

SH7734 グループ

R01AN0665JJ0101

SH7734 初期設定例

Rev.1.01

2012.07.18

要旨

本アプリケーションノートでは、パワーオンリセット時の CS0 ブートで必要な設定項目の例について説明します。

SH7734 のブートについて

- 起動時のモード設定によって、CS0 空間に接続されたメモリ、NAND フラッシュメモリ、シリアルフラッシュメモリ、MMC 接続のフラッシュメモリ、eSD デバイスからのブート、および HIF ブートによるブートが選択可能です。
- 本アプリケーションノートでは、CS0 空間に接続されたメモリ (NOR フラッシュ) によるブートモード選択時の初期設定例について説明します。

本書の位置づけ

本アプリケーションノートは、同 SH-4A 製品である、「SH7730 初期設定例 (RJJ06B0864)」のアプリケーションノートを基に作成しています。SH-4A コアの SH マイコンを初めて使用される方や、関連する基礎知識について確認したい方は、あらかじめ、「SH7730 初期設定例 (RJJ06B0864)」のアプリケーションノートを参照いただくことを推奨します。

動作確認デバイス

SH7734 グループ (R8A77343)

製品型名の詳細については、「SH7734 グループ ユーザーズマニュアル ハードウェア編 (R01UH0233JJ)」を参照ください。

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、そのマイコンの使用にあわせて変更し、十分評価してください。

目次

1. 仕様	2
2. 動作確認条件	4
3. 関連アプリケーションノート	4
4. 周辺機能説明	5
5. ハードウェア説明	6
6. ソフトウェア説明	8
7. 応用例	15
8. サンプルコード	26
9. 参考ドキュメント	26

1. 仕様

リセット解除後に、クロック発振器 (CPG)、ローカルバスステートコントローラ (LBSC)、メモリコントローラ (DBSC3) およびキャッシュの初期設定を行います。

また、「7. 応用例」で、SH7734 評価用プラットフォーム (R0P7734C00000RZ) でのデバッグ用サンプル (LED、デバッグシリアル、タイマ、キーマトリクスインタフェース) について紹介します。

SH7734 の各アプリケーションノートでは、本アプリケーションノートで説明する参考プログラムを初期設定プログラムとして使用することを前提としています。

表 1.1 に本アプリケーションノートで使用する周辺機能と用途を、図 1.1 にパワーオンリセットからの処理フローを示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
CPG	クロックモード設定 CPG 設定は、外部端子の設定に依存します。詳細は、「SH7734 ユーザーズマニュアルハードウェア編 (R01UH0233JJ) クロック発振器 (CPG)」の章を参照ください。
FPU	浮動少数点モード設定
LBSC	エリア 0 : NOR 型フラッシュメモリ接続 (JS28F512M29EWLA (Numonyx) x 1) データバス幅 : 16 ビット 標準 (SRAM) のインタフェース選択 詳細は、「SH7734 ユーザーズマニュアルハードウェア編 (R01UH0233JJ)」ローカルバスステートコントローラ (LBSC) の章を参照ください。
DBSC3	エリア 2、3 : DDR2-SDRAM 接続 (MT47H64M16HR-3 (Micron) x 1) データバス幅 : 16 ビット 詳細は、「SH7734 ユーザーズマニュアルハードウェア編 (R01UH0233JJ)」メモリコントローラ (DBSC3) の章を参照ください。
キャッシュ	命令キャッシュ有効 オペランドキャッシュ有効
GPIO (デバッグ用途)	ユーザー開放用 LED 制御
SCIF (デバッグ用途)	デバッグシリアルインタフェース制御 ターミナルソフトを接続した、puts 関数や printf 関数など標準入出力
TMU (デバッグ用途)	タイマ制御
MTU2、ADC (デバッグ用途)	キーマトリクスインタフェース制御

