

要旨

本アプリケーションノートでは、RZ/T1 の評価ボード上に実装されている NOR フラッシュメモリへのリード、ライト、イレースを実行するサンプルプログラムについて説明します。

NOR サンプルプログラムの特長を以下に示します。

- ホストコンピュータと USB ケーブルで接続し、メニュー形式で NOR フラッシュのデータをリード/ライトできる。

制限事項

本サンプルプログラムには以下の制限事項があります。

- (1) NOR フラッシュのエリアのみに対応しています。他のエリアにリード/ライトすることはできません。
- (2) NOR フラッシュへライトできるデータはメモリ上のデータのみです。

動作確認デバイス

RZ/T1

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

目次

1.	仕様	3
2.	動作環境	4
3.	関連アプリケーションノート	5
4.	周辺機能説明	6
5.	ハードウェア説明	7
5.1	ハードウェア構成例	7
5.2	使用端子一覧	7
6.	ソフトウェア説明	8
6.1	動作概要	8
6.1.1	プロジェクト設定	8
6.1.2	使用準備	8
6.2	メモリマップ	10
6.2.1	サンプルプログラムのセクション配置	10
6.2.2	MPU の設定	11
6.2.3	例外処理ベクタテーブル	11
6.3	固定幅整数一覧	11
6.4	定数／エラーコード一覧	11
6.5	大域変数一覧	12
6.6	関数一覧	12
6.7	関数仕様	13
6.7.1	nor_flash_init	13
6.7.2	nor_flash_write_buf	13
6.7.3	nor_flash_write_unit	14
6.7.4	nor_flash_erase_sector	14
6.7.5	nor_flash_erase_chip	14
6.7.6	R_NOR_FLASH_Open	15
6.7.7	R_NOR_FLASH_Erase_Chip	15
6.7.8	R_NOR_FLASH_Erase_Sector	15
6.7.9	R_NOR_FLASH_Write_Buf	16
6.7.10	R_NOR_FLASH_Close	16
6.7.11	nor_flash_command	16
6.7.12	main	17
7.	フローチャート	18
7.1	メイン処理	18
8.	サンプルプログラムの動作例	19
9.	サンプルプログラムの入手先	23
10.	参考ドキュメント	24

1. 仕様

表 1.1 に使用する周辺機能と用途を、図 1.1 にサンプルプログラム実行時の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
FIFO内蔵シリアルコミュニケーションインタフェース (SCIFA)	SCIFAの歩調同期式を使用し、RS-232CインタフェースによるCOMポート通信に使用

