット整数、符号なし (標準ライブラリにて定義)
uint16_t	16ビット整数、符号なし (標準ライブラリにて定義)
uint32_t	32ビット整数、符号なし (標準ライブラリにて定義)
uint64_t	64ビット整数、符号なし (標準ライブラリにて定義)

## 5.5 定数／エラーコード一覧

表 5.8 にサンプルプログラムで使用する定数を示します。

表5.8 サンプルプログラムで使用する定数 (1 / 2)

定数名	設定値	内容
ATCM_WAIT_1_OPT	(0)	ATCMウエイト制御：1-wait最適化あり
ATCM_WAIT_1	(1)	ATCMウエイト制御：1-wait最適化なし
ATCM_WAIT_0	(2)	ATCMウエイト制御：0-wait
CPG_CPUCLK_150_MHz	(0)	CPUクロック周波数：150MHz
CPG_CPUCLK_300_MHz	(1)	CPUクロック周波数：300MHz
CPG_CPUCLK_450_MHz	(2)	CPUクロック周波数：450MHz
CPG_CPUCLK_600_MHz	(3)	CPUクロック周波数：600MHz
CPG_PLL1_OFF	(0)	CPG PLL1制御：停止
CPG_PLL1_ON	(1)	CPG PLL1制御：動作
CPG_SELECT_PLL0	(0)	システムクロックソース選択：PLL0
CPG_SELECT_PLL1	(1)	システムクロックソース選択：PLL1
CPG_CKIO_75_MHz	(0)	外部バスクロック選択：75MHz
CPG_CKIO_50_MHz	(1)	外部バスクロック選択：50MHz
CPG_CKIO_37_5_MHz	(2)	外部バスクロック選択：37.5MHz
CPG_CKIO_30_MHz	(3)	外部バスクロック選択：30MHz
CPG_CKIO_25_MHz	(4)	外部バスクロック選択：25MHz
CPG_CKIO_21_43_MHz	(5)	外部バスクロック選択：21.43MHz
CPG_CKIO_18_75_MHz	(6)	外部バスクロック選択：18.75MHz
CPG_LOCO_ENABLE	(0x00000000)	低速オンチップオシレータ制御：動作
CPG_LOCO_DISABLE	(0x00000001)	低速オンチップオシレータ制御：停止
ECM_COMMAND_KEY	(0x000000A5)	ECM 保護レジスタ書き込み解除キー
ICU_EXT_PIN_0 ~ ICU_EXT_PIN_15	(0) ~ (15)	IRQ端子割り込み番号：0～15
ICU_DETECT_LOW	(0x00)	IRQ検出設定選択：Lowレベル
ICU_DETECT_FALL	(0x04)	IRQ検出設定選択：立ち下がりエッジ
ICU_DETECT_RISE	(0x08)	IRQ検出設定選択：立ち上がりエッジ
ICU_DETECT_RISE_FALL	(0x0C)	IRQ検出設定選択：両エッジ
ICU_DNF_DIVISION_1	(0)	IRQデジタルノイズフィルタ設定： サンプリングクロックPCLKBで使用
ICU_DNF_DIVISION_8	(1)	IRQデジタルノイズフィルタ設定： サンプリングクロックPCLKB/8で使用
ICU_DNF_DIVISION_32	(2)	IRQデジタルノイズフィルタ設定： サンプリングクロックPCLKB/32で使用
ICU_DNF_DIVISION_64	(3)	IRQデジタルノイズフィルタ設定： サンプリングクロックPCLKB/64で使用
ICU_DNF_NO_USE	(4)	IRQデジタルノイズフィルタ設定：デジタルノイズフィルタ無効
ICU_VEC_NUM_1 ~ ICU_VEC_NUM_300	(0) ~ (300)	割り込みベクタ番号：0～300
ICU_TYPE_LEVEL	(0)	割り込み入力検出タイプ：レベル検出
ICU_TYPE_EDGE	(1)	割り込み入力検出タイプ：エッジ検出
ICU_PRIORITY_0 ~ ICU_PRIORITY_15	(0) ~ (15)	割り込み優先レベル：0～15

表5.8 サンプルプログラムで使用する定数 (2/2)

定数名	設定値	内容
ICU_IEC_MASK_SET	(1)	割り込み要求クリア：割り込みマスク（ディスエーブル）状態、IENレジスタの該当ビットをクリア（0）
ICU_PIC_EDGE_CLEAR	(1)	エッジ検出クリア：エッジ検出をクリアします
MPC_IRQ_DISABLE	(0)	割り込み入力機能選択：IRQ入力端子として使用しない
MPC_IRQ_ENABLE	(1)	割り込み入力機能選択：IRQ入力端子として使用する
PORT_DIRECTION_HIZ	(0)	PORT方向制御：不使用（Hi-z入力保護）
PORT_DIRECTION_INPUT	(2)	PORT方向制御：入力（入力ポートとして機能）
PORT_DIRECTION_OUTPUT	(3)	PORT方向制御：出力（出力ポートとして機能（ポートリード可能））
PORT_OUTPUT_LOW	(0)	PORT出力データ：Low出力
PORT_OUTPUT_HIGH	(1)	PORT出力データ：High出力
PORT_MODE_GENERAL	(0)	PORT端子モード制御：汎用入出力ポートとして使用
PORT_MODE_PERIPHERAL	(1)	PORT端子モード制御：周辺機能として使用
PORT_PULL_UPDOWN_DISABLE	(0)	PORT入力プルアップ／プルダウン抵抗制御：入力プルアップ抵抗／プルアップ抵抗無効
PORT_PULL_DOWN	(1)	PORT入力プルアップ／プルダウン抵抗制御：入力プルダウン抵抗有効
PORT_PULL_UP	(2)	PORT入力プルアップ／プルダウン抵抗制御：入力プルアップ抵抗有効
PORT_P10_NORMAL_DRIVE	(0)	P10駆動能力制御：通常出力
PORT_P10_HIGH_DRIVE	(1)	P10駆動能力制御：高駆動出力
RST_SOURCE_RES	(0x00000002)	RES#端子リセット検出フラグ：RES#端子リセット検出
RST_SOURCE_ECM	(0x00000004)	ECMリセット検出フラグ：ECMリセット検出
RST_SOURCE_SWR1	(0x00000008)	ソフトウェアリセット検出フラグ：ソフトウェアリセット検出

## 5.6 構造体／共用体／列挙型一覧

図 5.4、図 5.5 にサンプルプログラムで使用する構造体／共用体／列挙型を示します。

```
typedef enum
{
    ECM_MASTER,
    ECM_CHECKER,
    ECM_COMMON,
    ECM_TYPE_MAX
} ecm_reg_type_t;

typedef struct
{
    uint32_t cdb;
    uint32_t ocdb;
    uint32_t adb;
    uint32_t opdb;
    uint32_t spidb;
    uint32_t cde;
    uint32_t ocde;
    uint32_t ade;
    uint32_t opde;
    uint32_t spide;
    uint32_t sslkp;
    uint32_t spire;
    uint32_t spiwe;

    uint32_t dme; /* Dummy cycle enable at the time of a SPI mode */
    uint32_t addr; /* Address DDR enable */
    uint32_t opdre; /* Option data DDREnable */
    uint32_t spidre; /* Transmission data DDR enable */

    uint32_t dmdb; /* The dummy cycle bit width of the time of a SPI mode */
    uint32_t dmcyc; /* The number of dummy cycles of the time of a SPI mode */

    uint32_t cmd;
    uint32_t ocmd;
    uint32_t addr;
    uint32_t opd[4];
    uint32_t smrdr;
    uint32_t smwdr;
} st_spibsc_spimd_reg_t;
```

図 5.4 サンプルプログラムで使用する構造体／共用体／列挙型

```
typedef struct
{
    uint32_t udef_cmd;
    uint32_t udef_cmd_width;
    uint32_t udef_opd3;
    uint32_t udef_opd2;
    uint32_t udef_opd1;
    uint32_t udef_opd0;
    uint32_t udef_opd_enable;
    uint32_t udef_opd_width;
    uint32_t udef_dmycyc_num;
    uint32_t udef_dmycyc_enable;
    uint32_t udef_dmycyc_width;
    uint32_t udef_data_width;
    uint32_t udef_spbr;
    uint32_t udef_brdv;
    uint32_t udef_addr_width;
    uint32_t udef_addr_mode;
} st_spibsc_cfg_t;
```

図 5.5 サンプルプログラムで使用する構造体／共用体／列挙型

## 5.7 大域変数一覧

表 5.9 に大域変数を示します。

表 5.9 大域変数一覧

型	変数名	内容	使用関数
static uint32_t*	g_pcmd_reg_adrr[]	ECM マスタ／チェッカ保護レジスタ ECM 保護コマンドレジスタ	R_ECM_Write_Reg8 R_ECM_Write_Reg32

## 5.8 関数一覧

表 5.10 に関数を示します。

表 5.10 関数一覧

関数名	ページ番号
loader_init1	25
loader_init2	25
reset_check	25
cpg_init	26
bsc_init	26
copy_to_atcm	26
cache_init	27
set_low_vec	27
main	27
ecm_init	28
icu_init	28
atcm_waitset	28
cpg_pll_wait	29
R_ECM_Init	29
R_ICU_Enable	29
R_ICU_Disable	30
R_ICU_ExtPinInit	30
R_ICU_Regist	31
Userdef_SFLASH_Set_Mode	31
Userdef_SFLASH_Write_Enable	32
Userdef_SFLASH_Busy_Wait	32
sflash_exmode_init	32
sflash_exmode_setting	33
sflash_wait_tend	33
sflash_set_config	34
SPIBSC_Exread_Mode_Config	34
R_IRQ9_isr	35
R_IRQ16_isr	35



## 5.9 API 関数仕様

サンプルプログラムの関数仕様を示します。

### 5.9.1 loader\_init1

---

#### loader\_init1

---

概要	ローダプログラム 1
宣言	loader_init1
説明	スタックやVFP (FPU)、ローダプログラム用の初期値あり変数の初期化を行い、loader_init2 関数へ分岐します。
引数	なし
リターン値	なし
補足	なし

### 5.9.2 loader\_init2

---

#### loader\_init2

---

概要	ローダプログラム 2
宣言	loader_init2
説明	CPU 関連やクロック、バスに関する初期設定を行い、アプリケーションプログラムをATCM に転送し、メイン処理に分岐します。
引数	なし
リターン値	なし
補足	なし

### 5.9.3 reset\_check

---

#### reset\_check

---

概要	リセットフラグ判定処理
宣言	void reset_check(void)
説明	リセット要因を確認し、各シーケンスを実行します。 ECM リセットが発生し、かつエラー要因番号が 35 の時、P77 端子を“H”出力します (LED2 点灯)。
引数	なし
リターン値	なし
補足	なし

### 5.9.4 cpg\_init

---

#### cpg\_init

---

概要	クロック設定処理
宣言	void cpg_init(void)
説明	PLL1 を使用し、CPU クロックを 450MHz に設定します。 また低速オンチップオシレータを動作させます。
引数	なし
リターン値	なし
補足	なし

### 5.9.5 bsc\_init

---

#### bsc\_init

---

概要	バス設定処理
宣言	void bus_init (void)
説明	外部接続メモリ（NOR フラッシュ、シリアルフラッシュ、SDRAM）との接続設定を行います。
引数	なし
リターン値	なし
補足	シリアルフラッシュは SPI ブート版でのみ設定します。 また RAM 実行版では、本処理は実行しません。

### 5.9.6 copy\_to\_atcm

---

#### copy\_to\_atcm

---

概要	ユーザアプリケーションプログラムのコピー処理
宣言	void copy_to_atcm(void)
説明	4 バイト単位で外付けフラッシュ（NOR / シリアル）から内蔵 RAM（ATCM）へユーザアプリケーションプログラムのコピー処理を行います。また例外ベクタやユーザアプリケーション用初期値あり変数のコピーも行います。
引数	なし
リターン値	なし
補足	RAM 実行版では、本処理は実行しません。

### 5.9.7 cache\_init

---

#### cache\_init

---

概要	MPU、キャッシュ設定処理
宣言	void cache_init(void)
説明	I1、D1 キャッシュの全エントリを無効化し、MPU 未定義領域をデフォルトメモリマップに設定します。「表 5.4 MPU の設定」に示すよう各領域を設定した後、MPU をイネーブルし、I1、D1 キャッシュをイネーブルに設定します。
引数	なし
リターン値	なし
補足	なし

### 5.9.8 set\_low\_vec

---

#### set\_low\_vec

---

概要	ロウベクタ設定処理
宣言	void set_low_vec (void)
説明	例外処理ベクタテーブルをハイベクタからロウベクタに切替えます。
引数	なし
リターン値	なし
補足	なし

### 5.9.9 main

---

#### main

---

概要	メイン処理
宣言	int main (void)
説明	ユーザアプリケーションプログラムです。
引数	なし
リターン値	なし
補足	なし

### 5.9.10 port\_init

---

#### port\_init

---

概要	ポート設定処理
宣言	void port_init(void)
説明	ポート PF7、ポート P56、ポート P53 を出力設定し、low 出力します。
引数	なし
リターン値	なし
補足	なし

---

### 5.9.11 ecm\_init

---

#### ecm\_init

---

概要	ECM 初期設定
宣言	void ecm_init(void)
説明	ECM 機能の初期化を行います
引数	なし
リターン値	なし
補足	なし

---

### 5.9.12 icu\_init

---

#### icu\_init

---

概要	割り込み設定処理
宣言	void icu_init(void)
説明	ポート PN5 (IRQ5)、ポート P44 (IRQ12) を外部割り込み入力端子に設定し、割り込み許可に設定します。
引数	なし
リターン値	なし
補足	なし

---

### 5.9.13 atcm\_waitset

---

#### atcm\_waitset

---

概要	ATCM アクセスウェイト設定処理
ヘッダ	r_atcm_init.h
宣言	void atcm_waitset(uint32_t atcm_wait)
説明	ATCM のアクセスウェイトを設定します。
引数	uint32_t atcm_wait
リターン値	なし
補足	CPU からのフェッチアクセスを防ぐため、本設定関数は必ず ATCM 以外のメモリ領域に配置されたプログラムから実行してください。

## 5.9.14 cpg\_pll\_wait

## cpg\_pll\_wait

概要	PLL 安定待ち (100us) 処理
ヘッダ	r_cpg.h
宣言	void cpg_pll_wait(void)
説明	PLL1 の発振安定待ち時間として CMT0 を使用し、100 $\mu$ s 待ちを行います。
引数	なし
リターン値	なし
補足	関数内で CMT0 の初期化や動作停止を行います。 CMT0 を使用する場合はご注意ください。

## 5.9.15 R\_ECM\_Init

## R\_ECM\_Init

概要	ECM 機能初期化処理
ヘッダ	r_ecm.h
宣言	void R_ECM_Init(void)
説明	すべてのエラー要因をクリアし、ERROROUT# 端子の出力をインアクティブ (High) に設定します。
引数	なし
リターン値	なし
補足	関数内で CMT0 の初期化や動作停止を行います。 CMT0 を使用する場合はご注意ください。

## 5.9.16 R\_ICU\_Enable

## R\_ICU\_Enable

概要	割り込み許可設定
ヘッダ	r_icu_init.h
宣言	void R_ICU_Enable(uint32_t vec_num)
説明	引数の割り込みベクタ番号から、割り込みイネーブルレジスタを算出し、ベクタ番号に該当する割り込みを許可に設定します。
引数	uint32_t vec_num      割り込みベクタ番号を設定します。 設定範囲 (1 ~ 300)
リターン値	なし
補足	なし

## 5.9.17 R\_ICU\_Disable

## R\_ICU\_Disable

概要	割り込み禁止設定	
ヘッダ	r_icu_init.h	
宣言	void R_ICU_Disable(uint32_t vec_num)	
説明	引数の割り込みベクタ番号から、割り込みイネーブルクリアレジスタを算出し、ベクタ番号に該当する割り込みを禁止に設定します。	
引数	uint32_t vec_num	割り込みベクタ番号を設定します。 設定範囲 (1 ~ 300)
リターン値	なし	
補足	なし	

## 5.9.18 R\_ICU\_ExtPinInit

## R\_ICU\_ExtPinInit

概要	外部割り込み設定処理	
ヘッダ	r_icu_init.h	
宣言	void R_ICU_ExtPinInit(uint16_t pin_num, uint8_t detect, uint32_t dnf_set)	
説明	引数で渡す設定に従い、外部割り込み端子をします。	
引数	uint16_t pin_num	外部割り込みを設定するピン番号を設定します。 設定範囲 (0 ~ 15)
	uint8_t detect	外部割り込みの検出方法を設定します。 Low レベル、立下りエッジ、立ち上がりエッジ、両エッジ
	uint32_t dnf_set	デジタルノイズフィルタの有効/無効を設定します。有効の場合はサンプリングクロック数を指定します。
リターン値	なし	
補足	なし	

## 5.9.19 R\_ICU\_Regist

## R\_ICU\_Regist

概要	割り込みコントローラへの登録	
ヘッダ	r_icu_init.h	
宣言	void R_ICU_Regist(uint32_t vec_num, uint32_t type, uint32_t priority, uint32_t isr_addr)	
説明	引数で渡す設定に従い、割り込みコントローラを設定します。	
引数	uint32_t vec_num	割り込みベクタ番号を設定します。 設定範囲 (1 ~ 300)
	uint32_t type	割り込み検出タイプ
	uint32_t priority	割り込み優先レベルを設定します。 ベクタ番号が 1 ~ 255 の場合、割り込み優先レベルは、 0 ~ 15、ベクタ番号が 256 ~ 300 の場合、割り込み優先レベルは、16 ~ 31
	uint32_t isr_addr	割り込みサービスルーチンの先頭アドレスを設定します。
リターン値	なし	
補足	なし	

## 5.9.20 Userdef\_SFLASH\_Set\_Mode

## Userdef\_SFLASH\_Set\_Mode

概要	シリアルフラッシュメモリ内レジスタ設定 (ユーザ定義)	
宣言	int32_t Userdef_SFLASH_Set_Mode(uint32_t data_width, uint32_t addr_mode)	
説明	本関数内で使用するシリアルフラッシュメモリに応じて、SPIBSC を外部アドレスリードモードで使用する場合に必要なシリアルフラッシュメモリ内レジスタを設定する処理を実装してください。	
引数	uint32_t data_width	データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT : 1 ビット幅 SPIBSC_4BIT : 4 ビット幅
	uint32_t addr_mode	アドレスモード設定 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時にシリアルフラッシュメモリに出力するアドレスを設定します。 SPIBSC_OUTPUT_ADDR_24 : 24 ビットアドレス出力 SPIBSC_OUTPUT_ADDR_32 : 32 ビットアドレス出力
リターン値	0 : 設定成功 -1 : 設定失敗	
補足	16 ビットバスブートモード版、RAM 実行版では実施されません。	