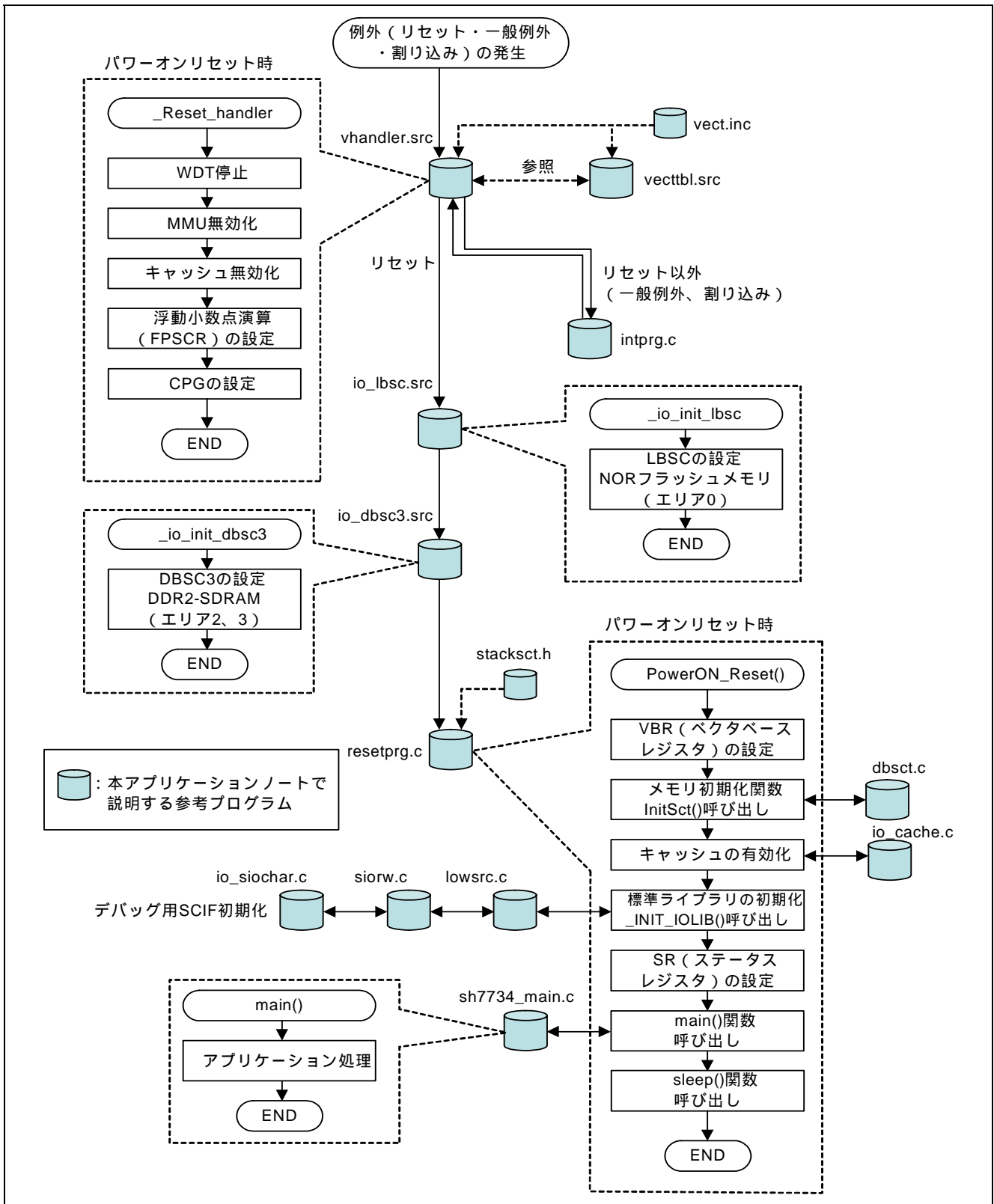


図1.1 パワーオンリセットからの処理フロー

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	SH7734 (R8A77343)
動作周波数	EXTAL 入力周波数: 33.3333MHz CPU クロック (clki) : 400MHz SHwy クロック (clks) : 200MHz SHwy クロック (clks1) : 100MHz DDR クロック (MCK0/MCK0#/MCK1/MCK1#) : 200MHz バスクロック (clk_b) : 50MHz 周辺クロック (clk_p) : 50MHz
動作電圧	IO supply power (3.3V) Core supply power (1.25V)
統合開発環境	ルネサス エレクトロニクス製 High-performance Embedded Workshop (Version 4.08.00.011)
C コンパイラ	ルネサス エレクトロニクス製 C/C++ Compiler Package for SuperH Family (V.9.04 release00) コンパイルオプション -cpu=sh4a -endian=little -include="\$ (PROJDIR)¥inc" -change_message=warning -object="\$ (CONFIGDIR)¥\$ (FILELEAF).obj" -debug -optimize=0 -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo
サンプルコードのバージョン	Ver 1.01
エンディアン	リトルエンディアン
処理モード	特権モードのみで動作
ブートモード	CS0 ブートモード
アドレス拡張モード	29 ビット
メモリマネジメントユニット (MMU)	ディスエーブル
ウォッチドッグタイマ (WDT)	ディスエーブル
使用ボード	ルネサス エレクトロニクス社製 SH7734 評価用プラットフォーム (R0P7734C00000RZ)

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。

- SH7730 グループ SH7730 初期設定例 (RJJ06B0864)
- SH7730 グループ SH7730 TMU 設定例 (RJJ06B0997)
- SH7730 グループ SH7730 SCIF 調歩同期式 送受信設定例 (RJJ06B0954)
- SH7730 グループ SH7730 ADC 設定例 (スキャンモード) (RJJ06B1088)

【注】 動作周波数は異なりますので、ご注意ください。

4. 周辺機能説明

本アプリケーションノートの動作モードについて補足します。詳細は、「SH7734 ユーザーズマニュアルハードウェア編 (R01UH0233JJ)」、「SH7734 評価用プラットフォーム ユーザーズマニュアル (R0P7734C00000RZ)」に記載しています。

表4.1 MD 端子と機能

端子名	入出力	設定	内容
MD0	入力	自走モード MD0=0	自走モード / ステップアップモード切り替え信号 SH7734 評価用プラットフォームでは設定値固定
MD1	入力	400MHz モード (2) MD0=0、MD1=0、MD2=0、 MD3=0	動作周波数を設定 MD4、MD3 とともに 0 固定のみサポート
MD2			
MD3			
MD4			
MD5	入力	16 ビットバス幅 MD5=0、MD6=1	エリア 0 のバス幅の設定 SH7734 評価用プラットフォームでは設定値固定
MD6			
MD7	入力	MD7=0、MD8=0 * ¹	エリア分割設定 SH7734 評価用プラットフォームでは設定値固定
MD9			
MD8	入力	リトルエンディアン MD8=1	ビッグ / リトルエンディアン設定
MD10	入力	EXTAL 端子に外部クロックを入力 MD10=0	水晶発振子 / 発振器の設定 SH7734 評価用プラットフォームでは設定値固定
MD11	入力	× 12 MD11=0、MD12=0	PLL1 の通倍率を設定
MD12			
MD13	入力	29 ビットアドレスモード MD13=0	29/32 ビットアドレスモード設定
MD14	入力	CS0 ブートモード MD14=0、MD16=0、 MD17=0、MD18=0、 MD19=0	ブートモード設定 * ² SH7734 評価用プラットフォームでは CS0 ブート、シリアルブート、NAND ブートに対応しています。ただし、NAND はコネクタのみ接続されています。
MD16			
MD17			
MD18			
MD19			
MD15	入力	MD15=0	MD15=0 のみサポート

【注】 *1 CPU から見た LBSC のサポートエリア、アドレスマップについては、「SH7734 ユーザーズマニュアルハードウェア編 (R01UH0233JJ)」ローカルバスステートコントローラ (LBSC) の章を参照ください。

*2 ブートモード設定については、「SH7734 ユーザーズマニュアルハードウェア編 (R01UH0233JJ)」動作モードの章を参照ください。

5. ハードウェア説明

5.1 使用端子一覧

表 5.1 に使用端子と機能を示します。

表5.1 使用端子と機能

モジュール	端子名	入出力	内容
LBSC	A[25:0] *	出力	アドレス出力
	D[15:0]	入出力	双方向データバス
	CS0#	出力	チップセレクト
	CKO	出力	システムクロック出力
	RD#	出力	リードストロープ
	RD/WR#	出力	リード/ライト
	WE1# ~ WE0#	出力	ライトイネーブル
	WAIT#	入力	外部からのウェイトサイクル要求入力
	BS#	出力	バスサイクルの開始を示す信号
	CS1#/A26	出力	本アプリケーションノートでは未使用
	EX_CS5~0#	出力	本アプリケーションノートでは未使用
	EX_WAIT0~2	入力	本アプリケーションノートでは未使用
	DACK0~1	出力	本アプリケーションノートでは未使用
	DREQ0~1	入力	本アプリケーションノートでは未使用
	DRACK0	出力	本アプリケーションノートでは未使用
DBSC3	MCK0	出力	クロック出力
	MCK0#	出力	クロック出力 MCK0 の反転クロック出力
	MCKE	出力	CKE 出力信号
	MCS#	出力	チップセレクト出力信号
	MWE#	出力	ライトイネーブル出力信号
	MRAS#	出力	ロウアドレスストロープ出力信号
	MCAS#	出力	カラムアドレスストロープ出力信号
	MA[13:0]	出力	アドレス出力信号
	MBA[2:0]	出力	バンクアドレス出力信号
	MDQ[15:0]	入出力	データ入出力信号
	MDQS1、 MDQS0	入出力	データストロープ入出力信号
	MDQS1#、 MDQS0#	入出力	データストロープ入出力信号 MDQS1、MDQS0 の反転
	MDM1、MDM0	出力	データマスク出力信号
	MODT	出力	SDRAM 内の ODT のイネーブル出力信号
	MZQ	入出力	キャリブレーション用端子 120 (精度 1%以上) の抵抗にて VSS に接続。
	MRESET#	出力	DDR3-SDRAM 用リセット出力。 DDR2-SDRAM 接続のため、オープン状態。
	SDBUP	入力	電源バックアップモニタ DDR2-SDRAM を使用する場合には、ローレベルに固定してください。
	SDSELF	出力	DDR セルフリフレッシュ通知 DDR セルフリフレッシュモードに移行したことを通知します。 本アプリケーションノートでは未使用。
	MVREFCA	入力	入力基準電圧。GND に接続してください。