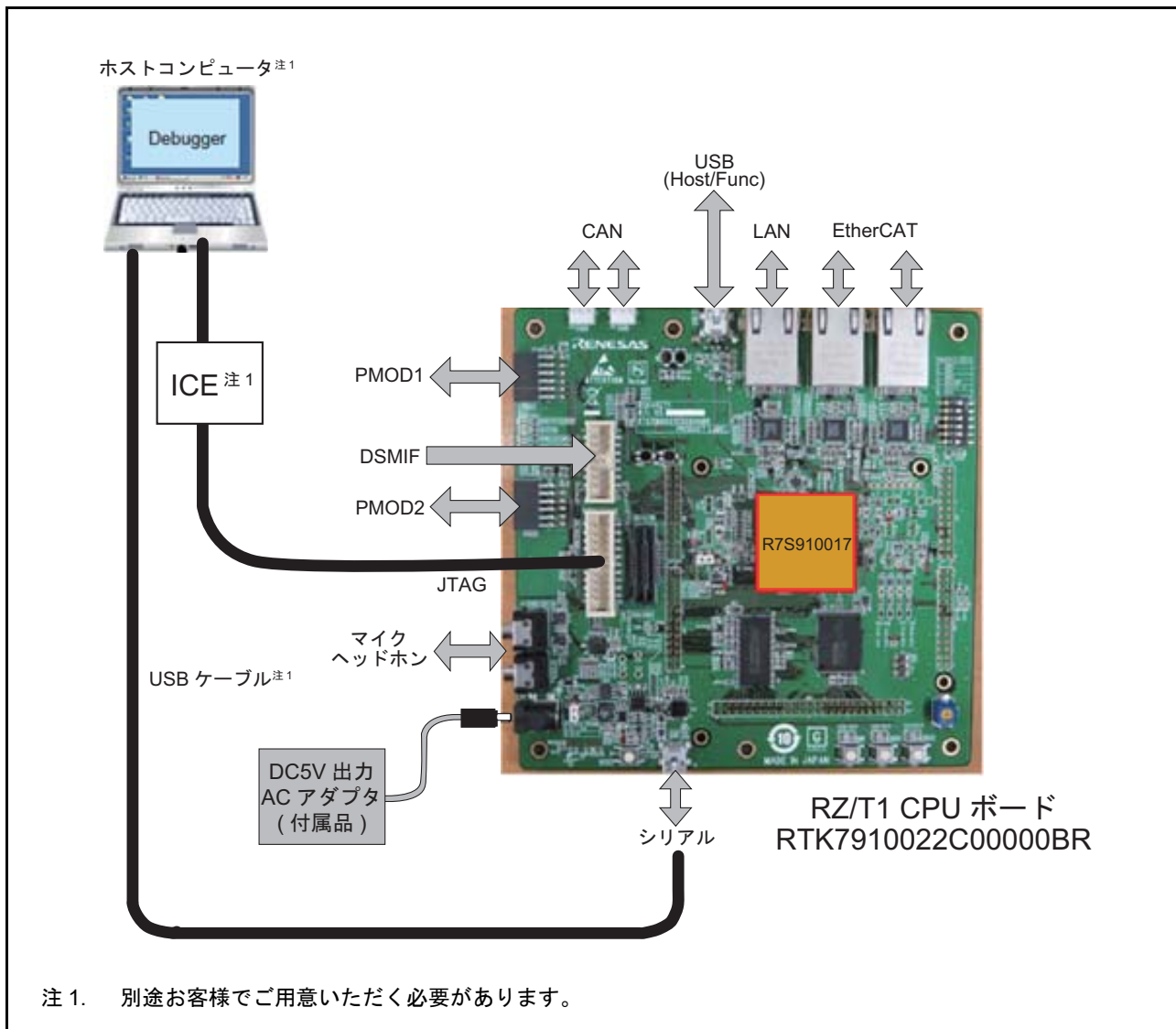


図 1.1 動作環境

2. 動作環境

本アプリケーション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	<ul style="list-style-type: none">• IARシステムズ製 Embedded Workbench® for Arm Version 8.20.2• Arm®総合開発環境 Arm Development Studio 5 (DS-5™) Version 5.26.2• Renesas e2studio Version: 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
ターミナルソフトの通信設定	<ul style="list-style-type: none">• 通信速度 : 115200bps• データ長 : 8ビット• パリティ : なし• ストップビット長 : 1ビット• フロー制御 : なし• 改行コード (受信) : CR• 改行コード (送信) : CR
使用ボード	RZ/T1評価ボード (RTK7910022C00000BR)
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none">• シリアルインタフェース (USB-Mini Bコネクタ J8)• NORフラッシュメモリ (CS0、CS1空間に接続) メーカー名 : Macronix International Co., Ltd. 型名 : MX29GL512FLT2I-10Q• シリアルフラッシュメモリ メーカー名 : Macronix International Co., Ltd. 型名 : MX25L51245G

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。

- RZ/T1 グループ 初期設定アプリケーションノート (R01AN2554JJ0100)
- RZ/T1 グループ FIFO 内蔵シリアルコミュニケーションインタフェース (R01AN2577JJ0100)

注. 本アプリケーションノートで記載しないレジスタに関しては、RZ/T1 グループ 初期設定アプリケーションノートおよび RZ/T1 グループ FIFO 内蔵シリアルコミュニケーションインタフェースで設定した値のまま使用しません。