## 5.9.21 Userdef\_SFLASH\_Write\_Enable

## Userdef\_SFLASH\_Write\_Enable

概要	シリアルフラッシュメモリライト許可設定（ユーザ定義）
宣言	int32_t Userdef_SFLASH_Write_Enable(void)
説明	本関数内で使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内レジスタを設定し、ライト許可に設定する処理を実装してください。
引数	なし
リターン値	0：設定成功 -1：設定失敗
補足	16ビットバスブートモード版、RAM実行版では実施されません。

## 5.9.22 Userdef\_SFLASH\_Busy\_Wait

## Userdef\_SFLASH\_Busy\_Wait

概要	シリアルフラッシュメモリレディー待ち処理（ユーザ定義）
宣言	int32_t Userdef_SFLASH_Busy_Wait(uint32_t data_width)
説明	本関数内で使用するシリアルフラッシュメモリに応じて、シリアルフラッシュメモリ内のレジスタを読み出し、シリアルフラッシュメモリがレディー状態に遷移する処理を実装してください。
引数	uint32_t data_width      データリードビット幅 SPI マルチ I/O バス空間へのリードを SPI 通信に変換する時のシリアルフラッシュメモリのデータリードビット幅 SPIBSC_1BIT：1ビット幅 SPIBSC_4BIT：4ビット幅
リターン値	なし
補足	16ビットバスブートモード版、RAM実行版では実施されません。

## 5.9.23 sflash\_exmode\_init

## sflash\_exmode\_init

概要	SPIBSC 外部アドレスモード初期設定処理
ヘッダ	r_spibsc_ioaset_api.h
宣言	int32_t sflash_exmode_init(st_spibsc_cfg_t *spibsccfg)
説明	SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。初期設定後、外部アドレスリードモードに設定します。
引数	st_spibsc_cfg_t *spibsccfg      SPIBSC 外部アドレスリード設定
リターン値	0：正常終了 -1：エラー
補足	16ビットバスブートモード版、RAM実行版では実施されません。



### 5.9.24 sflash\_exmode\_setting

---

#### sflash\_exmode\_setting

---

概要	SPIBSC 初期設定処理	
ヘッダ	r_spibsc_ioaset_api.h	
宣言	int32_t sflash_exmode_setting(st_spibsc_cfg_t *spibsccfg)	
説明	<p>SPIBSC にてシリアルフラッシュメモリを制御するために必要な初期設定および SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行います。また初期設定に合わせて、シリアルフラッシュメモリ内のレジスタ設定を行います。初期設定後、外部アドレスリードモードに設定します。本関数内で SPIBSC 外部アドレスモード初期設定関数 (sflash_exmode_init) を実行します。</p>	
引数	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定
リターン値	<p>0 : 正常終了 -1 : エラー</p>	
補足	16 ビットバスブートモード版、RAM 実行版では実施されません。	

### 5.9.25 sflash\_wait\_tend

---

#### sflash\_wait\_tend

---

概要	SPIBSC データ転送待ち処理	
ヘッダ	r_spibsc_ioaset_api.h	
宣言	void sflash_wait_tend(void)	
説明	SPIBSC より、データ転送が終了するのを待ちます。	
引数	なし	
リターン値	なし	
補足	16 ビットバスブートモード版、RAM 実行版では実施されません。	

## 5.9.26 sflash\_set\_config

---

**sflash\_set\_config**

---

概要	SPIBSC 外部アドレスリード設定処理	
ヘッダ	r_spibsc_ioaset_api.h	
宣言	int32_t sflash_set_config(st_spibsc_cfg_t *spibsccfg)	
説明	使用するシリアルフラッシュメモリに応じて、SPIBSC を外部アドレスリードモードで使用するための設定内容を決定します。 本関数内で、ユーザ定義関数（SPIBSC 外部アドレスリード設定関数：Userdef_SPIBSC_Set_Config）を実行します。	
引数	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定
リターン値	0 : 正常終了 -1 : エラー	
補足	16 ビットバスブートモード版、RAM 実行版では実施されません。	

## 5.9.27 SPIBSC\_Exread\_Mode\_Config

---

**SPIBSC\_Exread\_Mode\_Config**

---

概要	SPIBSC コンフィギュレーション処理	
宣言	static int32_t SPIBSC_Exread_Mode_Config(st_spibsc_cfg_t *spibsccfg)	
説明	引数で渡された各データの設定範囲のチェックを行います。	
引数	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定
リターン値	0 : 成功 -1 : 失敗	
補足	16 ビットバスブートモード版、RAM 実行版では実施されません。	

## 5.9.28 Userdef\_SPIBSC\_Set\_Config

---

**Userdef\_SPIBSC\_Set\_Config**

---

概要	SPIBSC 外部アドレスリード設定処理（ユーザ定義）	
宣言	void Userdef_SPIBSC_Set_Config(st_spibsc_cfg_t *spibsccfg)	
説明	使用するシリアルフラッシュメモリに応じて、SPIBSC 外部アドレスリードモードの設定する内容を決定します。本関数にて、引数 spibsccfg にて指定された領域に SPIBSC を外部アドレスリードモードで使用するために必要な初期設定を行ってください。	
引数	st_spibsc_cfg_t *spibsccfg	SPIBSC 外部アドレスリード設定
リターン値	0 : 正常終了 -1 : エラー	
補足	16 ビットバスブートモード版、RAM 実行版では実施されません。	

---

### 5.9.29 R\_IRQ9\_isr

---

#### R\_IRQ9\_isr

---

概 要	IRQ9 割り込み (IRQ 端子割り込み 5) 処理
宣 言	void R_IRQ9_isr(void)
説 明	LED1 を約 5 秒の間点灯させます。 また多重割り込みを許可します。
引 数	なし
リターン値	なし
補 足	なし

---

### 5.9.30 R\_IRQ16\_isr

---

#### R\_IRQ16\_isr

---

概 要	IRQ16 割り込み (IRQ 端子割り込み 12) 処理
宣 言	void R_IRQ16_isr(void)
説 明	LED1 が点灯状態の場合、LED5 出力を反転します。 LED1 が消灯状態の場合、拡張疑似エラー 35 を発生させ、ECM リセットを発生させます。
引 数	なし
リターン値	なし
補 足	RAM 実行版では ECM リセットの発生はせず、LED2 を点灯させます。