モジュール	端子名	入出力	内容
SCIF3 (デバッグ用)	TXD	出力	ターミナルソフトへの出力データ
	RXD	入力	ターミナルソフトからの入力データ
GPIO (デバッグ用)	GP1[15]	出力	LED4
	GP1[22]	出力	LED5
AD (デバッグ用)	AN0	入力	PS3 ~ 5
	AN1	入力	PS6 ~ 8
	AN2	入力	PS9 ~ 11

【注】 表 5.1では、本アプリケーションノートで関係する端子のみ記載します。

クロック、システム制御、H-UDI、割り込み、電源系端子については、省略します。

本 LSI 立ち上げ時、必ず LSI ピンマルチ設定レジスタにて、ピンマルチ設定を行ってください。

また、LSI ピン PULL_UP 制御レジスタにて、ピン PULL_UP 設定を検討ください。

* パワーオンリセット時、GP0[25]、GP0[26]は、初期値が GPIO 設定になっています。このため、フラッシュ書き込みツール (fmttool 等) を使用する際には、ダウンロード前に、A24、A25 に設定してください。本アプリケーションノートでは、添付の.hdc ファイルにより実施しています。

5.2 参考回路

「SH7734 評価用プラットフォーム ユーザーズマニュアル (R0P7734C00000RZ)」を参照ください。

6. ソフトウェア説明

6.1 動作概要

C 言語で作成されたメイン関数を実行するためには、パワーオンリセット後に、メモリの初期化など最低限のハードウェア初期化処理を行う初期設定プログラムが必要です。本アプリケーションノートでは初期設定プログラムにおける初期設定例を説明します。

図 6.1、図 6.2にパワーオンリセットからメイン関数までの処理フローを示します。

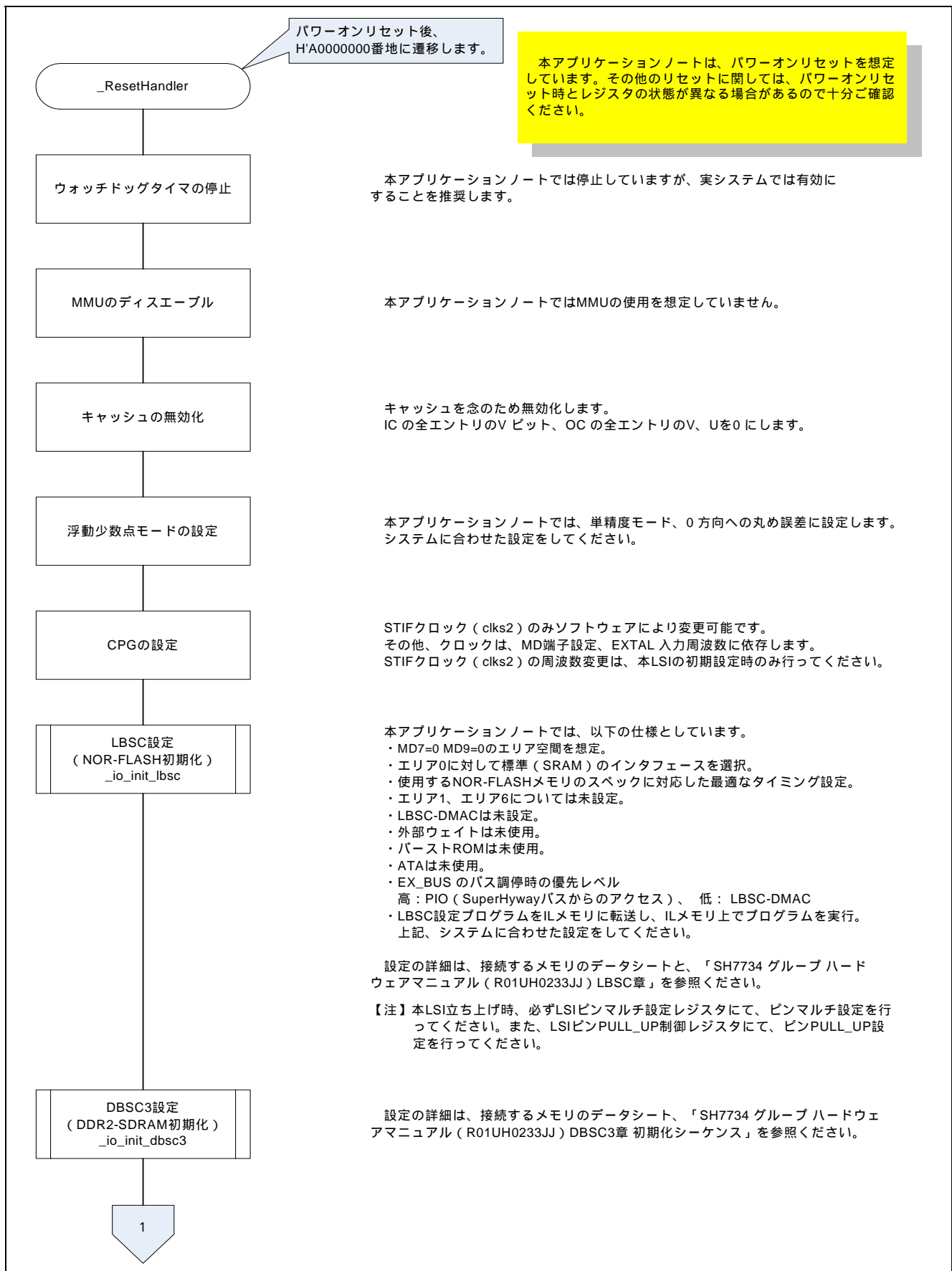


図6.1 パワーオンリセットからメイン関数までの処理フロー1

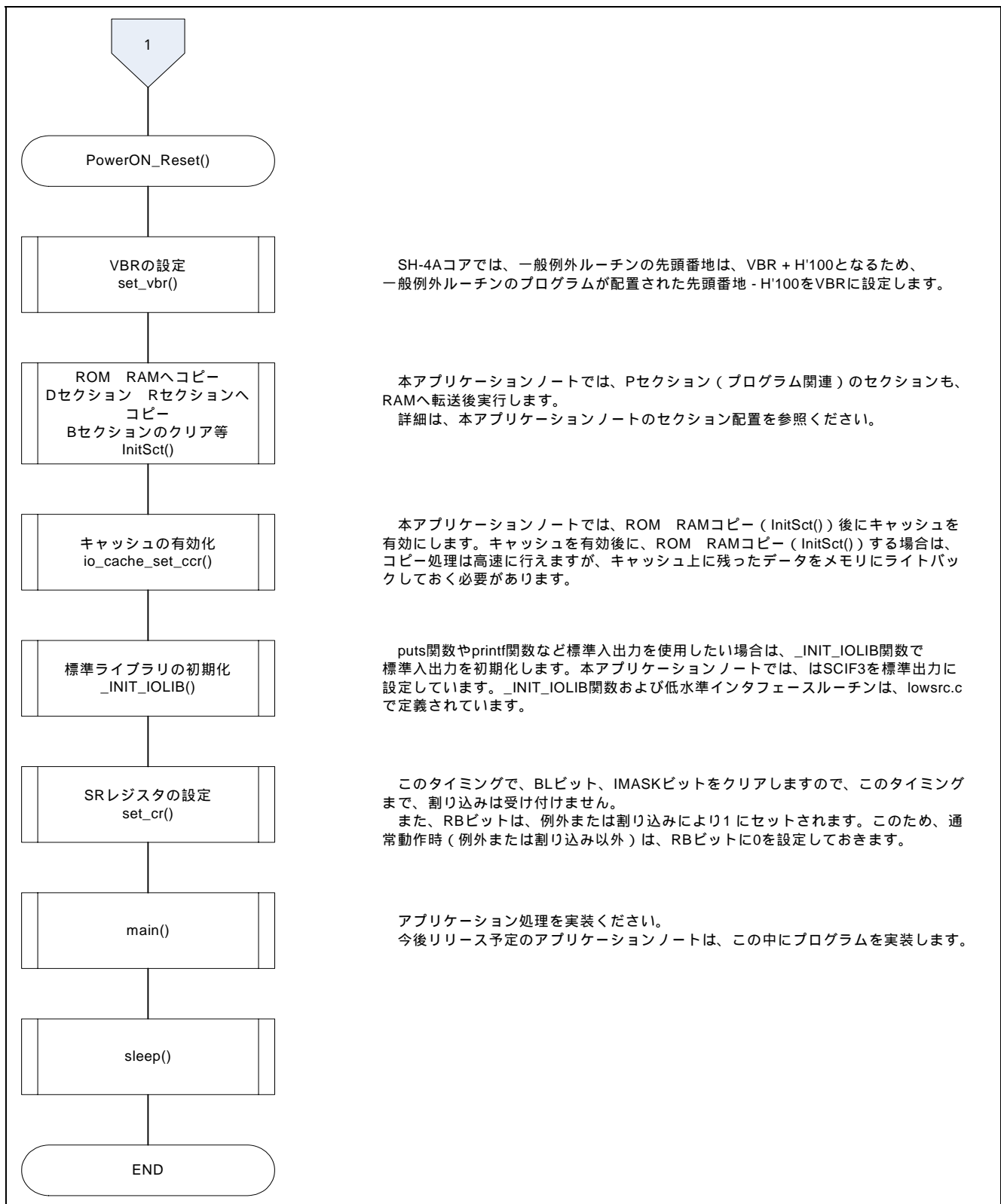


図6.2 パワーオンリセットからメイン関数までの処理フロー2

6.2 ファイル構成

表 6.1にサンプルコードで使用するファイルを示します。

表6.1 ファイル構成

ファイル名	概要	備考	
vhandler.src	例外（リセット、一般例外、割り込み）のハンドラが定義されています。	「SH7730グループ SH7730 初期設定例(RJJ06B0864)」のアプリケーションノートに各ファイルの詳細説明がありますので、そちらを参照ください。	
vecttbl.src	例外（リセット、一般例外、割り込み）関数テーブル、例外関数処理中の割り込みレベル設定テーブルが定義されています。		
resetprg.c	パワーオンリセット後、最初に行われる C 言語関数（PowerON_Reset()）が定義されています。		
dbstc.c	B セクションの 0 クリア、および D セクションから R セクションへのコピー処理等を行います。また、ROM 上のプログラムを RAM 上にコピーします。		
sh7734_main.c	main()が定義されています。		
intprg.c	一般例外、割り込み関数が定義されています。		
vect.inc	例外（リセット、一般例外、割り込み）関数の.global定義がされています。		
sbrk.c	統合開発環境で自動生成されるファイルです。		詳細は、「SuperH RISC engine C/C++コンパイラパッケージアプリケーションノート (RJJ05B0557)」を参照ください。
sbrk.h			
stackstc.h			
env.inc			
typedefine.h			
io_cache.c	キャッシュ制御関連が定義されたファイルです。	詳細は、「SH7730 グループ キャッシュメモリの設定例 (RJJ06B0868)」のアプリケーションノートを参照ください。	
io_cache.h			
iodefine.h	SH7734 用の iodefine.h です。		
io_lbsc.src	NOR フラッシュメモリ（エリア 0）の初期設定が定義されたファイルです。		
io_dbstc3.src	DDR2-SDRAM（エリア 2、3）の初期設定が定義されたファイルです。		
lowsrc.c	標準入出力をサポートしています。_INIT_IOLIB 関数および低水準インタフェースルーチンを実装しています。	デバッグ用途で使用します。「SuperH RISC engine C/C++コンパイラパッケージアプリケーションノート (RJJ05B0557)」を参照ください。	
lowsrc.h			
io_siorw.c			
io_siochar.c			
io_led.c	ユーザー開放用 LED 制御関連のサンプルが定義されたファイルです。	デバッグ用途で使用します。	
io_led.h			
io_key.c	キーマトリックス制御関連のサンプルが定義されたファイルです。	デバッグ用途で使用します。	
io_key.h			
io_tmu.c	タイマ関連のサンプルが定義されたファイルです。	デバッグ用途で使用します。	
io_tmu.h			
rop7734c0000orz.h	SH7734 評価用プラットフォームに依存する設定を定義したものです。		

6.3 定数一覧

表 6.2にサンプルコードで使用する定数を示します。

表6.2 サンプルコードで使用する定数

定数名	設定値	内容
IMASKclr 【vhandler.src】	H'FFFFFF0F	SR レジスタの IMASK ビットをクリアする際に使用
RBBLclr 【vhandler.src】	H'CFFFFFFF	SR レジスタの RB、BL ビットをクリアする際に使用
MDRBBLset 【vhandler.src】	H'70000000	SR レジスタの MD、RB、BL ビットをセットする際に使用
SR_Init 【vhandler.src】	H'40000000	main()実行前の SR レジスタの状態をセット
INT_OFFSET 【vhandler.src】	H'00000100	VBR の設定時に使用
D_CACHE_OFF 【io_cache.h】	H'00000000	CCR レジスタの初期化
D_CACHE_I_INVALID 【io_cache.h】	H'00000800	CCR レジスタの ICI ビットをセット IC の全エントリの V ビットを 0 にします。
D_CACHE_I_ON 【io_cache.h】	H'00000100	CCR レジスタの ICE ビットをセット 命令キャッシュ有効化
D_CACHE_O_INVALID 【io_cache.h】	H'00000008	CCR レジスタの OCI ビットをセット OC の全エントリの V、U ビットを 0 にします。
D_CACHE_O_ON 【io_cache.h】	H'00000001	CCR レジスタの OCE ビットをセット オペランドキャッシュ有効化
D_CACHE_IO_ON 【io_cache.h】	(CACHE_I_ON CACHE_O_ON)	命令キャッシュ、オペランドキャッシュ有効化
D_CACHE_O_WT 【io_cache.h】	H'00000002	ライトスルーモード設定