## 5.10 フローチャート

### 5.10.1 ローダプログラム 1 処理

図 5.6 にローダプログラム 1 処理のフローチャートを示します。

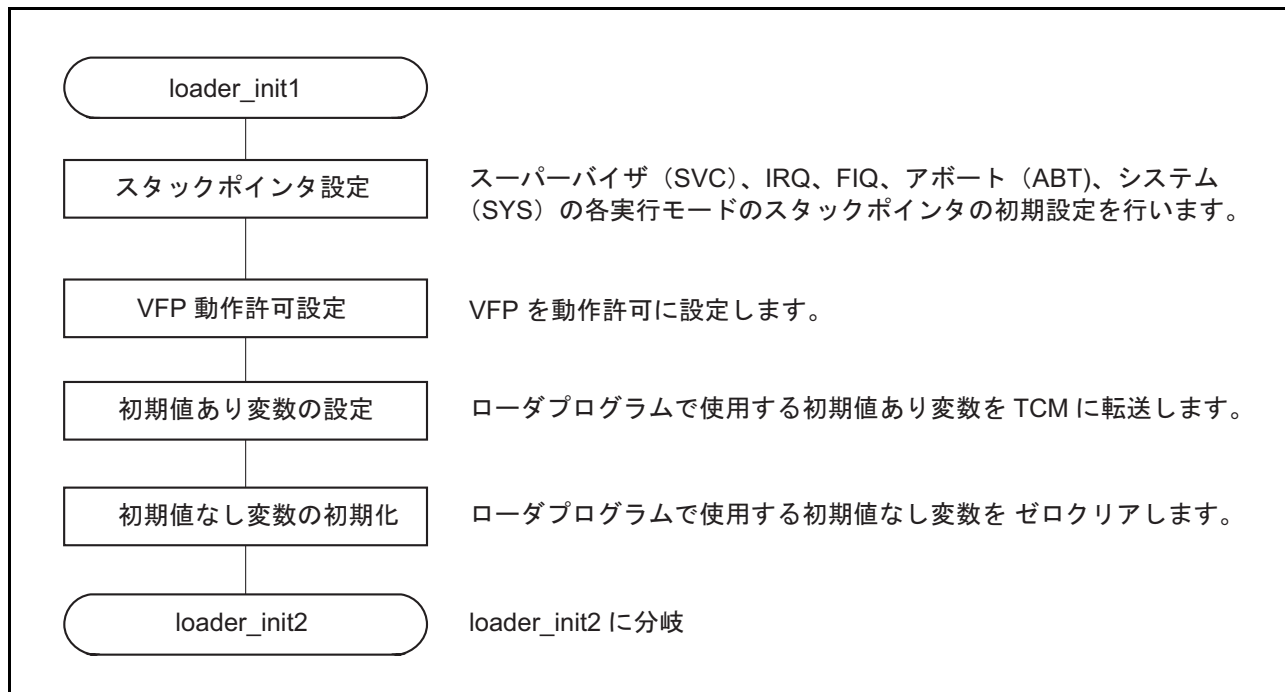


図 5.6 ローダプログラム 1 処理

## 5.10.2 ローダプログラム2処理

図 5.7 にローダプログラム2処理のフローチャートを示します。

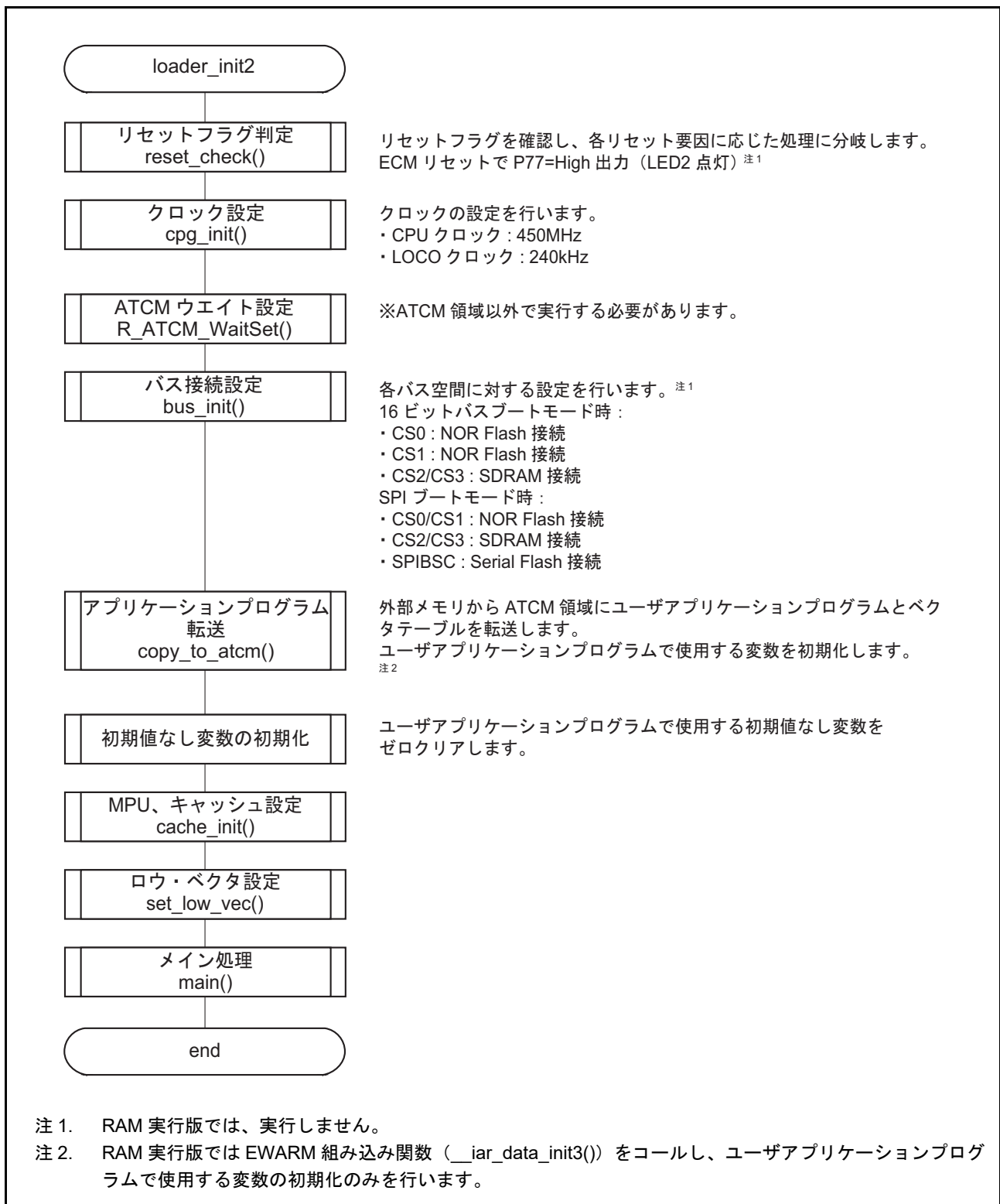


図 5.7 ローダプログラム2処理

### 5.10.3 メイン処理

図 5.8 にメイン処理のフローチャートを示します。

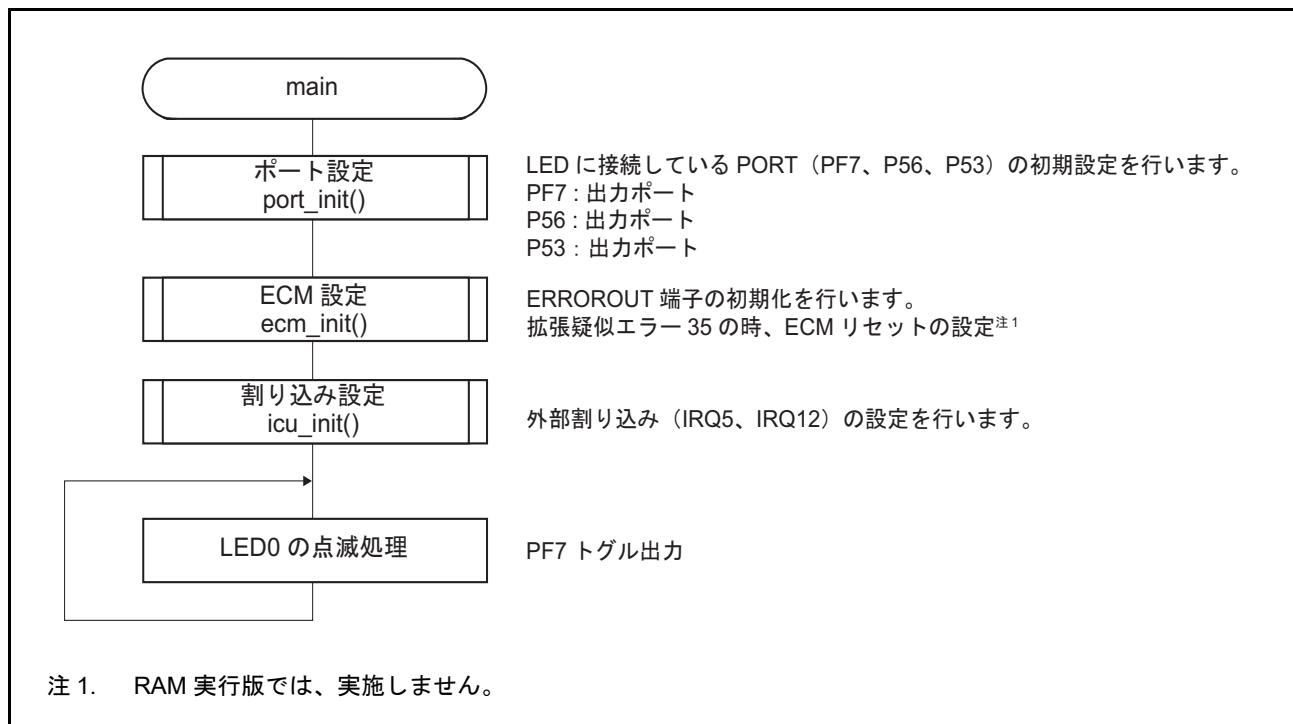


図 5.8 メイン処理

### 5.10.4 リセットフラグ判定処理

図 5.9、図 5.10 にリセットフラグ判定処理のフローチャートを示します。

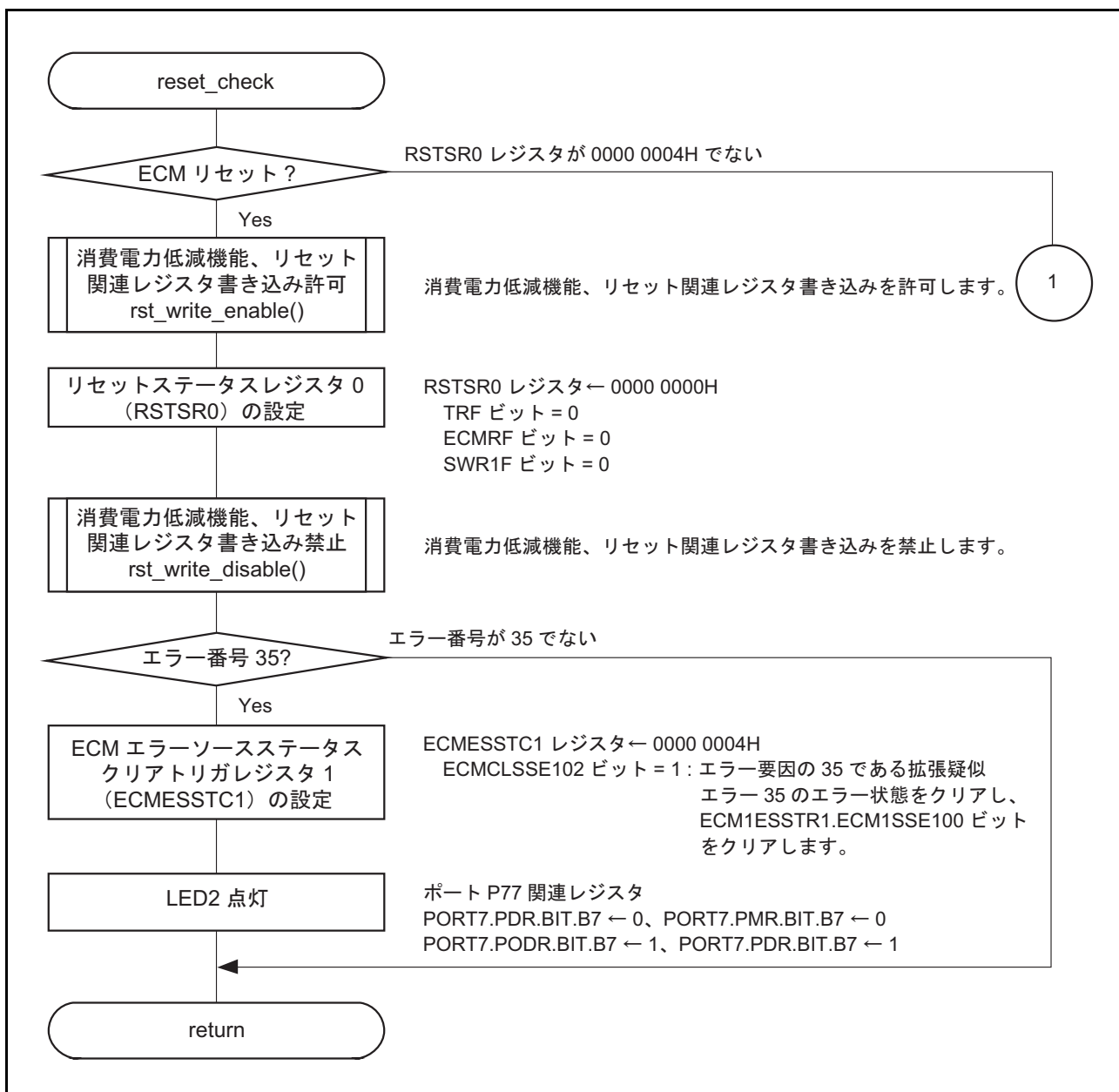


図 5.9 リセットフラグ判定処理

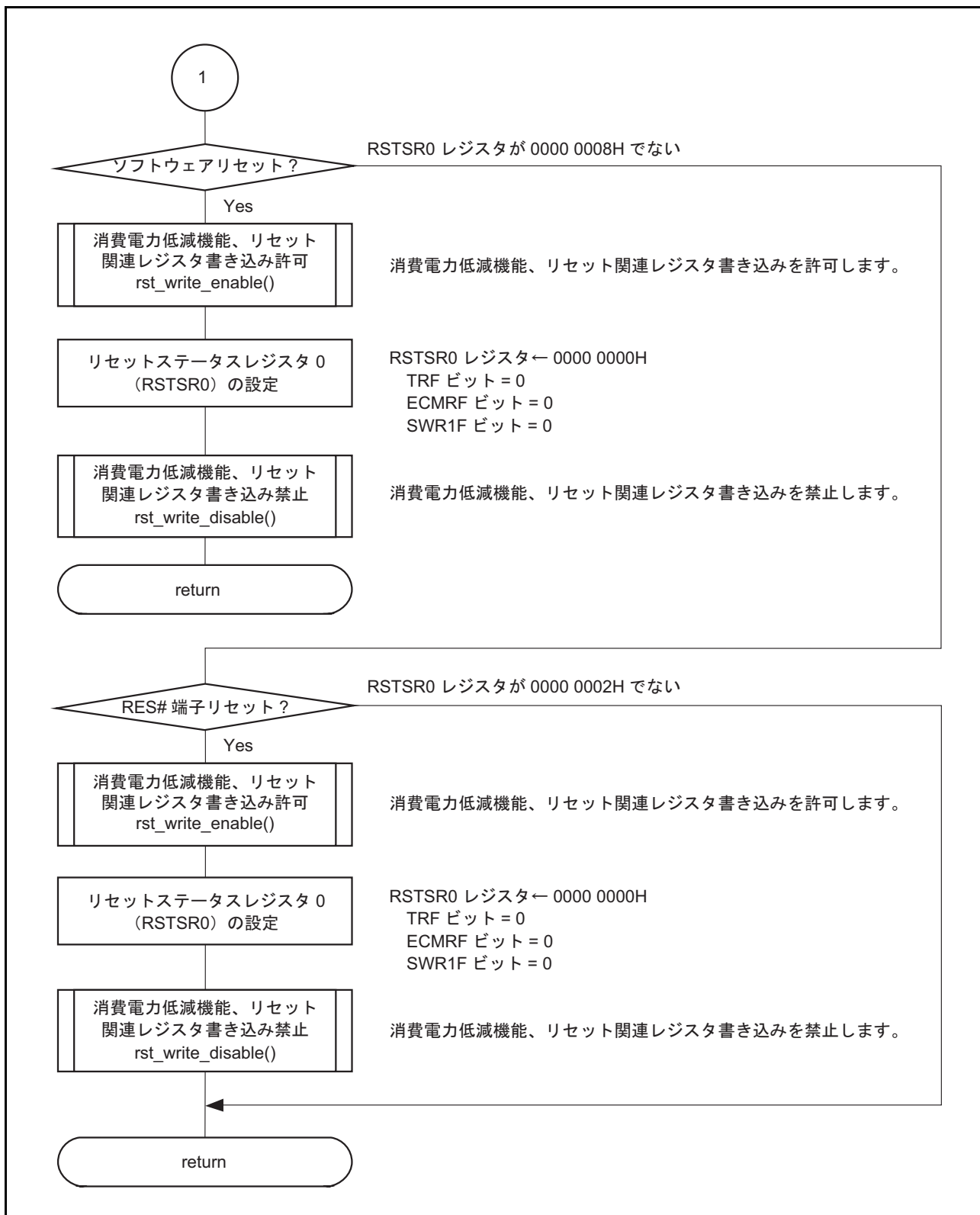


図 5.10 リセットフラグ判定処理



## 5.10.5 クロック設定処理

図 5.11 にクロック設定処理のフローチャートを示します。



図 5.11 クロック設定処理

## 5.10.6 バス設定処理

図 5.12 ~ 図 5.14 にバス設定処理のフローチャートを示します。

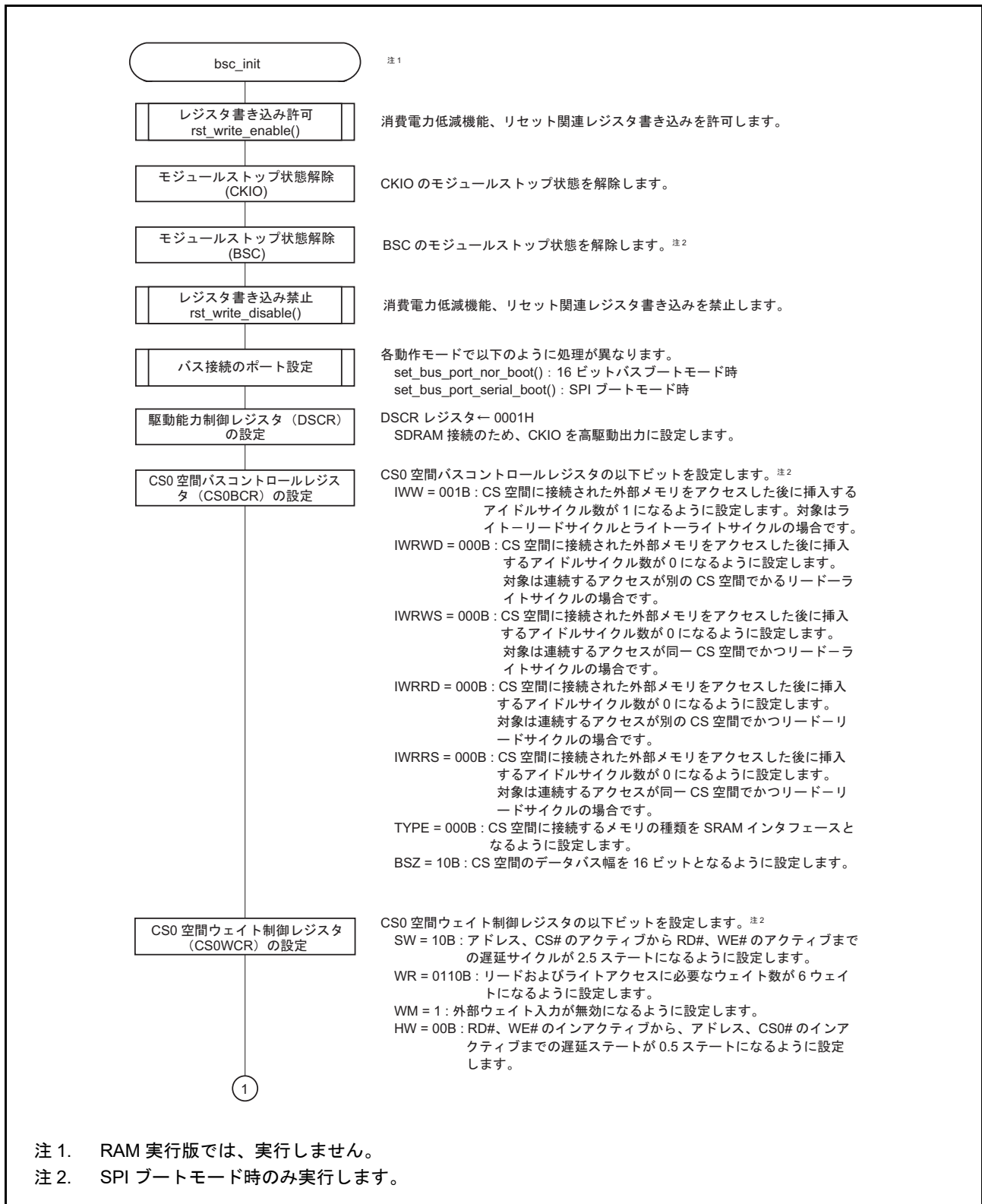


図 5.12 バス設定処理 (1)

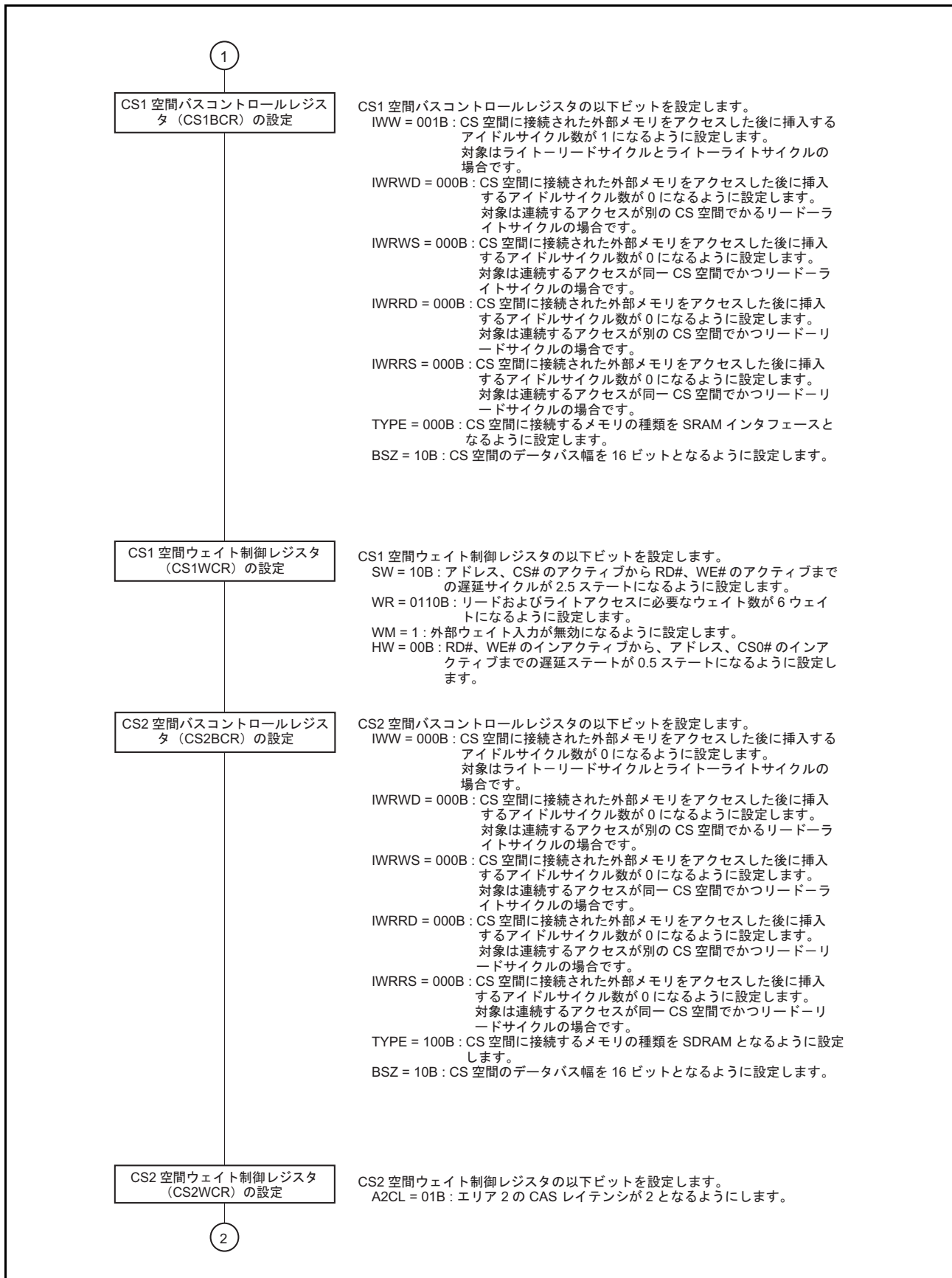


図 5.13 バス設定処理 (2)

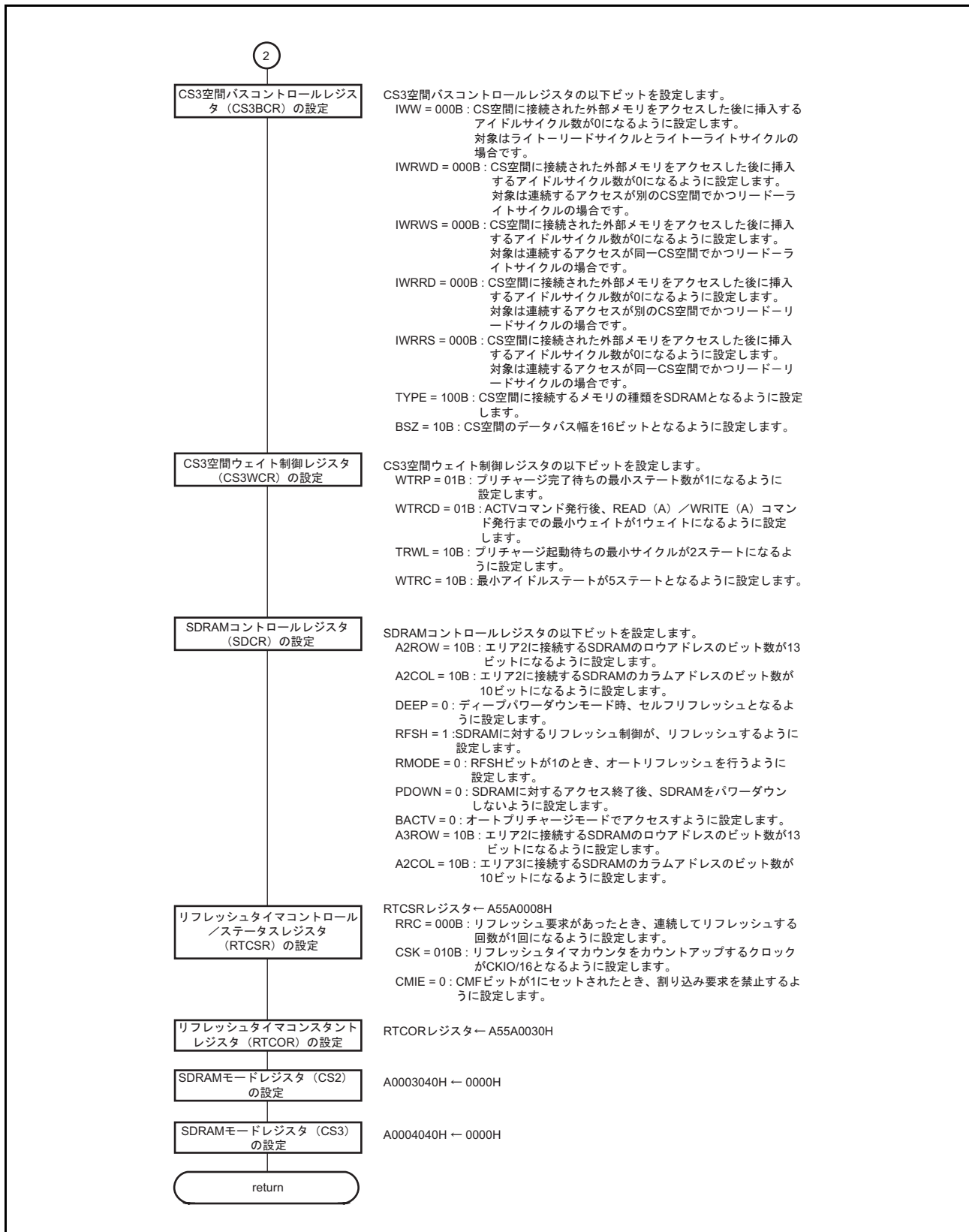


図 5.14 バス設定処理 (3)