【注】 レジスタアドレス関連、統合開発環境で自動生成されるもの、デバッグ用途に使用するものについては、記載を省略します。

6.4 セクション配置

表 6.3にセクション配置、表 6.4にROM から RAM へマップするセクションを示します。

表6.3 セクション配置

セクション名	セクション用途	領域	配置アドレス (仮想アドレス)	
P* ³	プログラム領域 (指定なしの場合)	ROM	H'00003000	P0 領域 (キャッシング可能、 MMU アドレス変換可能)
C	定数領域	ROM		
P\$PSEC* ⁴	セクション初期化プログラム領域	ROM		
C\$BSEC	未初期化データ領域用アドレス構造体	ROM		
C\$DSEC	初期化データ領域用アドレス構造体	ROM		
D	初期化データ (初期値)	ROM		
B	未初期化データ領域	RAM	H'0C000000	
R	初期化データ領域	RAM		
PRAM* ³	ROM 化プログラム (P) コピー領域	RAM		
S	スタック領域	RAM	0x0FFFF9F0	
PINTHandler* ⁵	例外 / 割り込みハンドラ	ROM	H'80000800	P1 領域 (キャッシング可能、 MMU アドレス変換不可)
VECTBL	リセットベクタテーブル	ROM		
INTTBL* ⁵	割り込みベクタテーブル 割り込みマスクテーブル	ROM		
PIntPRG* ⁵	割り込み関数	ROM		
SP_S* ¹	TLB ミスハンドラ専用スタック領域	RAM	H'8FFFFFFDF0	
RSTHandler	リセットハンドラ	ROM	H'A0000000	P2 領域 (キャッシング不可、 MMU アドレス変換不可)
PResetPRG	リセットプログラム	ROM		
P_LBSC_ROM* ⁶	ROM 化プログラム領域 (LBSC 用)	ROM		
P_DBSC3_ROM* ⁶	ROM 化プログラム領域 (DBSC3 用)	ROM		
PnonCache* ²	プログラム領域 (キャッシュ無効アクセス)	ROM		
INTTBL_OL* ⁵	割り込みマスクテーブルコピー領域	RAM	H'E500E000	OL メモリ
PINTHandler_IL* ⁵	例外 / 割り込みハンドラコピー領域	RAM	H'E5200000	IL メモリ
PIntPRG_IL* ⁵	割り込み関数コピー領域	RAM		
P_LBSC_IL* ⁶	ROM 化プログラムコピー領域 (LBSC 用)	RAM		

【注】 *1 特別なセクションを設けている理由については、「SH7730 グループ SH7730 初期設定例 (RJJ06B0864)」のアプリケーションノートを参照ください。

*2 特別なセクションを設けている理由については、「SH7730 グループ SH7730 初期設定例 (RJJ06B0864)」のアプリケーションノートを参照ください。

*3 本アプリケーションノートでは、P セクションに割り当てられるプログラム (NOR フラッシュ上) を PRAM セクション (SDRAM 上) で実行するようにセクションのコピーを行っています。

*4 【注】*3 のことを行う場合で、セクションのコピーに、標準ライブラリ関数 (_INITSCT()) を使用する際に次の考慮が必要です。

_INITSCT()は、P セクションに配置されますので、_INITSCT()実行時には、ROM から RAM へマップのため、P セクション (ROM) でなく、PRAM セクション (RAM) 上に配置されているプログラムを実行しようとしています。しかし、セクションのコピーの目的のために、_INITSCT()をコールする段階では、_INITSCT()自体が PRAM セクション (RAM) 上にコピーされていないため _INITSCT() を実行することができません。

これを回避するため、本アプリケーションノートでは、_INITSCT()と同機能の別関数 (InitSct()) を作成し、この関数を P\$PSEC セクション (ROM) 上に配置して、このセクションで実行するようにします。

- *5 本アプリケーションノートでは、例外 / 割り込みハンドラ、割り込み関数に割り当てられるプログラムの速度向上のため、IL メモリ上で実行するようにしています。また、割り込み処理中に参照されるテーブル (INTTBL) については、OL メモリ上から参照されるようにしています。
- *6 プログラム実行中のエリアで、そのエリアの設定 (LBSC、DBSC3 等) を変更することは推奨しません。(特にバースト ROM 設定時にご注意ください。) このため、本アプリケーションノートでは、プログラム実行中のエリア 0 以外の IL メモリ上のプログラムにて、エリア 0 を設定変更後 (LBSC 設定変更後) に、再度、エリア 0 にアクセスしています。DBSC3 の設定に関しては、エリア 0 上のプログラムで実行しているため、そのままとします。

表6.4 ROM から RAM へマップするセクション

ROM	メモリ	RAM	メモリ
P	NOR フラッシュ	PRAM	SDRAM
D	NOR フラッシュ	R	SDRAM
PINTHandler	NOR フラッシュ	PINTHandler_IL	IL メモリ
PIntPRG	NOR フラッシュ	PIntPRG_IL	IL メモリ
P_LBSC_ROM	NOR フラッシュ	P_LBSC_IL	IL メモリ
INTTBL	NOR フラッシュ	INTTBL_OL	OL メモリ