### 5.10.7 アプリケーションプログラム転送処理

図 5.15 にアプリケーションプログラム転送処理のフローチャートを示します。

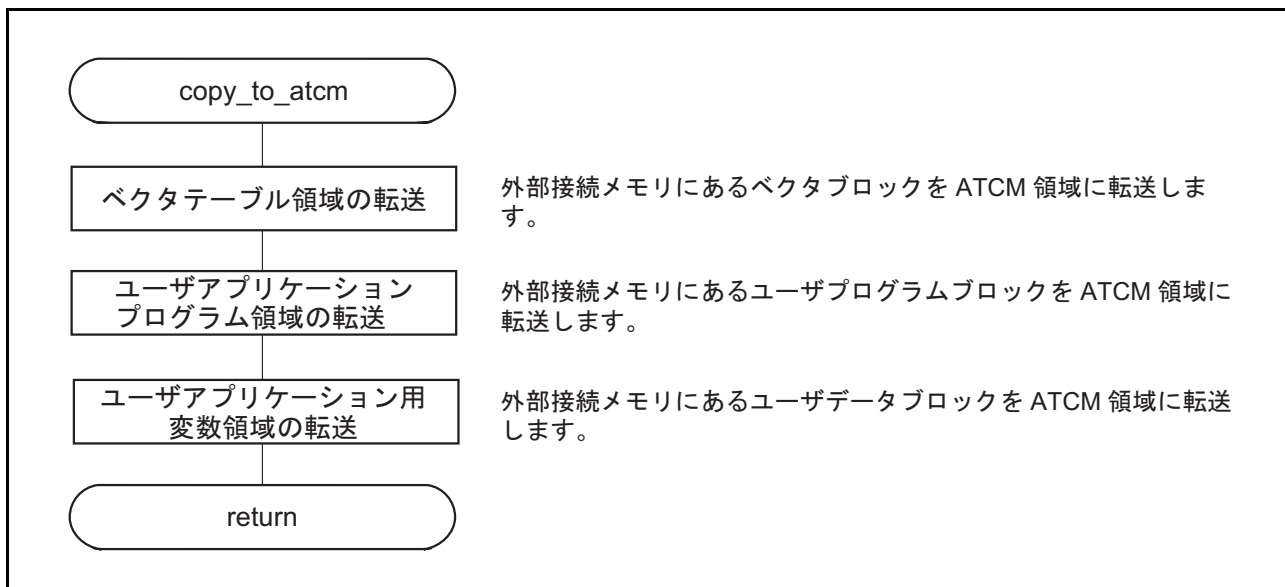


図 5.15 アプリケーションプログラム転送処理

### 5.10.8 MPU、キャッシュ設定処理

図 5.16 に MPU、キャッシュ設定処理のフローチャートを示します。

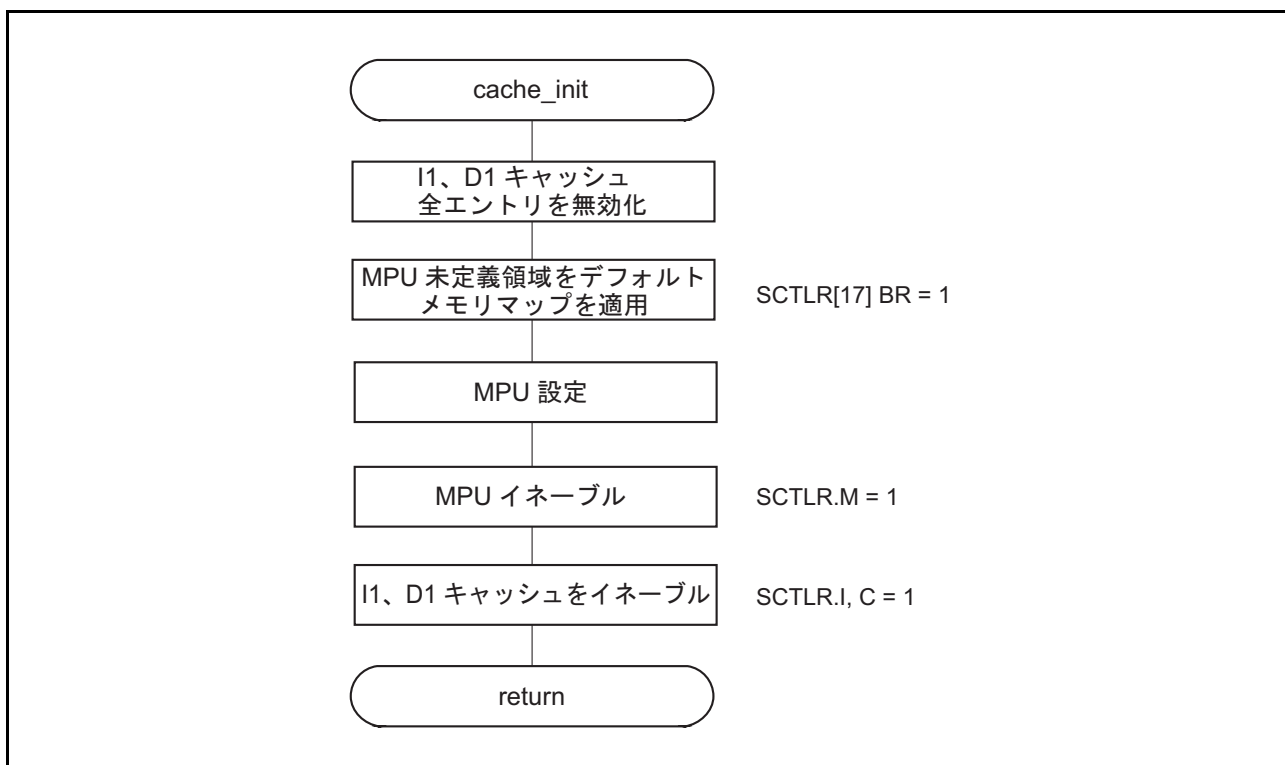


図 5.16 MPU、キャッシュ設定処理

### 5.10.9 ロウベクタ設定処理

図 5.17 にロウベクタ設定処理のフローチャートを示します。

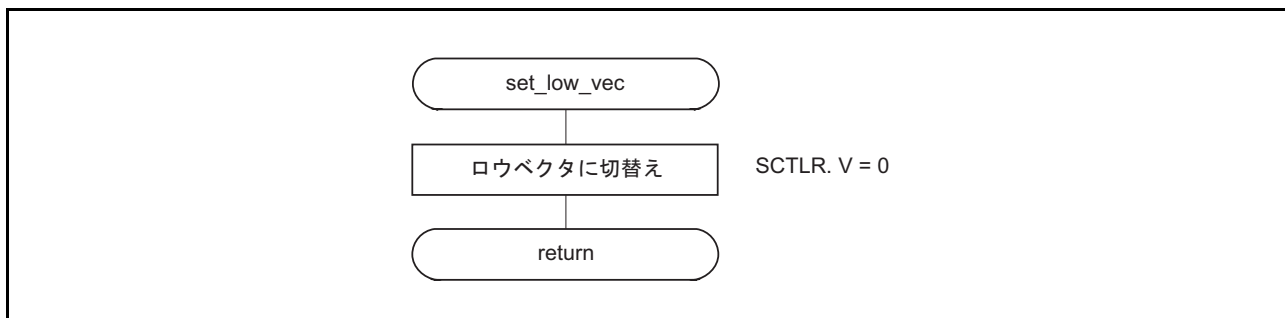


図 5.17 ロウベクタ設定処理

### 5.10.10 ポート設定処理

図 5.18 にポート設定処理のフローチャートを示します。

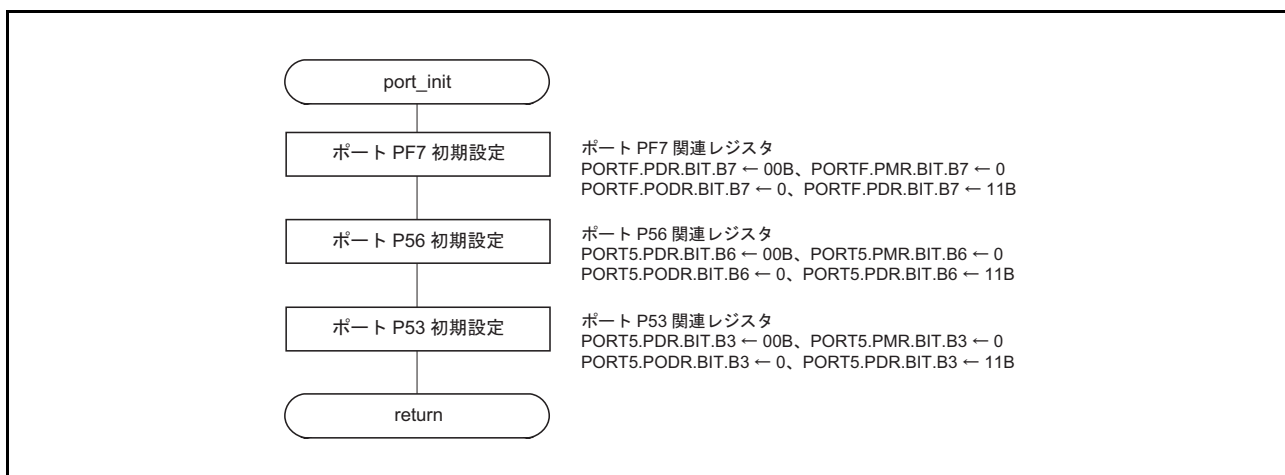


図 5.18 ポート設定処理

### 5.10.11 ECM 設定処理

図 5.19 に ECM 設定処理のフローチャートを示します。

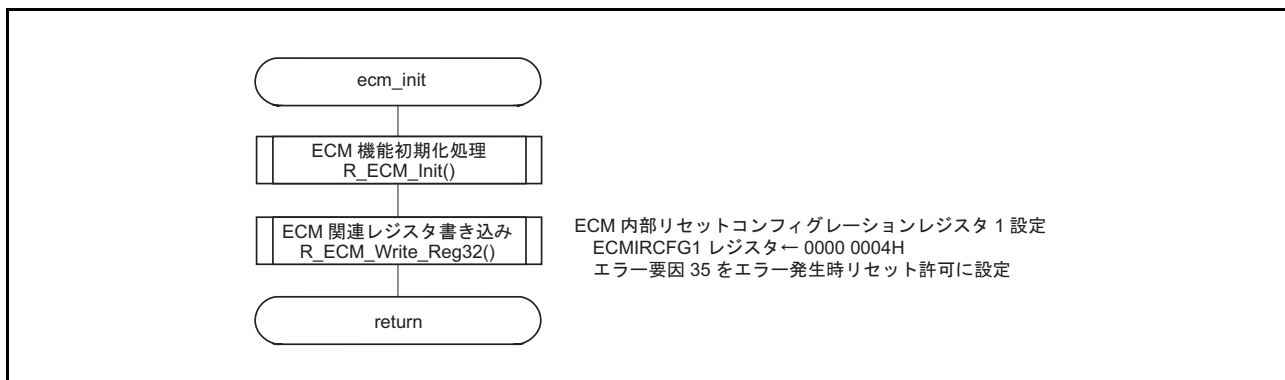


図 5.19 ECM 設定処理

### 5.10.12 ICU 設定処理

図 5.20 に ICU 設定処理のフローチャートを示します。

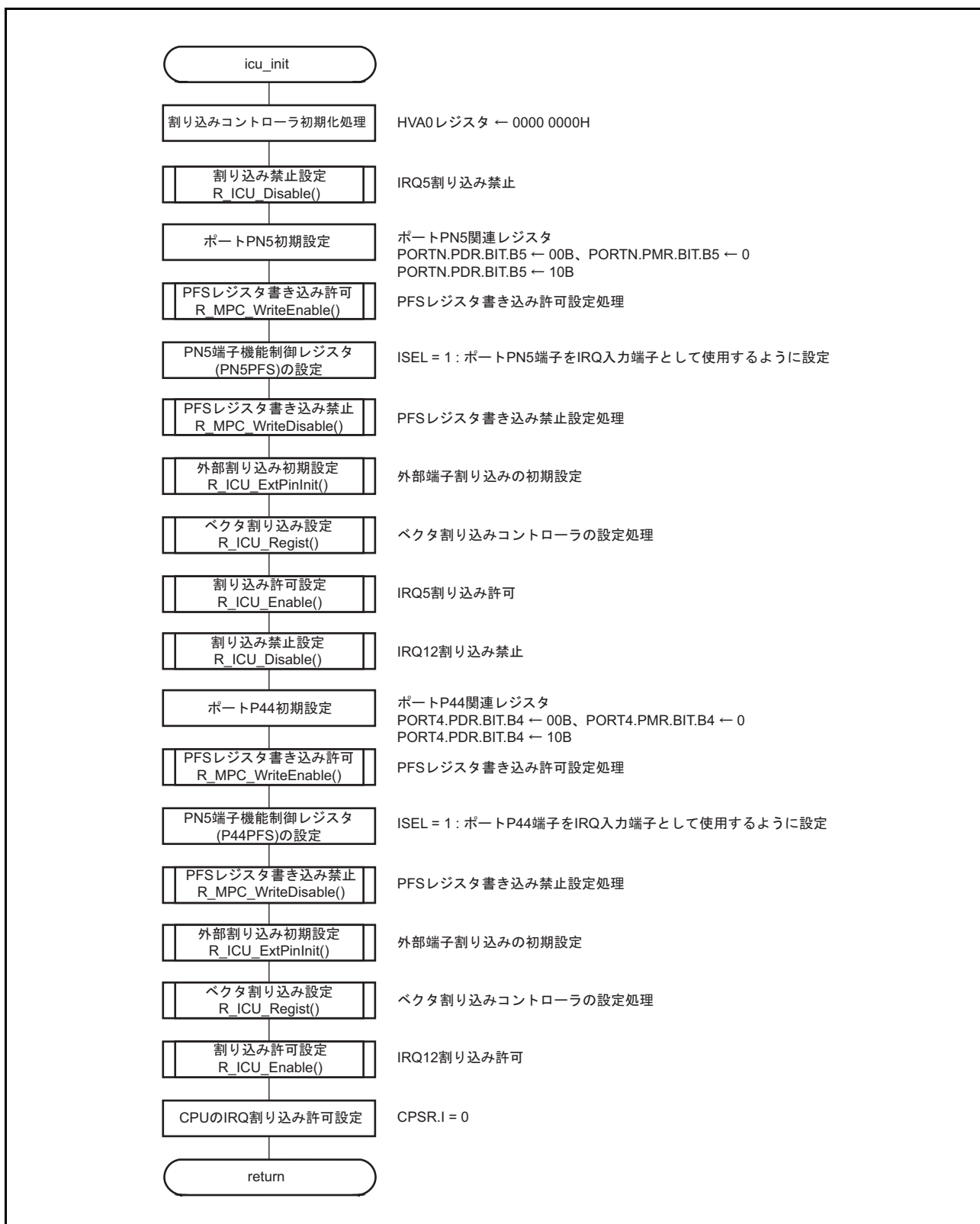


図 5.20 ICU 設定処理

## 5.10.13 ATCM アクセスウェイト設定処理

図 5.21 に ATCM アクセスウェイト設定処理のフローチャートを示します。

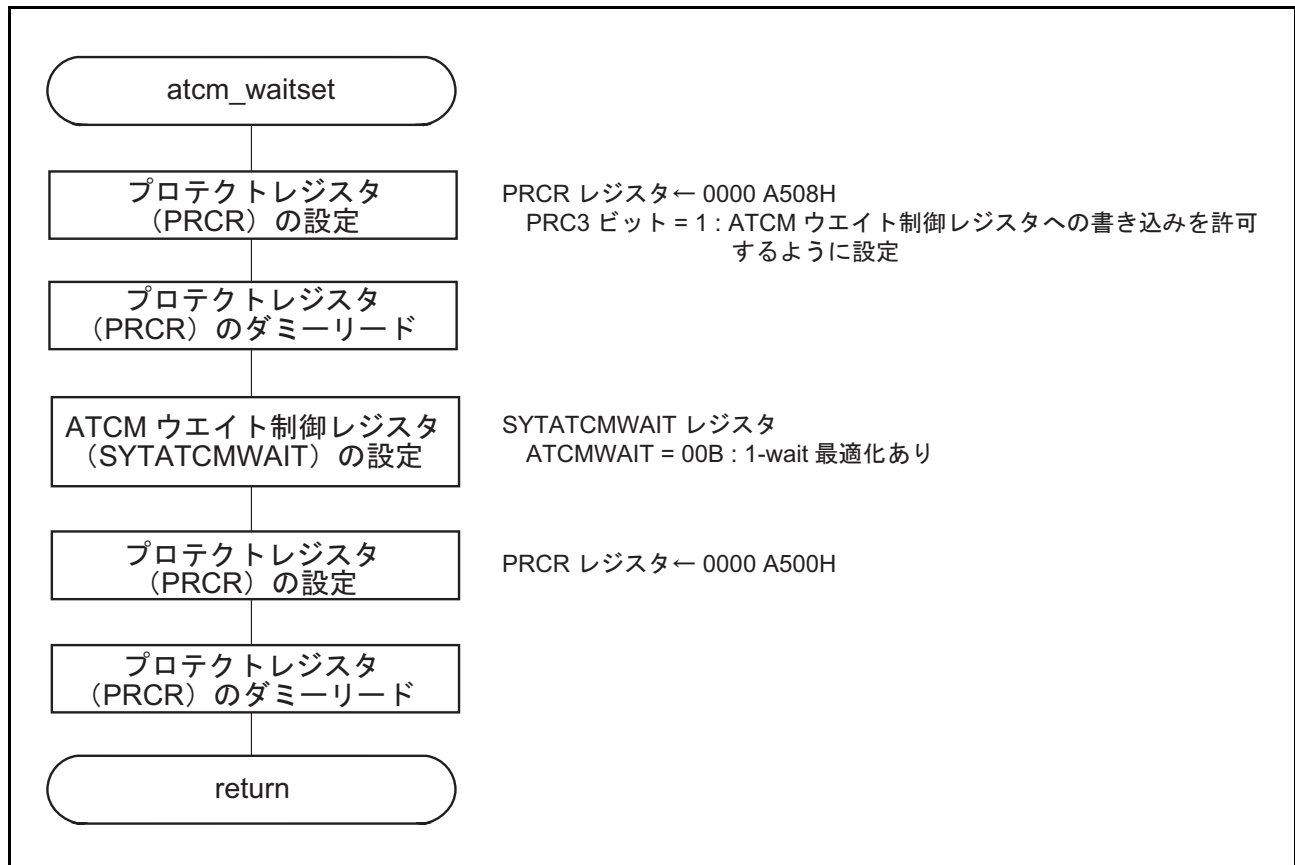


図 5.21 ATCM アクセスウェイト設定処理



5.10.14 PLL 安定待ち (100 $\mu$ s) 処理

図 5.22 に PLL 安定待ち (100 $\mu$ s) 処理のフローチャートを示します。

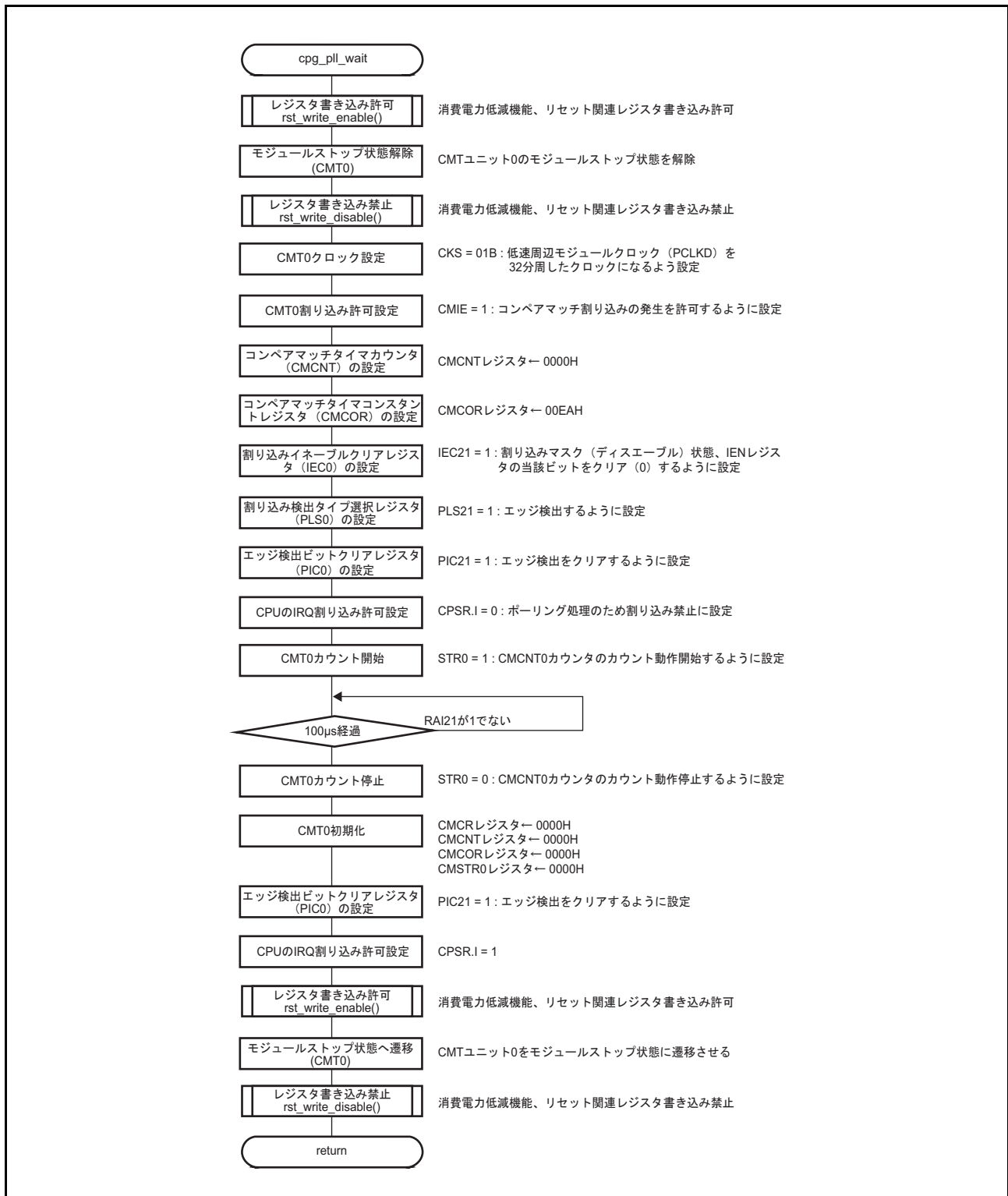


図 5.22 PLL 安定待ち (100 $\mu$ s) 処理

## 5.10.15 ECM 初期設定処理

図 5.23 に ECM 初期設定処理のフローチャートを示します。

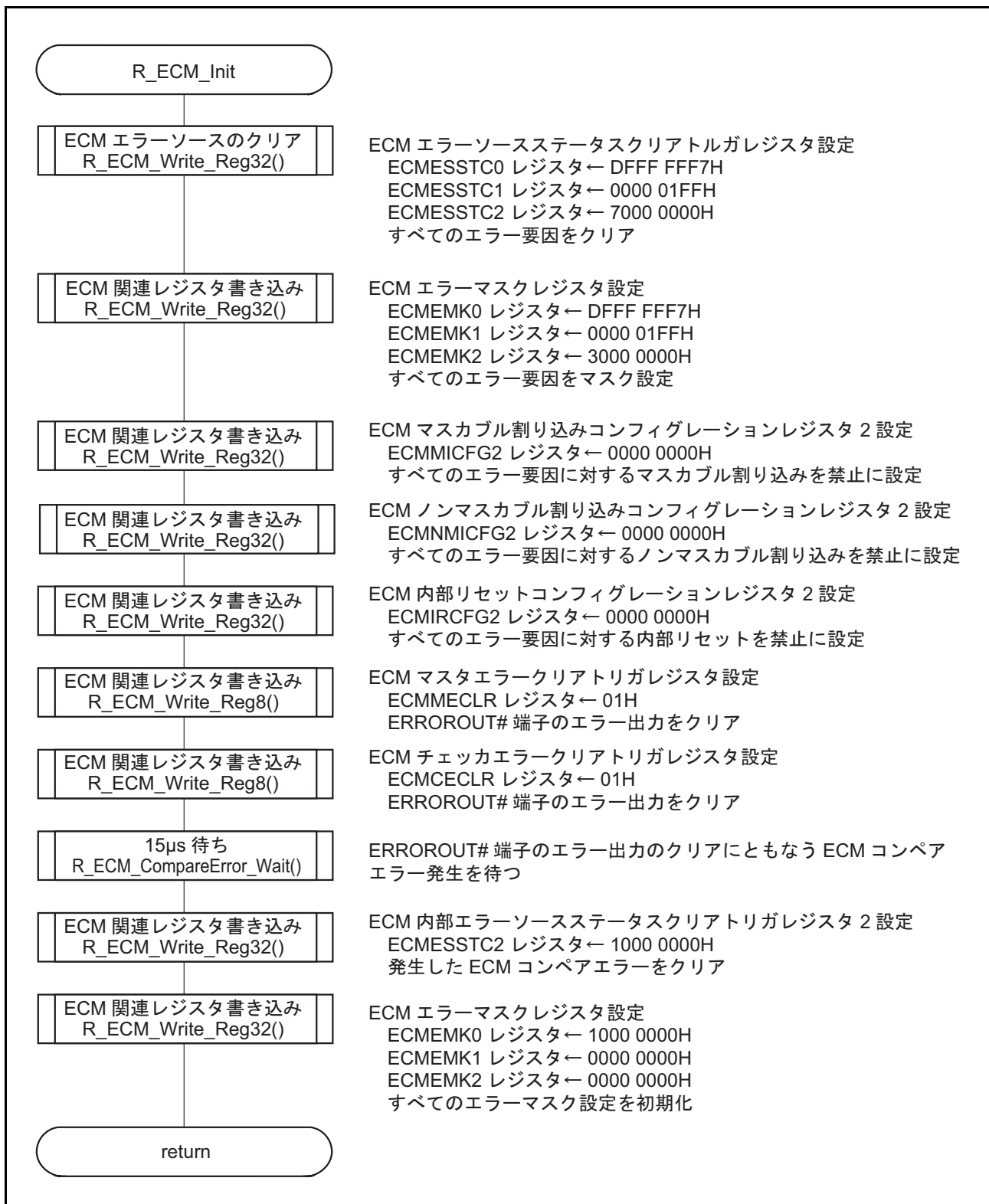


図 5.23 ECM 初期設定処理

### 5.10.16 割り込み許可設定処理

図 5.24 に割り込み許可設定処理のフローチャートを示します。

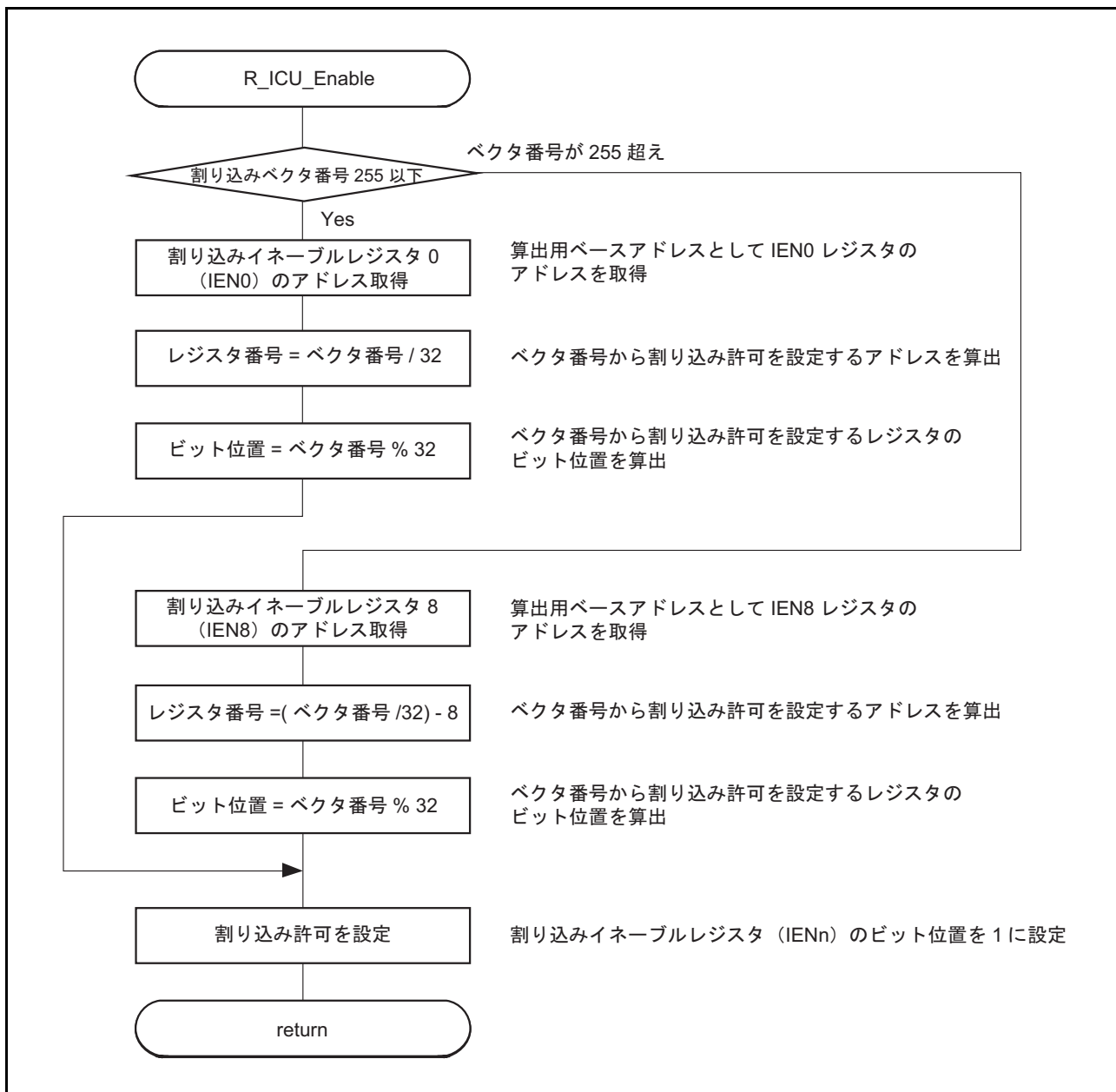


図 5.24 割り込み許可設定処理

### 5.10.17 割り込み禁止設定処理

図 5.25 に割り込み禁止設定処理のフローチャートを示します。

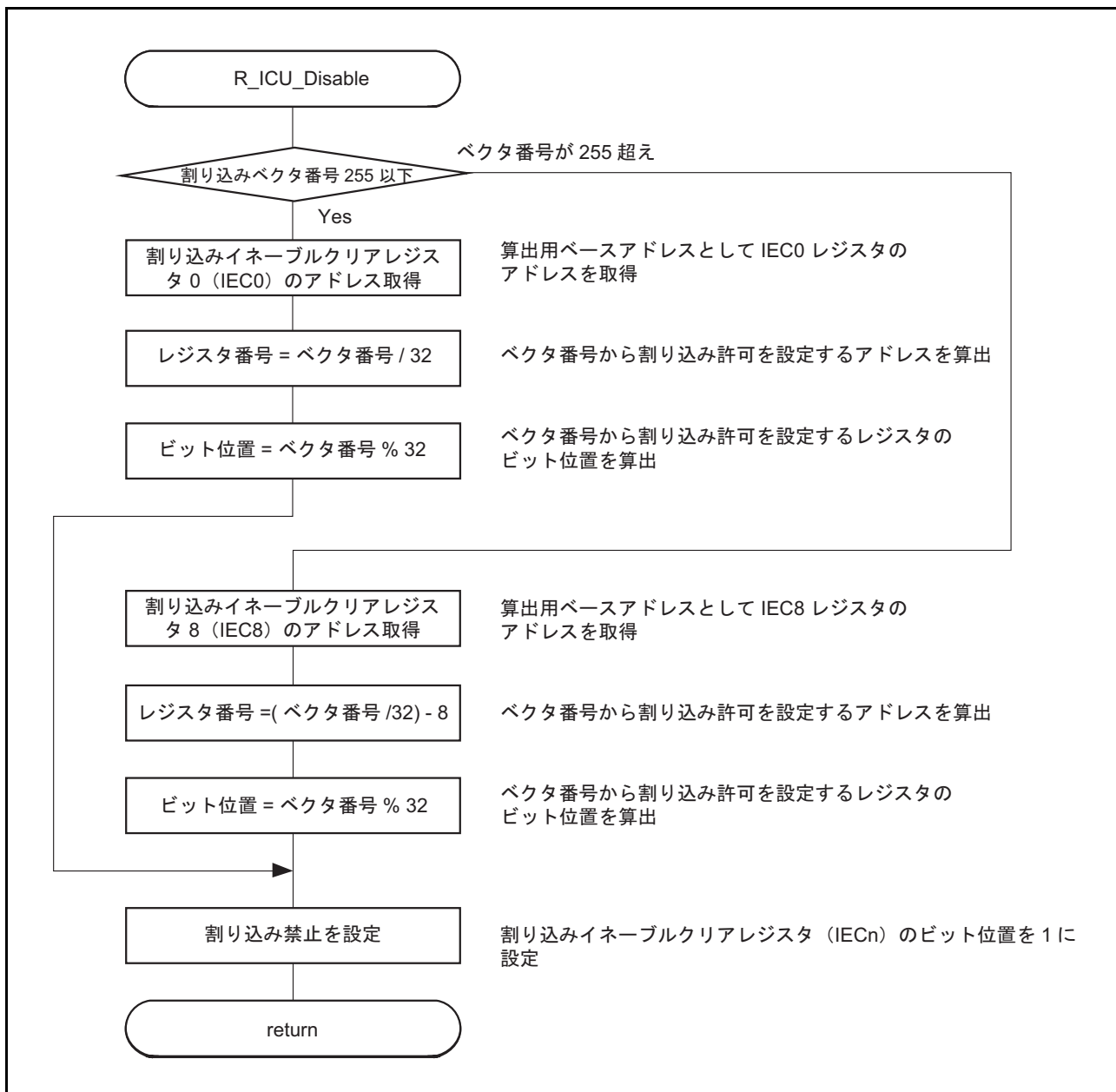


図 5.25 割り込み禁止設定処理

### 5.10.18 外部割り込み設定処理

図 5.26 に外部割り込み端子設定処理のフローチャートを示します。

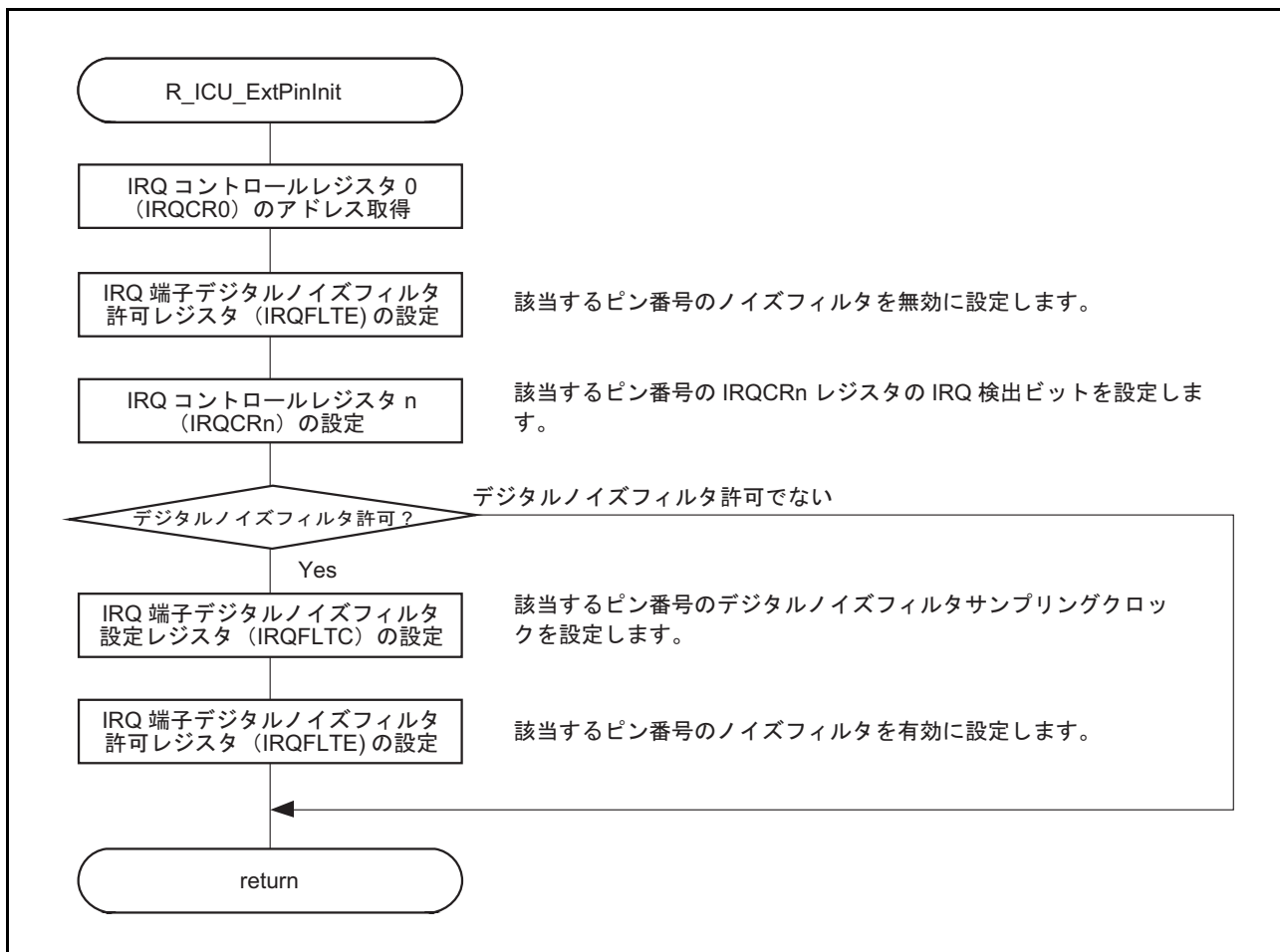


図 5.26 外部割り込み端子設定処理

### 5.10.19 割り込み登録設定処理

図 5.27 に割り込み登録設定処理のフローチャートを示します。



図 5.27 割り込み登録設定処理

### 5.10.20 シリアルフラッシュメモリ内レジスタ設定処理（ユーザ定義）

図 5.28 にシリアルフラッシュメモリ内レジスタ設定処理（ユーザ定義）のフローチャートを示します。

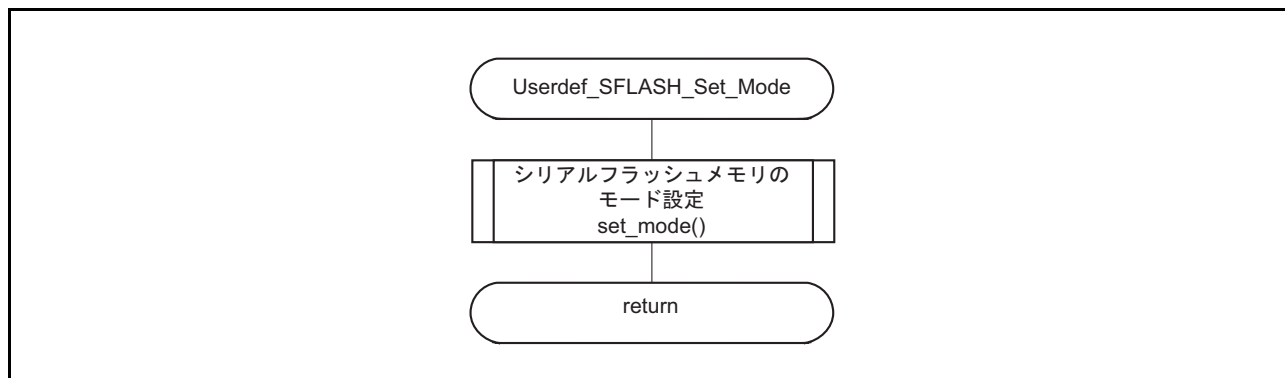


図 5.28 シリアルフラッシュメモリ内レジスタ設定処理（ユーザ定義）

### 5.10.21 シリアルフラッシュメモリライト許可設定処理（ユーザ定義）

図 5.29 にシリアルフラッシュメモリライト許可設定処理（ユーザ定義）のフローチャートを示します。

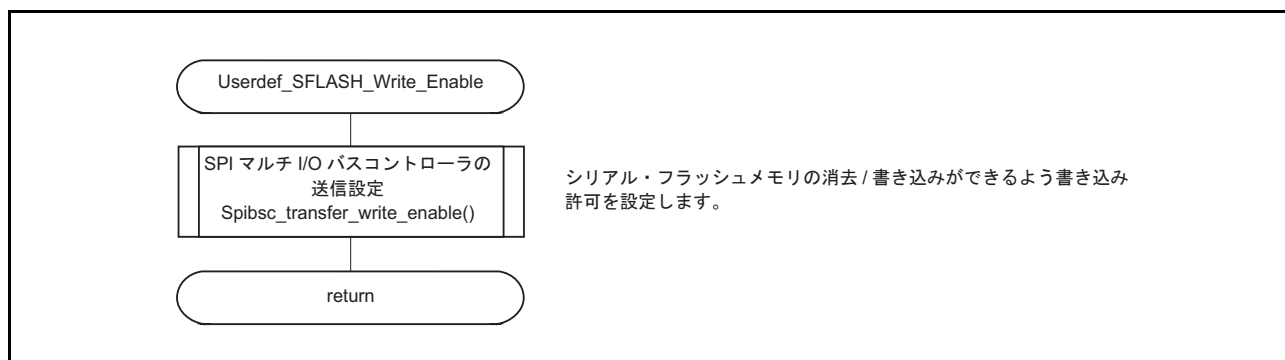


図 5.29 シリアルフラッシュメモリライト許可設定処理（ユーザ定義）

### 5.10.22 シリアルフラッシュメモリレディー待ち処理（ユーザ定義）

図 5.30 にシリアルフラッシュメモリレディー待ち処理（ユーザ定義）のフローチャートを示します。

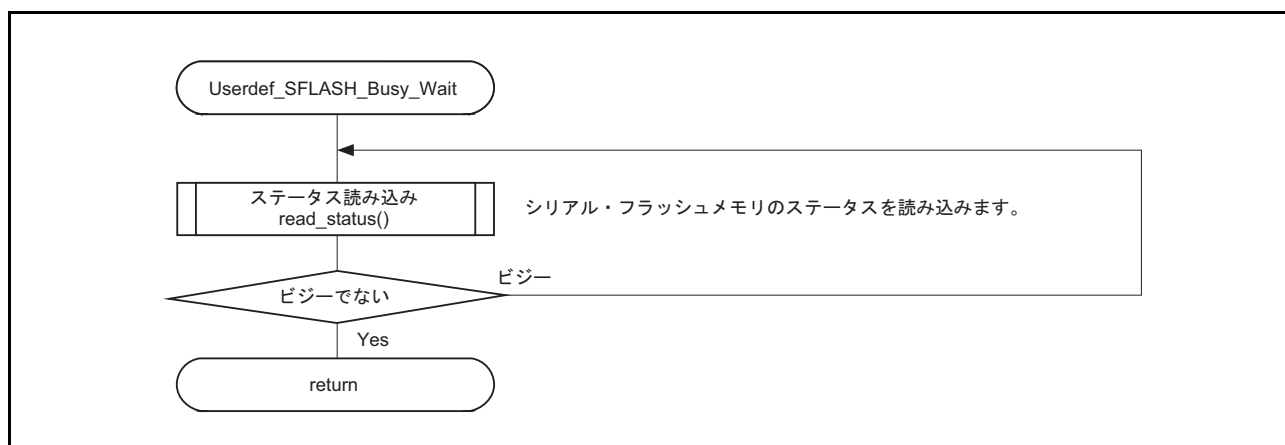


図 5.30 シリアルフラッシュメモリレディー待ち処理（ユーザ定義）

### 5.10.23 SPIBSC 外部アドレスモード初期設定処理

図 5.31 に SPIBSC 外部アドレスモード初期設定処理のフローチャートを示します。

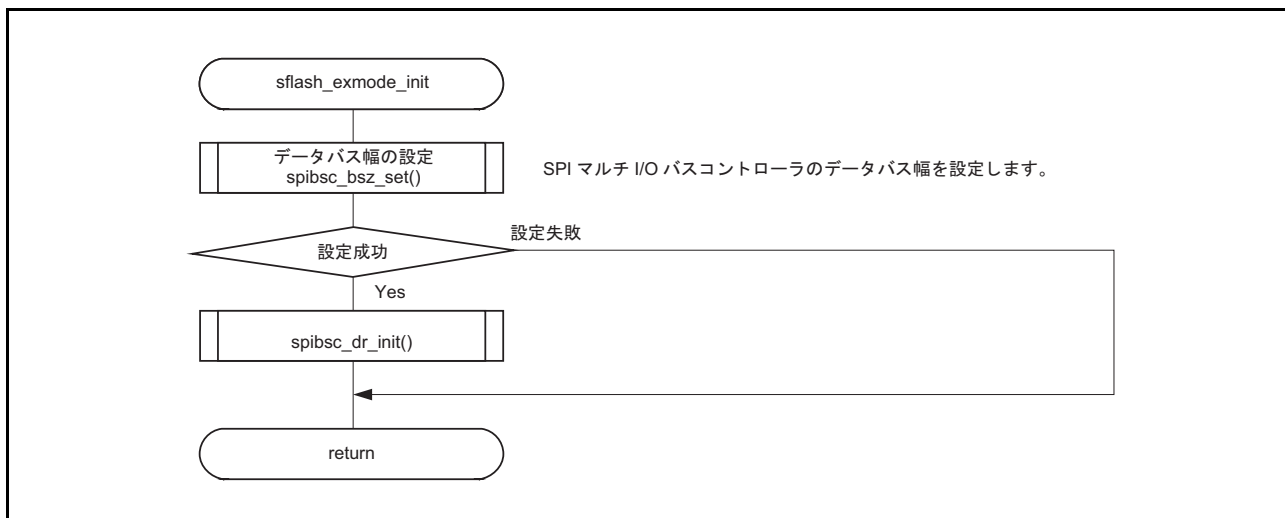