7. 応用例

SH7734 評価用プラットフォーム (R0P7734C00000RZ) でのデバッグ用サンプルについて紹介します。

7.1 ユーザー開放用 LED 制御

ユーザー開放用 LED 制御 (LED4、LED5) について示します。

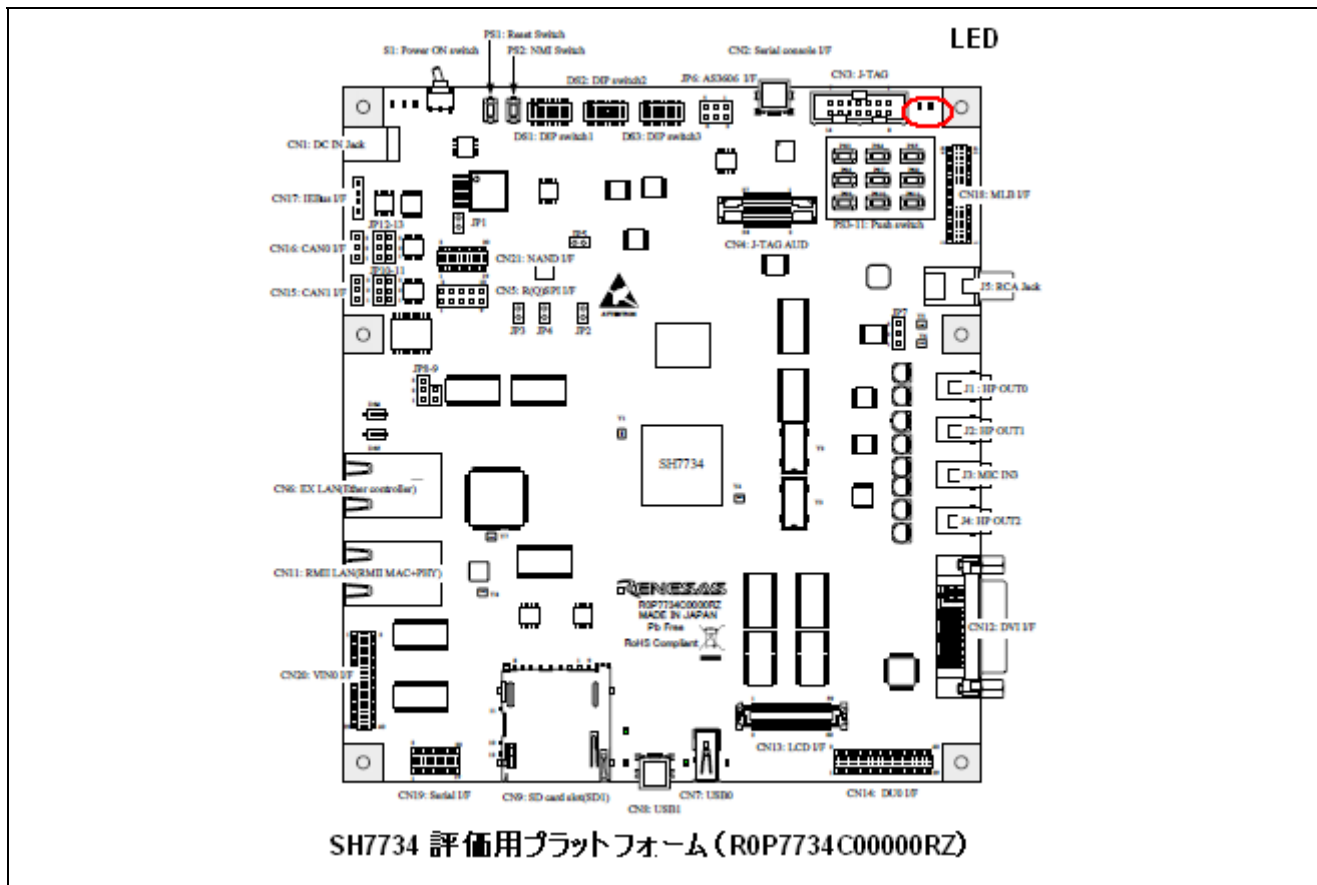


図7.1 ユーザー開放用 LED

7.1.1 関数一覧

表 7.1 にユーザー開放用 LED 制御関数を示します。main()からコールされることを想定します。

表7.1 ユーザー開放用 LED 制御関数

関数名	概要
io_led_init	GP1[15] (LED4)、GP1[22] (LED5) をポート出力に初期化します。
io_led_on	対象の LED を ON します。
io_led_off	対象の LED を OFF します。

7.1.2 関数仕様

サンプルコードの関数仕様を示します。

io_led_init	
概要	GP1[15]、GP1[22]をポート出力に初期化します。
ヘッダ	io_led.h
宣言	void io_led_init(void)
説明	GP1[15]、GP1[22]をポート出力に初期化します。
引数	なし
リターン値	なし
備考	

io_led_on	
概要	対象のLEDをONします。
ヘッダ	io_led.h
宣言	void io_led_on(E_ID_USER_LED ledno)
説明	引数で指定されたLEDをONにします。 一度にLED4、LED5をONにすることも可能です。
引数	第一引数 : ledno D_ID_USER_LED4 : LED4をON D_ID_USER_LED5 : LED5をON D_ID_USER_LED_ALL : LED4、LED5をON
リターン値	なし
備考	

io_led_off	
概要	対象のLEDをOFFします。
ヘッダ	io_led.h
宣言	void io_led_off(E_ID_USER_LED ledno)
説明	引数で指定されたLEDをOFFにします。 一度にLED4、LED5をOFFにすることも可能です。
引数	第一引数 : ledno D_ID_USER_LED4 : LED4をOFF D_ID_USER_LED5 : LED5をOFF D_ID_USER_LED_ALL : LED4、LED5をOFF
リターン値	なし
備考	

7.1.3 使用例

```
void main(void)
{
    /* 初期化 */
    io_led_init();

    io_led_on(D_ID_USER_LED4); /* LED4 ON */
    io_led_on(D_ID_USER_LED5); /* LED5 ON */

    io_led_off(D_ID_USER_LED4); /* LED5 OFF */
    io_led_off(D_ID_USER_LED5); /* LED5 OFF */

    io_led_on(D_ID_USER_LED_ALL); /* LED4,5 ON */
    io_led_off(D_ID_USER_LED_ALL); /* LED4,5 OFF */
}
```

7.2 デバッグシリアルインタフェース(SCIF3)

デバッグシリアルインタフェース (SCIF3) について示します。

PC 上のターミナルソフトとのデバッグ情報の入出力ができます。

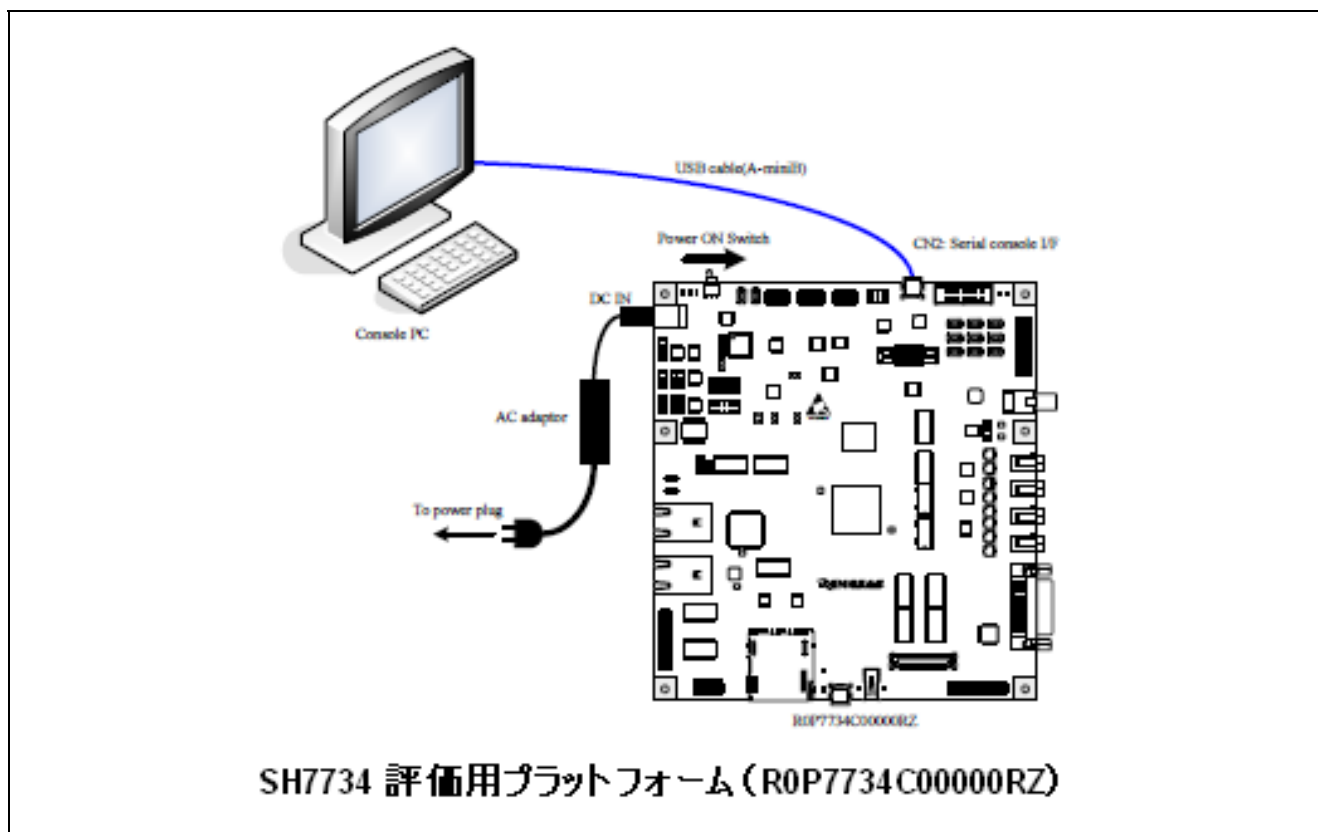


図7.2 デバッグシリアルインタフェース(SCIF3)