図 5.31 SPIBSC 外部アドレスモード初期設定処理

### 5.10.24 SPIBSC 初期設定処理

図 5.32 に SPIBSC 初期設定処理のフローチャートを示します。

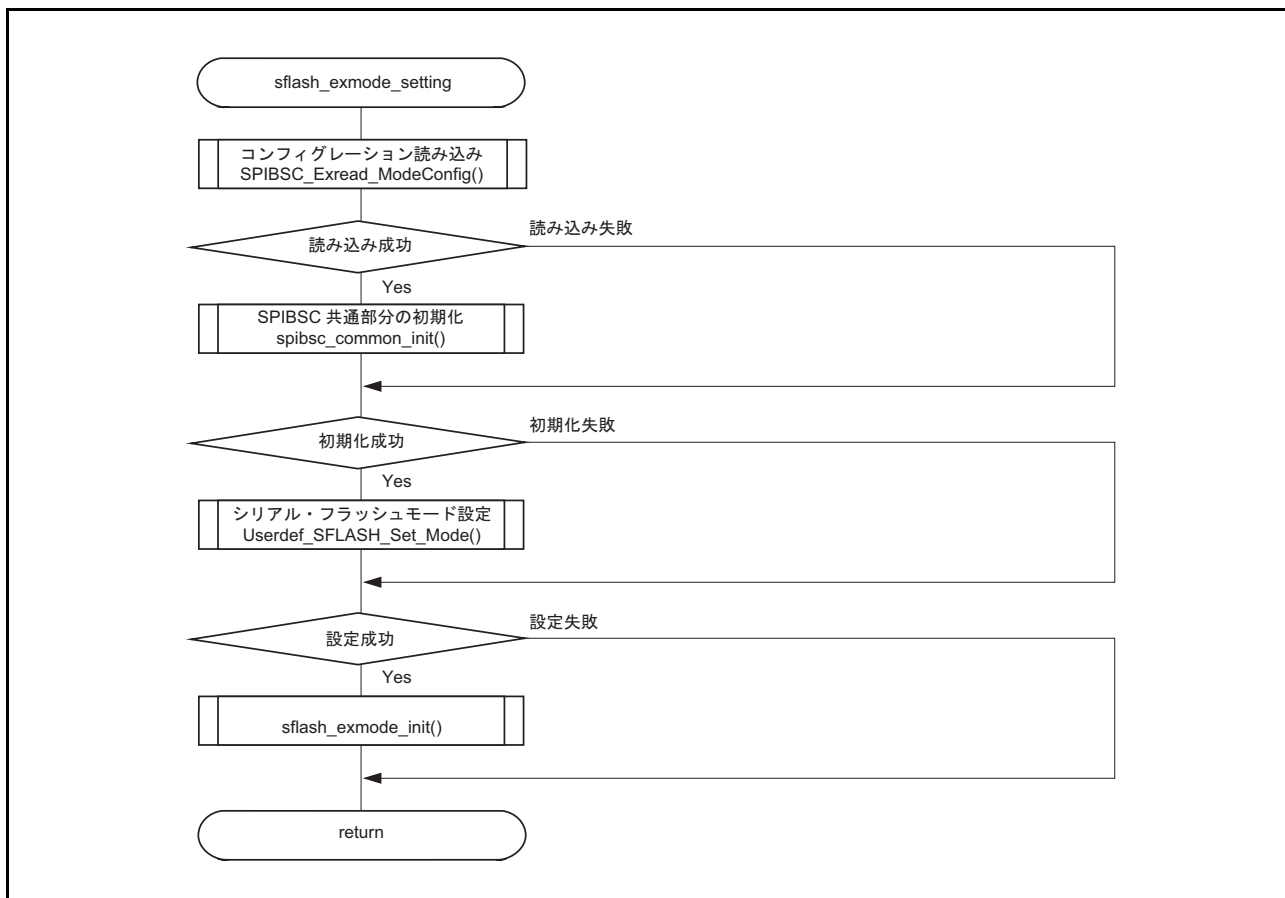


図 5.32 SPIBSC 初期設定処理



### 5.10.25 SPIBSC データ転送終了待ち処理

図 5.33 に SPIBSC データ転送待ち処理のフローチャートを示します。

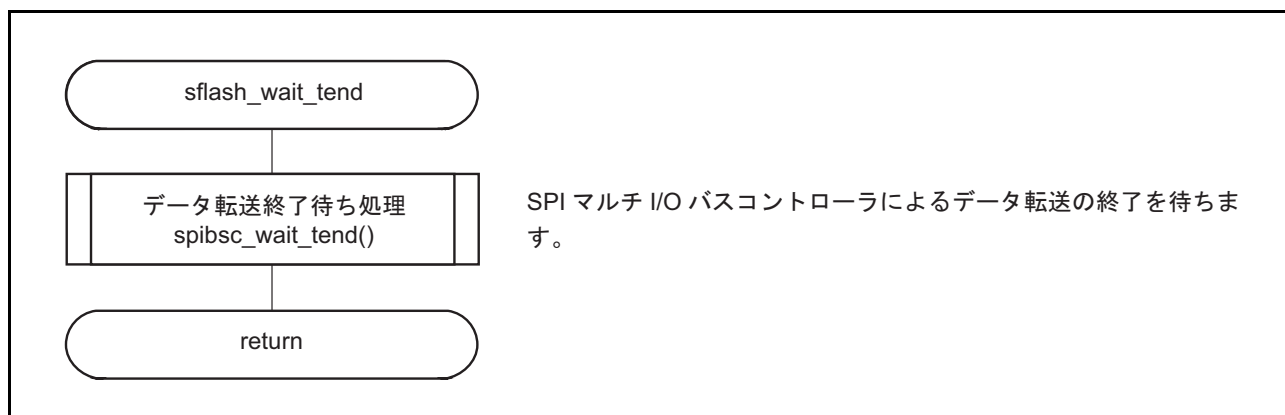


図 5.33 SPIBSC データ転送待ち処理

### 5.10.26 SPIBSC 外部アドレスリードモード設定処理

図 5.34 に SPIBSC 外部アドレスリードモード設定処理のフローチャートを示します。

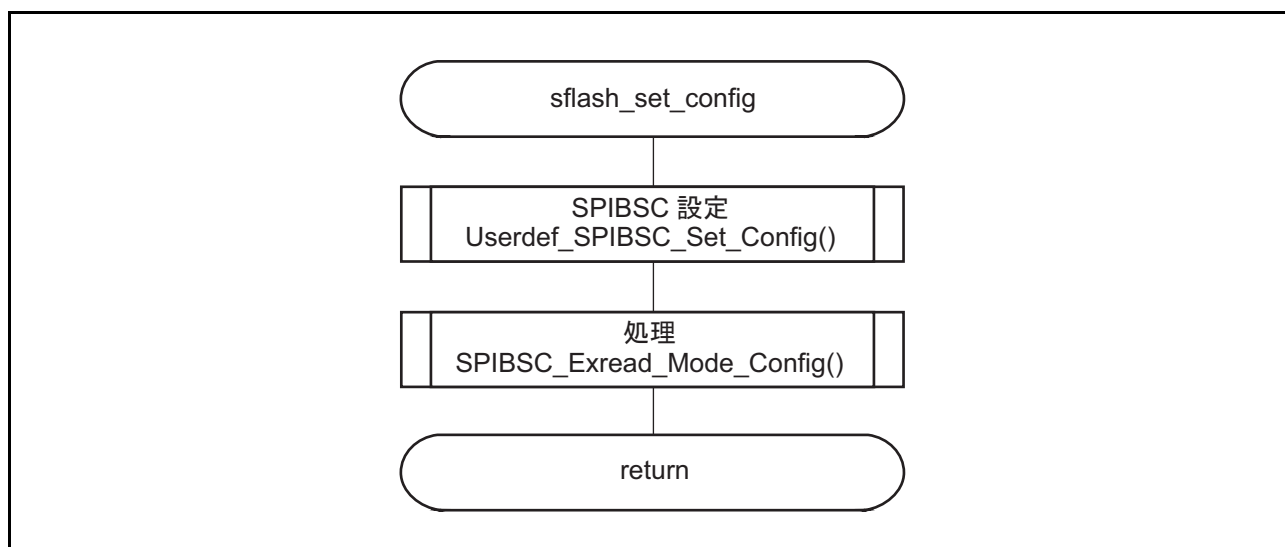


図 5.34 SPIBSC 外部アドレスリードモード設定処理

### 5.10.27 SPIBSC コンフィギュレーション処理

図 5.35 に SPIBSC コンフィギュレーション処理のフローチャートを示します。

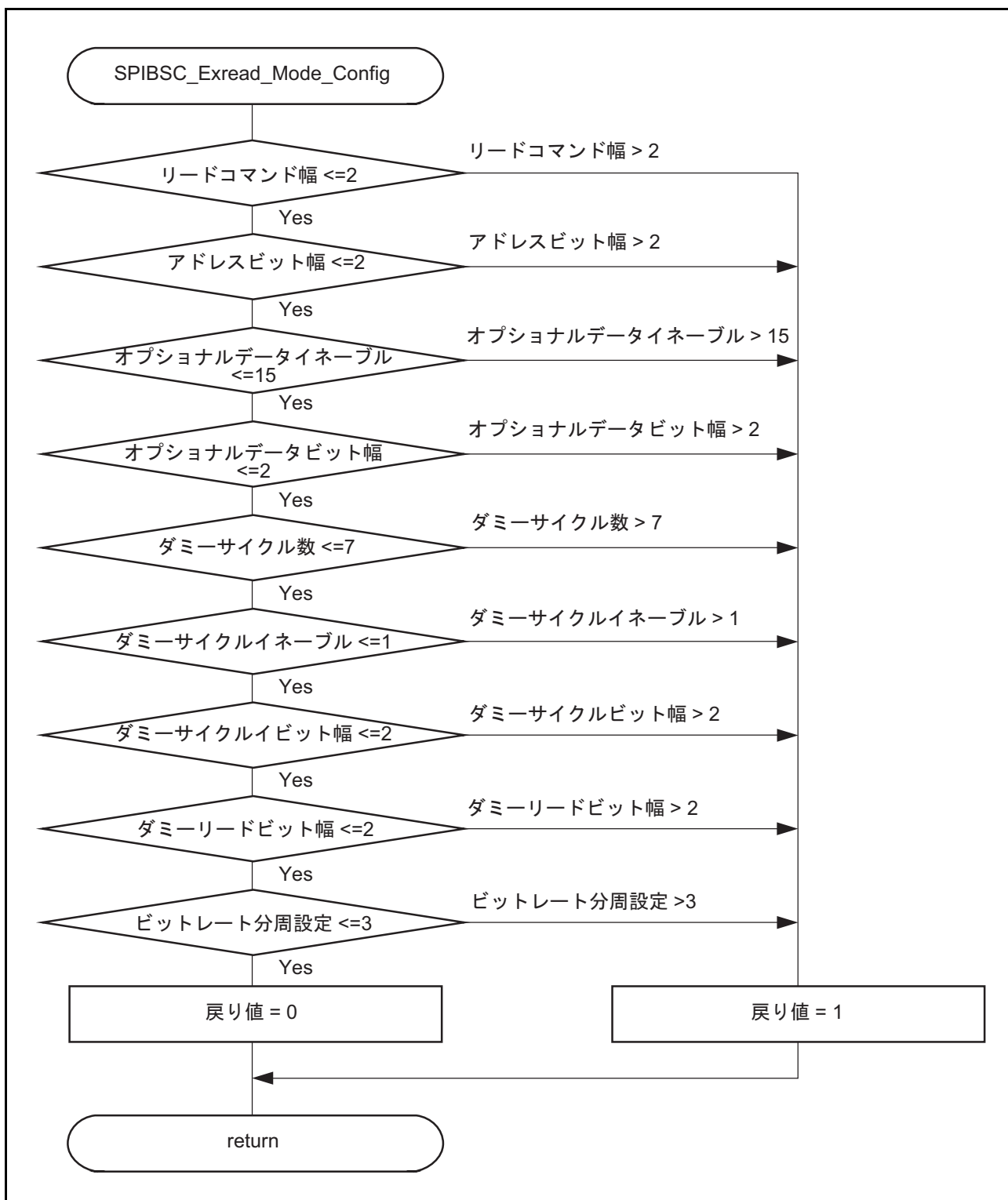


図 5.35 SPIBSC コンフィギュレーション処理

## 5.10.28 SPIBSC 外部アドレスリード設定処理（ユーザ定義）

図 5.36 に SPIBSC 外部アドレスリード設定処理（ユーザ定義）のフローチャートを示します。

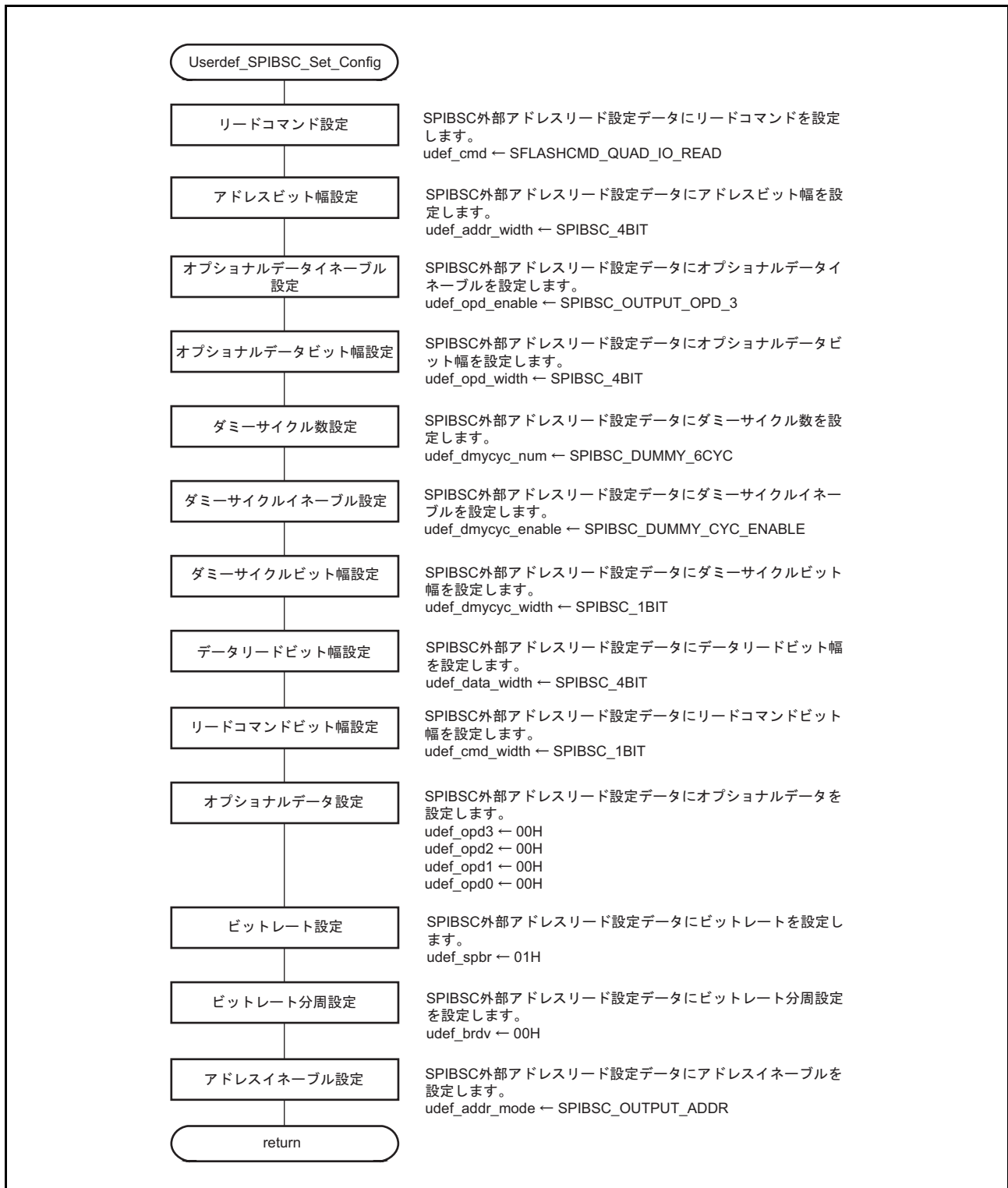


図 5.36 SPIBSC 外部アドレスリード設定処理（ユーザ定義）

## 5.10.29 IRQ9 割り込み（IRQ 端子割り込み 5）処理

図 5.37 に IRQ9 割り込み（IRQ 端子割り込み 5）処理のフローチャートを示します。

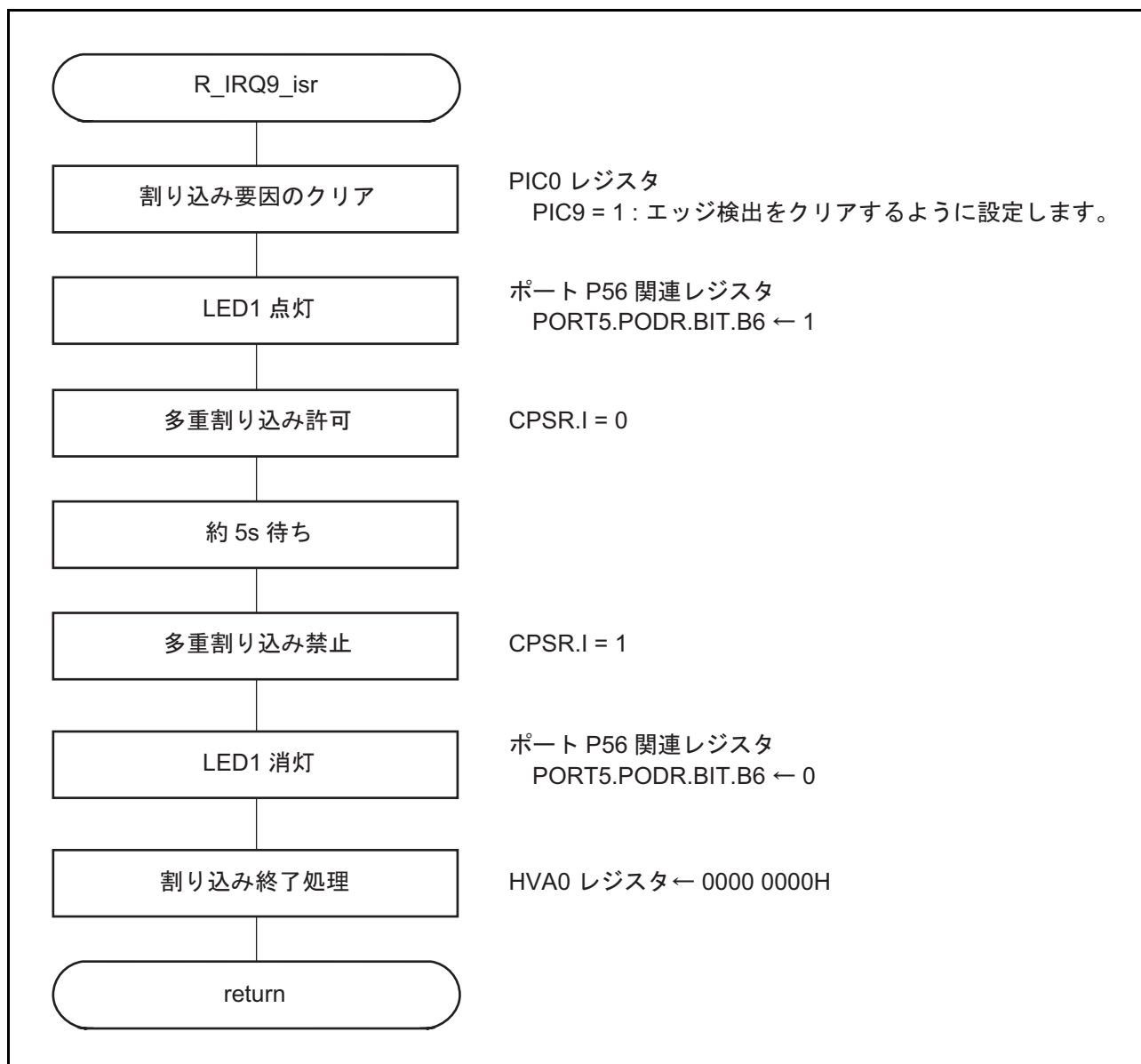


図 5.37 IRQ9 割り込み（IRQ 端子割り込み 5）処理

### 5.10.30 IRQ16 割り込み (IRQ 端子割り込み 12) 処理

図 5.38 に IRQ16 割り込み (IRQ 端子割り込み 12) 処理のフローチャートを示します。

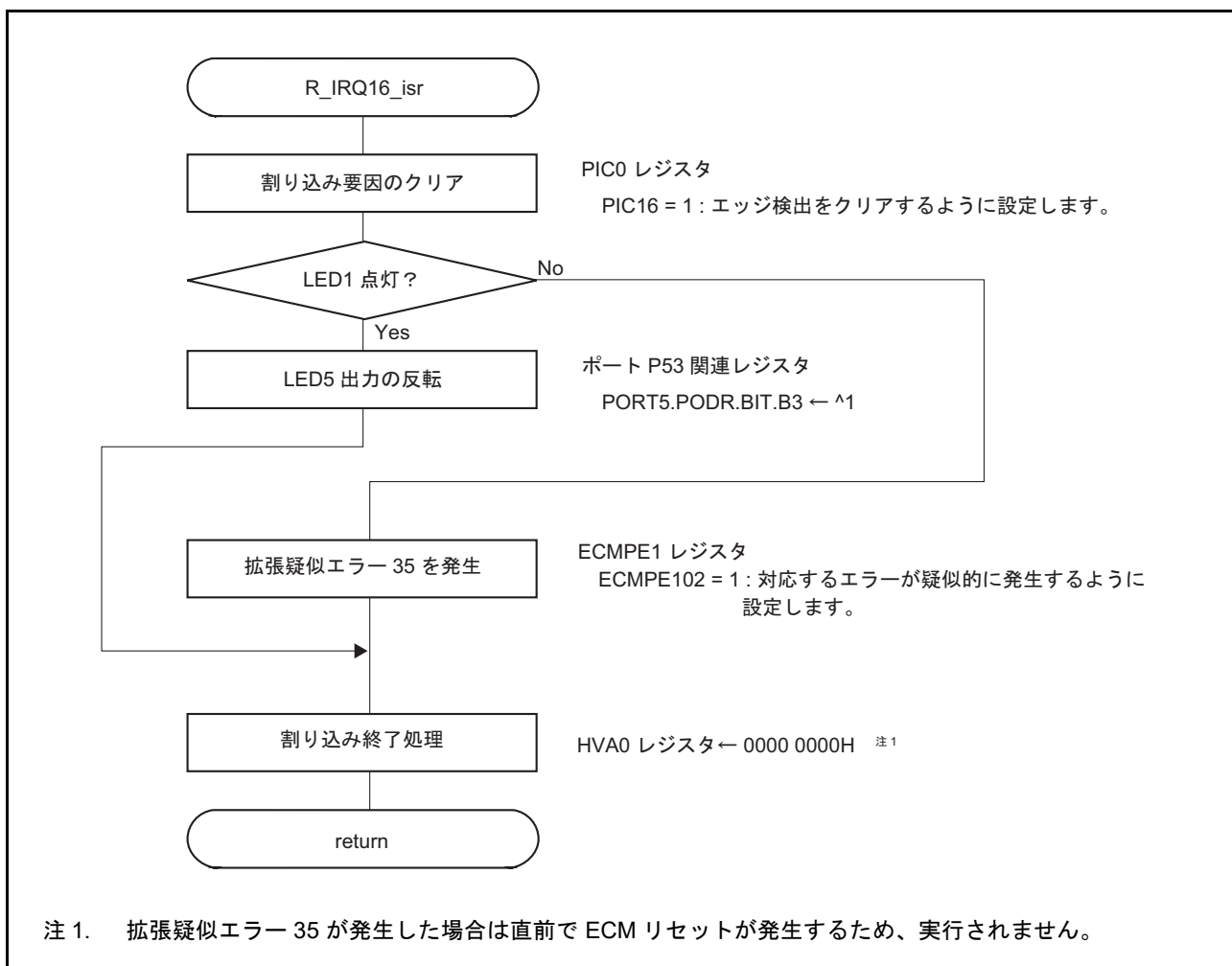


図 5.38 IRQ16 割り込み (IRQ 端子割り込み 12) 処理

## 6. サンプルプログラム

サンプルプログラムは、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

- ユーザーズマニュアル：ハードウェア

RZ/T1 グループ ユーザーズマニュアルハードウェア編

(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。)

RZ/T1 Evaluation Board RTK7910022C00000BR ユーザーズマニュアル

(最新版をルネサス エレクトロニクスホームページから入手してください。)

- テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

- ユーザーズマニュアル：開発環境

IAR 統合開発環境 (IAR Embedded Workbench® for Arm) に関しては、最新版を IAR ホームページから入手してください。

Arm 統合開発環境 (Development Studio 5™) に関しては、最新版を Arm ホームページから入手してください。

ルネサス エレクトロニクス統合開発環境 (e2 studio) に関しては、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。

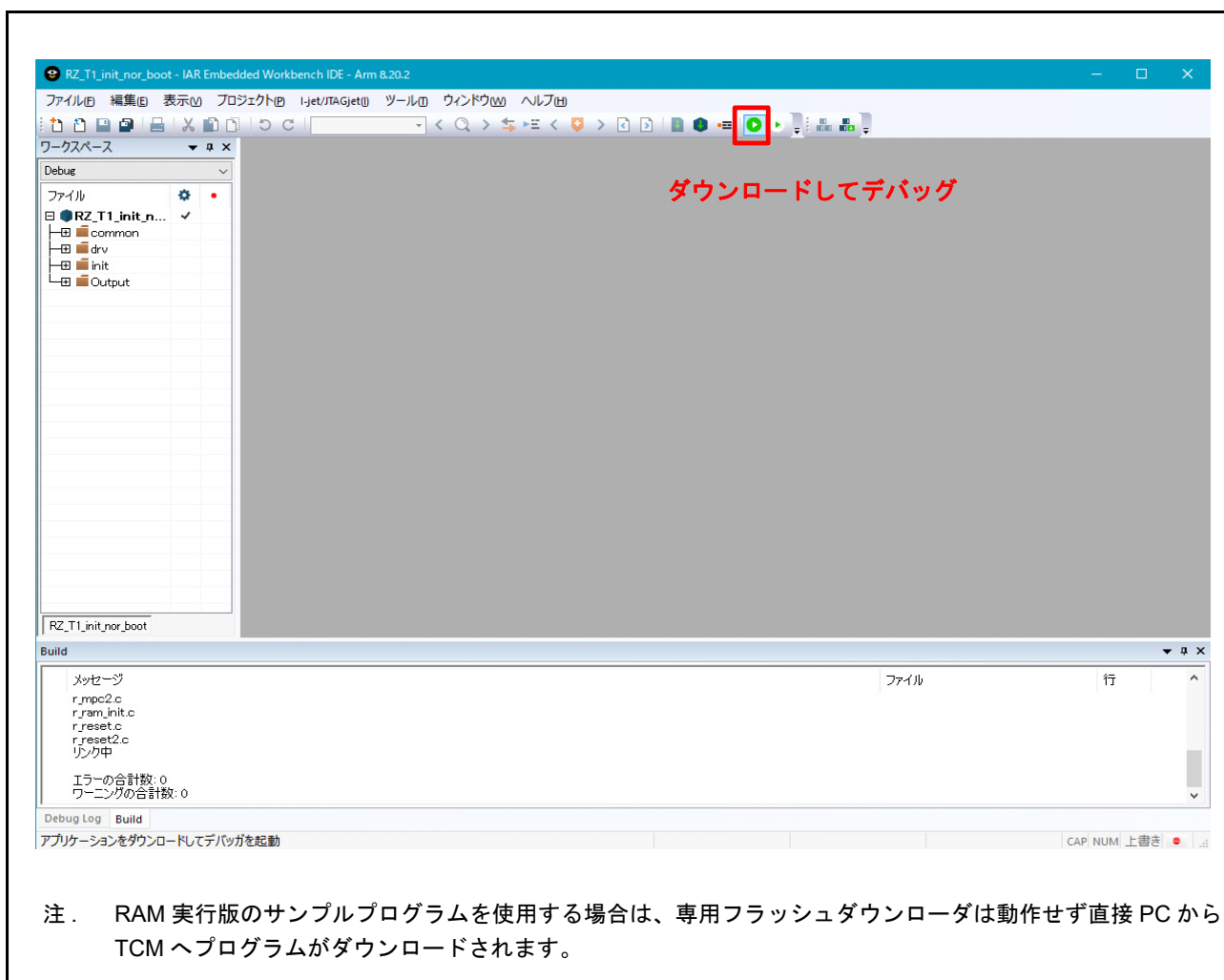
## 付録 1. 各開発環境における補足内容

ここでは、本サンプルプログラムを各開発環境で使用する際のデバッグまでの手順を説明します。また、ソースコードを異なる開発環境へ移植する際の注意点についても説明します。

### ■サンプルプログラムのデバッグ手順（SPI ブートの場合）

#### （EWARM : IAR 社製）

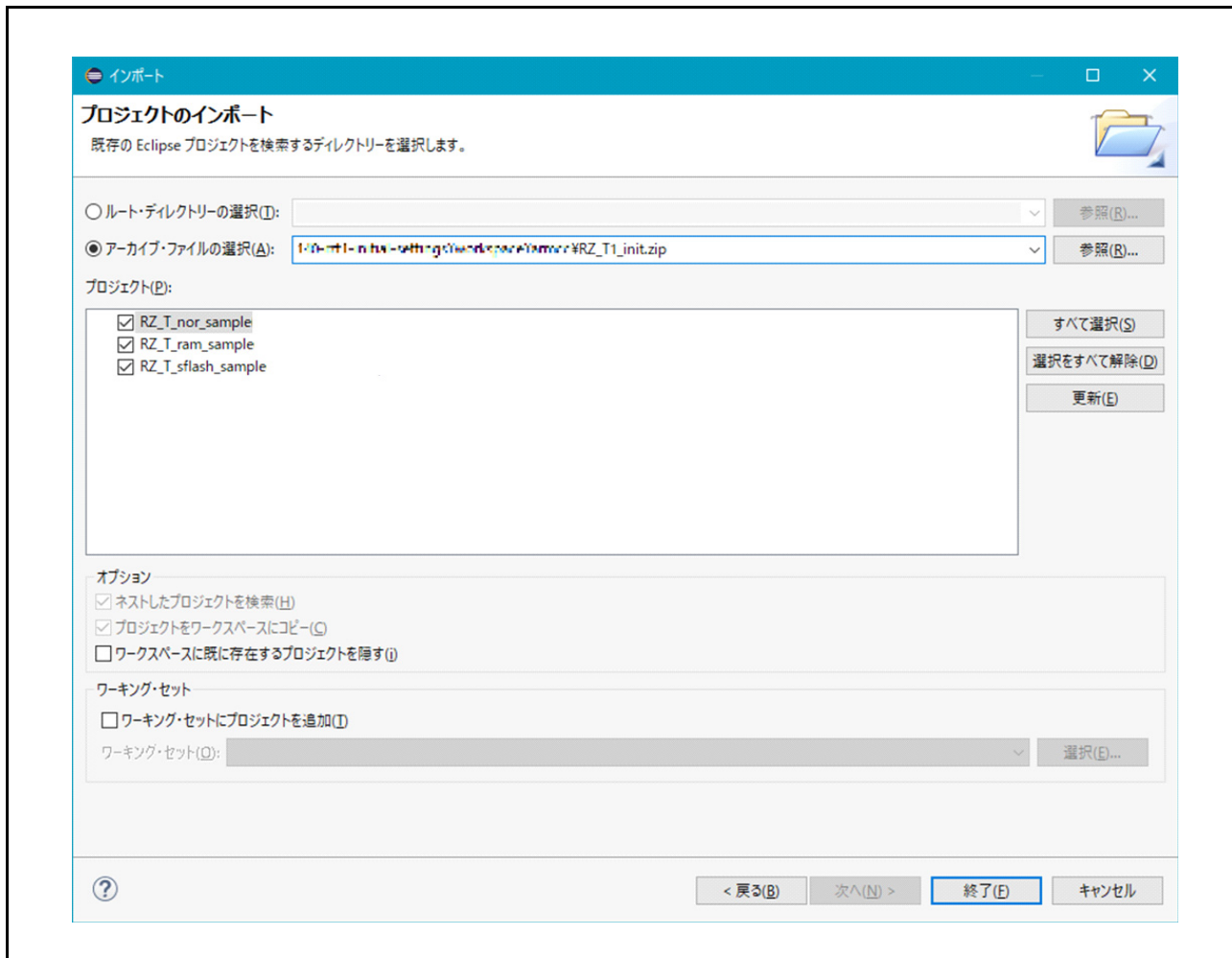
- (1) EWARM を起動し、[ファイル]→[開く]→[ワークスペース]で RZ\_T1\_init\_serial\_boot.eww を指定します。
- (2) [プロジェクト]→[すべてを再ビルド]を実行します。
- (3) RZ/T1 評価ボードと I-jet を接続した状態で、[プロジェクト]→[ダウンロードしてデバッグ]を選択します。
- (4) エミュレータ接続後、専用フラッシュダウンローダ機能により外付けシリアルフラッシュメモリへプログラムの書き込みが行われた後に、デバッグが開始されます。