7.2.1 標準入出力

puts 関数、get 関数、printf 関数など標準入出力が使用できます。SCIF3 を標準入出力に設定しています。_INIT_IOLIB 関数および低水準インタフェースルーチンは、lowsrc.c で定義されています。

【注】 排他制御をしていません。

割り込み関数からコールされることを想定していません。

7.2.2 ターミナルソフトの設定

表 7.2 にターミナルソフトの設定を示します。

表7.2 ターミナルソフトの設定

項目	設定値
ビットレート	115200bps
データ長	8 ビット
ストップビット長	1 ビット
パリティ	なし

7.2.3 使用例

```
void main(void)
{
    puts( " " );
    puts("¥nSH7734 Sample Program. Ver.1.00.00");
    puts("Copyright (C) 2011 Renesas Electronics Corp. All Rights Reserved");
    puts("and Renesas Solutions Corp. All Rights Reserved");
    puts( " " );
    printf("test sample¥n");
    fflush(stdout);
}
```

7.3 タイマ制御

タイマ制御 (TMU0) について示します。タイムアウト時に割り込みが発生し、登録したコールバック関数がコールされます。

7.3.1 関数一覧

表 7.3 にタイマ制御関数を示します。

表7.3 タイマ制御関数

関数名	概要
io_tmu0_create	TMU0 を初期化します。
io_tmu0_start	タイマを起動します。
io_tmu0_stop	タイマを停止します。
io_tmu0_compare_match	タイムアウト時の割り込み関数からコールされます。

7.3.2 関数仕様

サンプルコードの関数仕様を示します。

io_tmu0_create	
概要	TMU0 を初期化します。
ヘッダ	io_tmu.h
宣言	void io_tmu0_create(T_TMU_TimerTYPE *i_pTimerType)
説明	割り込みを有効にします。 引数で指定された割り込み優先度を設定します。
引数	第一引数 : i_pTimerType タイマ種別構造体アドレス
リターン値	なし
備考	

io_tmu0_start	
概要	タイマを起動します。
ヘッダ	io_tmu.h
宣言	void tmu0_start(T_TMU_SETTCOR_INFO *i_pTCORInfo, TMU_TIMEOUT_CALLBACK i_func)
説明	引数で指定されたタイマ時間をタイマカウンタに設定します。 引数で指定されたコールバック関数を登録します。 タイマを開始します。
引数	第一引数 : i_pTCORInfo TCOR 設定情報構造体アドレス 第一引数 : i_func タイムアウト時コールバック関数
リターン値	なし
備考	タイマ設定値は ms 単位で、1ms ~ 1000ms のみ対応しています。 上記以外のタイマ値を指定した場合、1ms として対応します。

io_tmu0_stop	
概要	タイマを停止します。
ヘッダ	io_tmu.h
宣言	void tmu0_stop(void)
説明	タイマを停止します。
引数	なし
リターン値	なし
備考	

io_tmu0_compare_match	
概要	タイムアウト時の割り込み関数からコールされます。
ヘッダ	io_tmu.h
宣言	void tmu0_compare_match(void)
説明	タイマ割り込み要因をクリアします。 io_tmu0_start()で登録しておいたコールバック関数がコールされます。
引数	なし
リターン値	なし
備考	

7.3.3 使用例

例として、500ms ごとに LED を点滅させる (led_onoff() をコールする) プログラムを示します。

```
/* メイン関数 */
void main(void)
{
    T_TMU_TimerTYPE      TimerType;
    T_TMU_SETTCOR_INFO   SetTCORInfo;

    memset(&TimerType, 0x00, sizeof(TimerType));
    memset(&SetTCORInfo, 0x00, sizeof(SetTCORInfo));

    io_led_init(); /* LED 初期化 */

    /* タイマ生成用構造体の設定 */
    TimerType.mClockSelect = D_TMU_DIV_64;          /* clkp /64 */
    TimerType.mIntcPri = D_TMU_PRI_1;              /* 割り込みレベル 1 */

    /* TMU0 タイマ生成処理 */
    io_tmu0_create(&TimerType);

    /* タイマ時間設定 */
    SetTCORInfo.mTimeValue = 500;                  /* タイマ設定値(500ms) */

    /* TMU0 タイマ開始処理 */
    io_tmu0_start(&SetTCORInfo, led_onoff);
}

/* 割り込み関数 */
void INT_TMU00(void)
{
    io_tmu0_compare_match();
}

/* コールバック関数 */
void led_onoff(void)
{
    if(g_onoff == 0)
    {
        io_led_on(D_ID_USER_LED4);
        g_onoff = 1;
    }
    else
    {
        io_led_off(D_ID_USER_LED4);
        g_onoff = 0;
    }
}
}
```