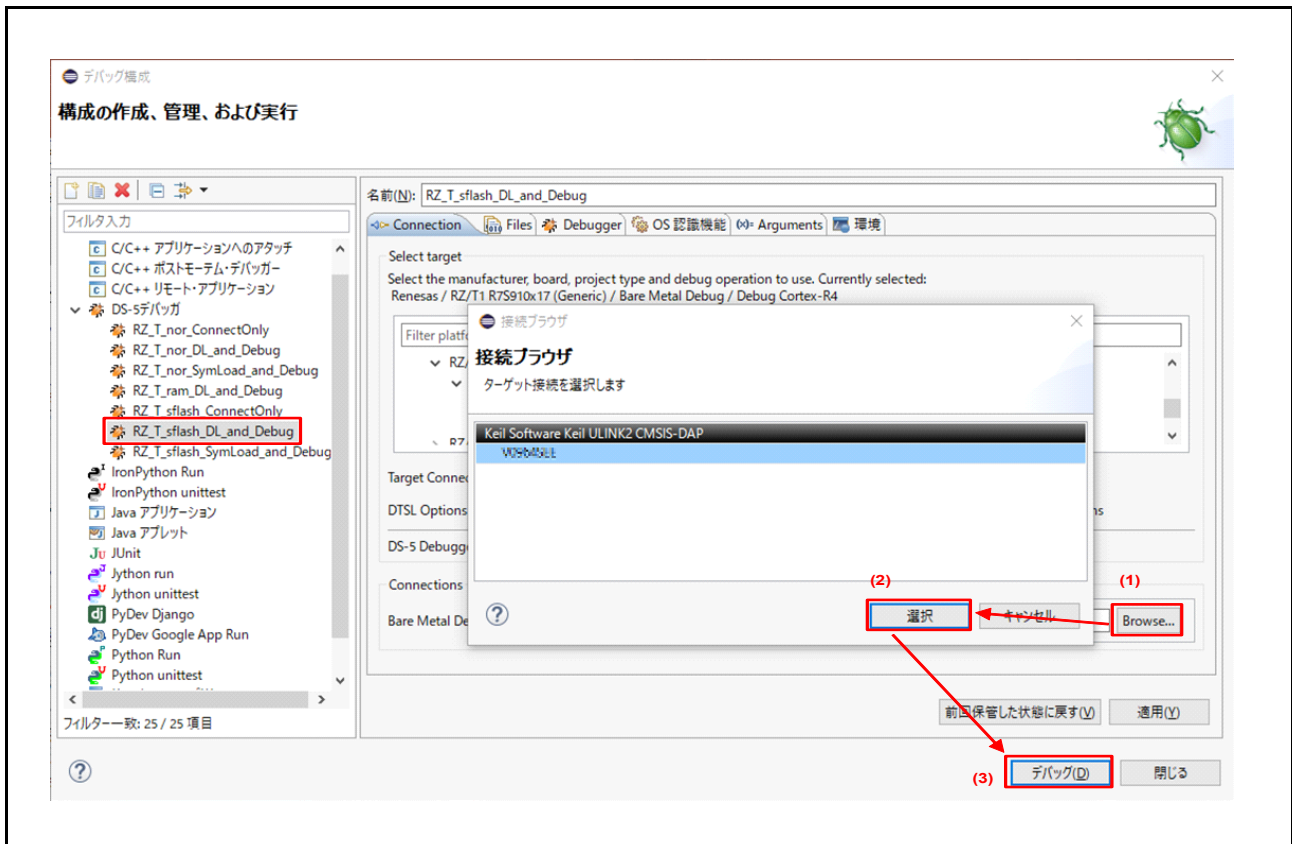
**(DS-5 : Arm 社製)**

- (1) DS-5 を起動しワークスペースへ移動後、[ファイル]→[インポート]をクリックし、一般>既存プロジェクトをワークスペースへ を選択して[次へ]をクリックします。
- (2) プロジェクトのインポート画面でアーカイブ・ファイルの選択にて RZ\_T1\_initial.zip を指定して、[終了]をクリックします。



- (3) [プロジェクト]→[すべてをビルド]を実行します。

- (4) “RZ\_T\_sflash\_DL\_and\_Debug” の接続設定を選択し、RZ/T1 評価ボードと ULINK2 を接続した状態で、下記画面の [ 参照 ] を選択し接続ブラウザを開きます。検出されたターゲット接続を選択後、デバッグ構成画面の [ デバッグ ] を選択しデバッグを開始します。



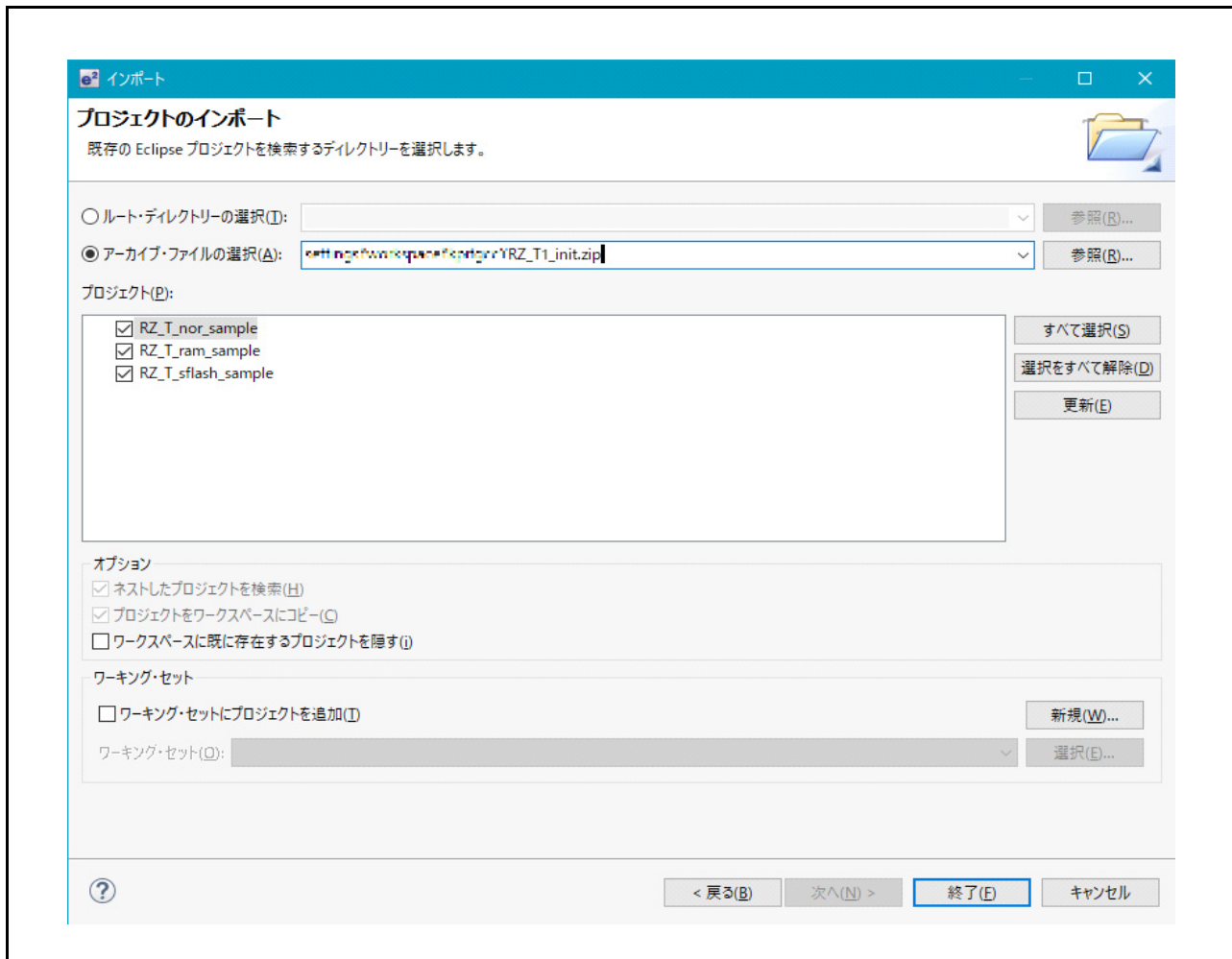
- (5) エミュレータ接続後、スクリプトファイル RZ\_T\_sflash\_sample.ds が実行されセミホスティング機能を使用した外付けシリアルフラッシュメモリへのプログラム書き込みが行われます。セミホスティング機能を使用したフラッシュメモリへの書き込み機能については以下のアプリケーションノートを参照してください。
- RZ/T1 グループ Arm® Development Studio 5 (DS-5™) のセミホスティング機能を使用した NOR 型フラッシュメモリへのダウンロード例
  - RZ/T1 グループ Arm® Development Studio 5 (DS-5™) のセミホスティング機能を使用したシリアルフラッシュメモリへのダウンロード例
- (6) フラッシュ書き込みが完了すると、アプリケーションコンソール画面に「Flash Programming Complete」が表示され、デバッグを開始できます。

#### 備考

一度外付けシリアルフラッシュへ書き込み後、プログラムに変更がない場合は、(4) で “RZ\_T\_sflash\_SymLoad\_and\_Debug” を選択することで以降のデバッグ接続にフラッシュメモリへの書き込みを省略してシンボル情報のダウンロードのみを行いデバッグ開始することが可能です。

**(e2 studio : RENESAS 社製)**

- (1) e2studio を起動しワークスペースへ移動後、[ファイル]→[インポート]をクリックし、一般>既存プロジェクトをワークスペースへ を選択して [次へ]をクリックします。
- (2) プロジェクトのインポート画面でアーカイブ・ファイルの選択にて RZ\_T1\_initial.zip を指定して、[終了]をクリックします。



- (3) [プロジェクト]→[すべてをビルド]を実行します。

- (4) “RZ\_T\_sflash\_sample\_HardwareDebug” の接続設定を選択し、RZ/T1 評価ボードと J-Link を接続した状態で、下記画面の [ デバッグ ] を選択しデバッグを開始します。



- (5) エミュレータ接続後、専用フラッシュダウンローダ機能により外付けシリアルフラッシュメモリへプログラムの書き込みが行われた後に、デバッグが開始されます。

## ■ ソースコード移植時の注意点

ソースコードを異なる開発環境向けに移植する際の注意点を、本サンプルプログラムを例に説明します。

注． 以下ではサンプルプログラムで使用している一例について説明しています。詳細な設定方法や拡張設定などについては各開発環境のマニュアルなどをご参照ください。

### (1) 割り込み関数の指定子（IRQ 割り込み関数の指定例）

EWARM : IAR社製	DS-5 : Arm社製	e2 studio : RENESAS製
#pragma type_attribute=__irq__arm void R_IRQ_isr (void);	__irq void R_IRQ_isr (void);	void R_IRQ_isr (void) __attribute__((interrupt));

### (2) インラインアセンブラの記述（nop 命令の記述例）

EWARM : IAR社製	DS-5 : Arm社製	e2 studio : RENESAS製
asm volatile ("nop");	__asm("nop");	asm("nop");

### (3) ソースコード内でのセクションの指定（ローダ用パラメータの指定例）

EWARM : IAR社製	DS-5 : Arm社製	e2 studio : RENESAS製
#pragma location="ldr_param" __root const uint32_t loader_param_SPI[19] = {	#pragma arm section rodata = "CONST_LOADER_TABLE" const stBOOT_SPIBSC spibsc_loader_table = {	const unsigned int Loader_Param[] __attribute__((section(".loader_param"))) = {

注． セクション配置については、各開発環境のリンカ設定も参照してください。

## 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/>

お問合せ先

<http://japan.renesas.com/contact/>

改訂記録	初期設定 アプリケーションノート
------	------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
0.10	2015.02.03	—	初版発行
0.20	2015.03.10	19	5.2.4 必要メモリサイズ 表5.6 必要メモリサイズ 修正
		20	5.4 固定幅整数一覧 表5.8 サンプルプログラムで使用する固定幅整数 修正
1.00	2015.04.10	—	Web掲載に際しRevのみ変更
1.10	2015.08.31	2. 動作環境	
		6	表2.1 動作環境 統合開発環境 表記一部修正、追加
		5. ソフトウェア説明	
		11	説明文追加
		12	5.1.1 プロジェクト設定 表5.1 修正
		13	5.1.2 説明文追加
		17	5.2.2 MPUの設定 表5.4 修正
		19	5.2.4 説明文 参照を追加
		19	表5.6 タイトル一部追加
		20	表5.7 追加
		21	表5.8 追加
		7. 参考ドキュメント	
		65	ユーザーズマニュアル：開発環境 内容追加
		付録 1. 各開発環境における補足内容	
66 ~ 73	付録 1. 各開発環境における補足内容 追加		
1.20	2015.12.03	2. 動作環境	
		6	表2.1 動作環境 統合開発環境 一部修正
		5. ソフトウェア説明	
		50	5.10.12 ICU 設定処理 図 5.20 一部追加
		付録 1. 各開発環境における補足内容	
67 ~ 70	■サンプルプログラムのデバッグ手順 (SPI ブートの場合) DS-5 : ARM 社製、e2 studio : RENESAS 社製 内容修正		
1.30	2017.05.30	2. 動作環境	
		6	表 2.1 動作環境 統合開発環境の内容を変更
		4. ハードウェア説明	
		10	表 4.1 使用端子と機能 P53 端子を追加
		5. ソフトウェア説明	
		11	5.1 動作概要 本文を変更
		11	図 5.1 ブート処理後の概要動作 処理を変更、注 3. を追加
		—	5.2.4 必要メモリサイズ 削除
		19	表 5.6 サンプルプログラムで使用する割り込み IRQ 端子割り込み 5(IRQ9) の処理概要を変更、IRQ 端子割り込み 12(IRQ16) の優先度および処理概要を変更
		24	表 5.10 関数一覧 関数名を変更 (atcm_waitset, cpg_pll_wait, sflash_exmode_init, sflash_exmode_setting, sflash_wait_tend, sflash_set_config)
		27	5.9.10 port_init 説明を変更
		28	5.9.13 atcm_waitset 関数名を変更
		29	5.9.14 cpg_pll_wait 関数名を変更
		32	5.9.23 sflash_exmode_init 関数名を変更
		33	5.9.24 sflash_exmode_setting 関数名を変更
		33	5.9.25 sflash_wait_tend 関数名を変更
		34	5.9.26 sflash_set_config 関数名を変更
		35	5.9.29 R_IRQ9_isr 説明を変更
		35	5.9.30 R_IRQ16_isr 説明および補足を変更
		38	図 5.8 メイン処理 P53 を追加
		39	図 5.9 リセットフラグ判定処理 関数名を変更
		40	図 5.10 リセットフラグ判定処理 関数名を変更
		41	図 5.11 クロック設定処理 関数名を変更
		42	図 5.12 バス設定処理 (1) 関数名を変更
		46	図 5.18 ポート設定処理 処理を追加 定義済み処理を変更
		48	図 5.21 ATCM アクセスウエイト設定処理 関数名を変更
		49	図 5.22 PLL 安定待ち (100μs) 処理 関数名を変更

## 改訂記録

## 初期設定 アプリケーションノート

Rev.	発行日	改訂内容	
		ページ	ポイント
1.30	2017.05.30	50	図 5.23 ECM 初期設定処理 ECM エラーマスクレジスタ設定を変更
		56	図 5.31 SPIBSC 外部アドレスモード初期設定処理 関数名を変更
		56	図 5.32 SPIBSC 初期設定処理 関数名を変更
		57	図 5.33 SPIBSC データ転送待ち処理 関数名を変更
		57	図 5.34 SPIBSC 外部アドレスリードモード設定処理 関数名を変更
		60	図 5.37 IRQ9 割り込み (IRQ 端子割り込み 5) 処理 処理を追加
		61	図 5.38 IRQ16 割り込み (IRQ 端子割り込み 12) 処理 判断および処理を追加、注を追加
1.40	2018.04.27	全般	ARM → Arm に変更
		2. 動作環境	
		6	表 2.1 動作環境 IAR システムズ製、Arm 製、RENESAS 製の統合開発環境のバージョンを変更
		付録 1. 各開発環境における補足内容	
		64	■サンプルプログラムのデバッグ手順 (EWARM : IAR 社製) (4) : 図を変更
		65	■サンプルプログラムのデバッグ手順 (DS-5 : Arm 社製) (2) : 図を変更
		66	■サンプルプログラムのデバッグ手順 (DS-5 : Arm 社製) (4) : 図を変更
		67	■サンプルプログラムのデバッグ手順 (e2 studio : RENESAS 社製) (2) : 図を変更
		68	■サンプルプログラムのデバッグ手順 (e2 studio : RENESAS 社製) (4) : 図を変更
		69	■ソースコード移植時の注意点 (2) : EWARM : IAR 社製のインラインアセンブラの記述を変更
1.41	2018.07.13	5. ソフトウェア説明	
		36	図 5.6 ローダプログラム 1 処理 処理 : 初期値なし変数の初期化 追加
		37	図 5.7 ローダプログラム 2 処理 処理 : 初期値なし変数の初期化 追加

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。



## 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）がありません。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っていると、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。型名が異なる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生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
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）、交通制御（信号）、大規模通信機器、  
金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
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、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)



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：<https://www.renesas.com/contact/>