7.4 キーマトリックスインタフェース制御

キーマトリックスインタフェース制御について示します。

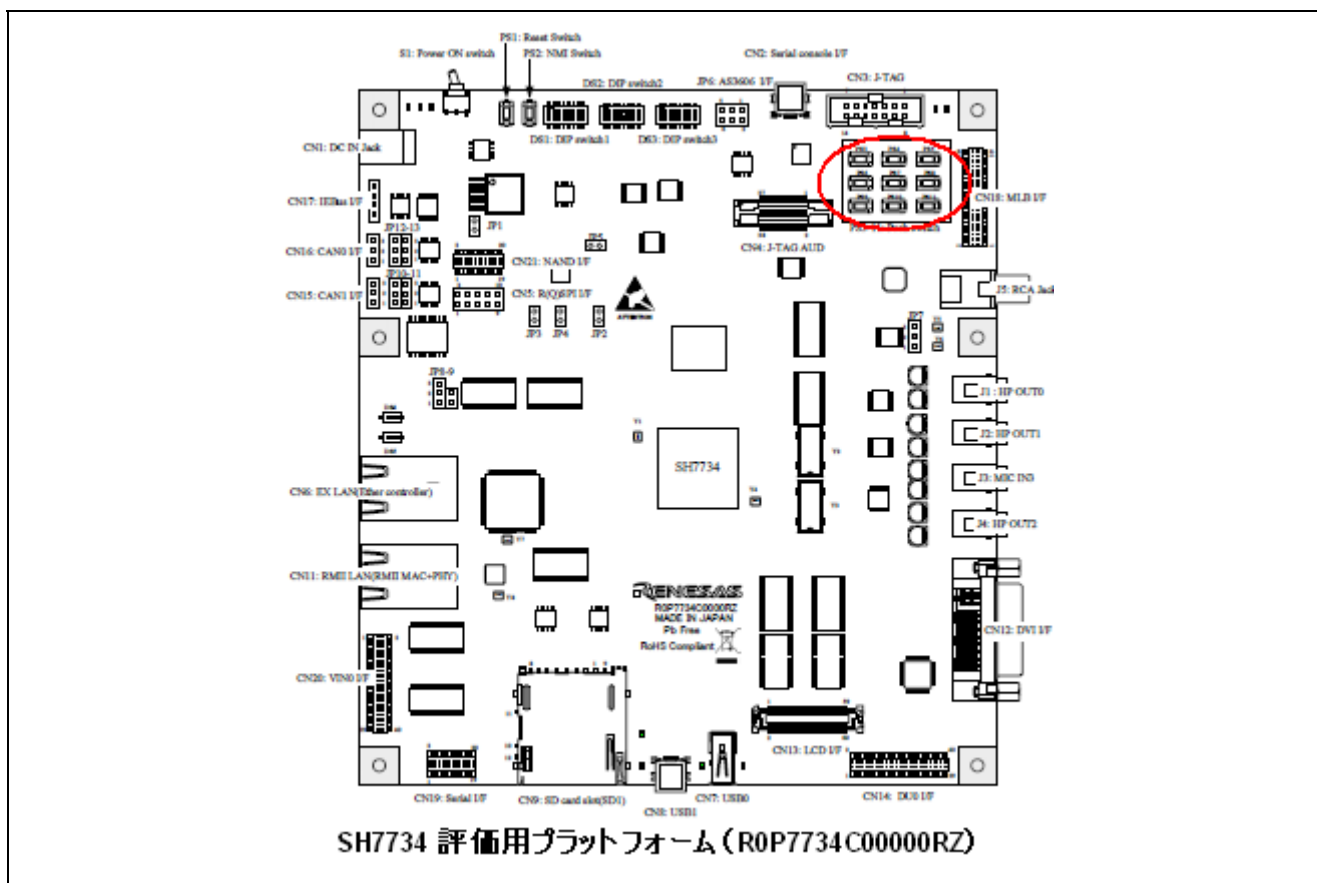


図7.3 キーマトリックスインタフェース

7.4.1 関数一覧

表 7.4にキーマトリックスインタフェース制御関数を示します。main()からコールされることを想定します。

「SH7734 評価用プラットフォーム ユーザーズマニュアル (R0P7734C00000RZ) キーマトリックスインタフェース (ADC) 章」のように Push されたボタンにより、AN0 ~ AN2 が変化します。

ADC (AN0 ~ AN2) を MTU2 からの変換トリガ (10ms 間隔) で常に A/D 変換します。

A/D 変換完了時の割り込み処理で、変換された A/D データより Push されたボタンを判別し、対応するコールバック関数をコールします。

仮に同時に複数のスイッチが Push された場合は、本アプリケーションノートでは、PS9 ~ PS11 > PS6 ~ PS8 > PS3 ~ PS5 の優先順位でスイッチを認識します。

表7.4 キーマトリックスインタフェース制御関数

関数名	概要
io_key_init	MTU2、ADC を初期化します。キーに対応するコールバック関数を登録します。
io_key_start	MTU2 カウントを開始します。 MTU2 からの変換トリガ (10ms 間隔) による A/D 変換開始。
io_key_stop	MTU2 カウントを停止します。
io_key_int	AD 変換完了時の割り込み関数からコールされます。

7.4.2 関数仕様

サンプルコードの関数仕様を示します。

io_key_init	
概要	MTU2、ADC を初期化します。キーに対応するコールバック関数を登録します。
ヘッダ	io_key.h
宣言	void io_key_init(T_KEY_MTX *key_mtx)
説明	MTU2 を初期化します。 ADC を初期化します。 キーに対応するコールバック関数を登録します。 キーPush 時に対応の登録関数がコールされます。
引数	第一引数：*key_mtx キーマトリックス情報構造体
リターン値	なし
備考	

io_key_start	
概要	MTU2 カウントを開始します。 MTU2 からの変換トリガ（10ms 間隔）による A/D 変換開始。
ヘッダ	io_key.h
宣言	void io_key_start(void)
説明	MTU2 カウント（10ms）を開始します。 MTU2 カウントでオーバーフロー時（10ms）経過後、AD 変換を開始します。
引数	なし
リターン値	なし
備考	

io_key_stop	
概要	MTU2 カウント、AD 変換を停止します。
ヘッダ	io_key.h
宣言	void io_key_stop(void)
説明	MTU2 カウントを停止します。 AD 変換を停止します。
引数	なし
リターン値	なし
備考	

io_key_int	
概要	ADC 変換完了時の割り込み関数からコールされます。
ヘッダ	io_key.h
宣言	void io_key_int(void)
説明	AN0 ~ AN2 の変換された A/D データより Push されたボタンを判別します。 ADC 入力仕様については、「SH7734 評価用プラットフォーム ユーザーズマニュアル（R0P7734C00000RZ）」を参照ください。 割り込み要因フラグをクリアします。 登録しておいたコールバック関数がコールされます。
引数	なし
リターン値	なし
備考	

7.4.3 使用例

例として、キー0 Push 時に、LED4 を点灯させ、キー8 Push 時に、LED4 を消灯するプログラムを示します。
また、その他キーは無処理とします。

```
/* キー0 Push 時の処理 */
void Key_0(void)
{
    io_led_on(D_ID_USER_LED4);
}

/* キー8 Push 時の処理 */
void Key_8(void)
{
    io_led_off(D_ID_USER_LED4);
}

/* キー押下時のコールバック関数の定義 */
static T_KEY_MTX t_key_tbl[E_KEY_FLG_MAX] =
{
    Key_0,
    NULL, /* 登録なし */
    NULL, /* 登録なし */
    NULL, /* 登録なし */
    NULL, /* 登録なし */
    NULL, /* 登録なし */
    NULL, /* 登録なし */
    NULL, /* 登録なし */
    Key_8,
}

/* メイン関数 */
void main(void)
{
    /* LED 初期化処理 */
    io_led_init();

    /* キーマトリックス初期化処理 */
    io_key_init(t_key_tbl);

    /* キーマトリックス開始処理 */
    io_key_start();
}

/* 割り込み関数 */
void INT_ADC (void)
{
    /* キーマトリックス割り込み処理 */
    io_key_int();
}
```

8. 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。

9. 参考ドキュメント

- SH7734 ユーザーズマニュアル ハードウェア編(R01UH0233JJ) Rev.1.00
(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。)
- C コンパイラマニュアル
SuperH C/C++コンパイラパッケージ V.9.04 ユーザーズマニュアル Rev.1.00
(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。)

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/inquiry>

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	2011.09.27	—	初版発行
1.01	2012.07.18	—	サンプルコードの修正 (iodefine.h, io_tmu.c) とサンプルコードバージョンを変更

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、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認ください。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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、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が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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■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/>