4. 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。

5. ハードウェア説明

5.1 ハードウェア構成例

図 5.1 にハードウェア構成例を示します。

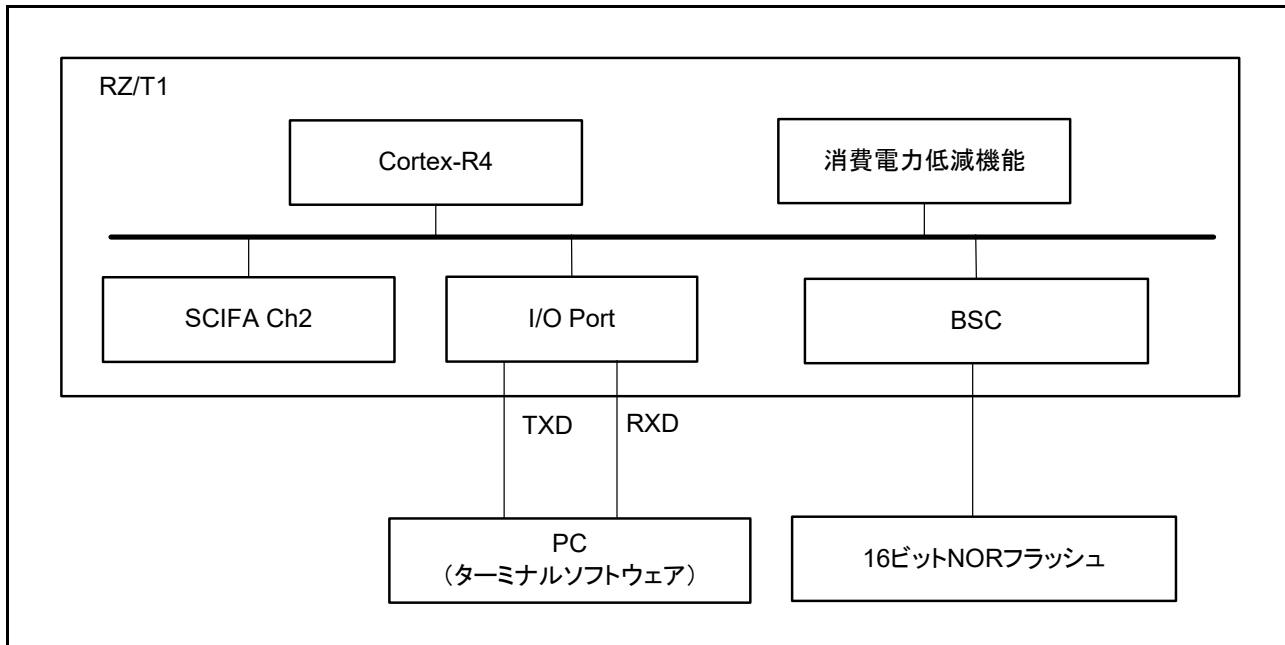


図 5.1 ハードウェア構成例

5.2 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

端子名	入出力	内容
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	
TXD2	出力	シリアル送信データ信号
RXD2	入力	シリアル受信データ信号
D15～D0	入出力	BSCデータ (NORフラッシュ接続用)
A25～A1	出力	BSCアドレス (NORフラッシュ接続用)
RD#	出力	BSCリードストローブ (NORフラッシュ接続用)
WE#	出力	BSCライトストローブ (NORフラッシュ接続用)
CS0#	出力	BSCチップセレクト (NORフラッシュ接続用)
CS1#	出力	BSCチップセレクト (NORフラッシュ接続用)

6. ソフトウェア説明

6.1 動作概要

本ソフトウェアには、NORフラッシュを使用するためのドライバ、NORフラッシュを消去、書き込みするためのAPI関数が含まれています。

またFIFO内蔵シリアルコミュニケーションインタフェース（SCIFA）の調歩同期式通信を用い、ホストPCとRS-232インタフェースのCOMポート通信を行い、ホストPC上のターミナルソフトウェアにてNORフラッシュを読み書き、消去するサンプルプログラムも含まれています。

6.1.1 プロジェクト設定

開発環境となるEWARM/DS-5/e2studio上で使用されるプロジェクト設定については、RZ/T1グループ初期設定アプリケーションノートに記載しています。

注：本サンプルでは、初期設定から変更があります。詳細は6.2.1を参照ください。

6.1.2 使用準備

本サンプルプログラムでは、PCとの通信動作を行いますので、その実行準備を説明します。

- (1) ホストPCにてターミナルソフトを起動し、シリアルポートの設定を次のように設定します。
(Tera TermでCOM9使用の場合)

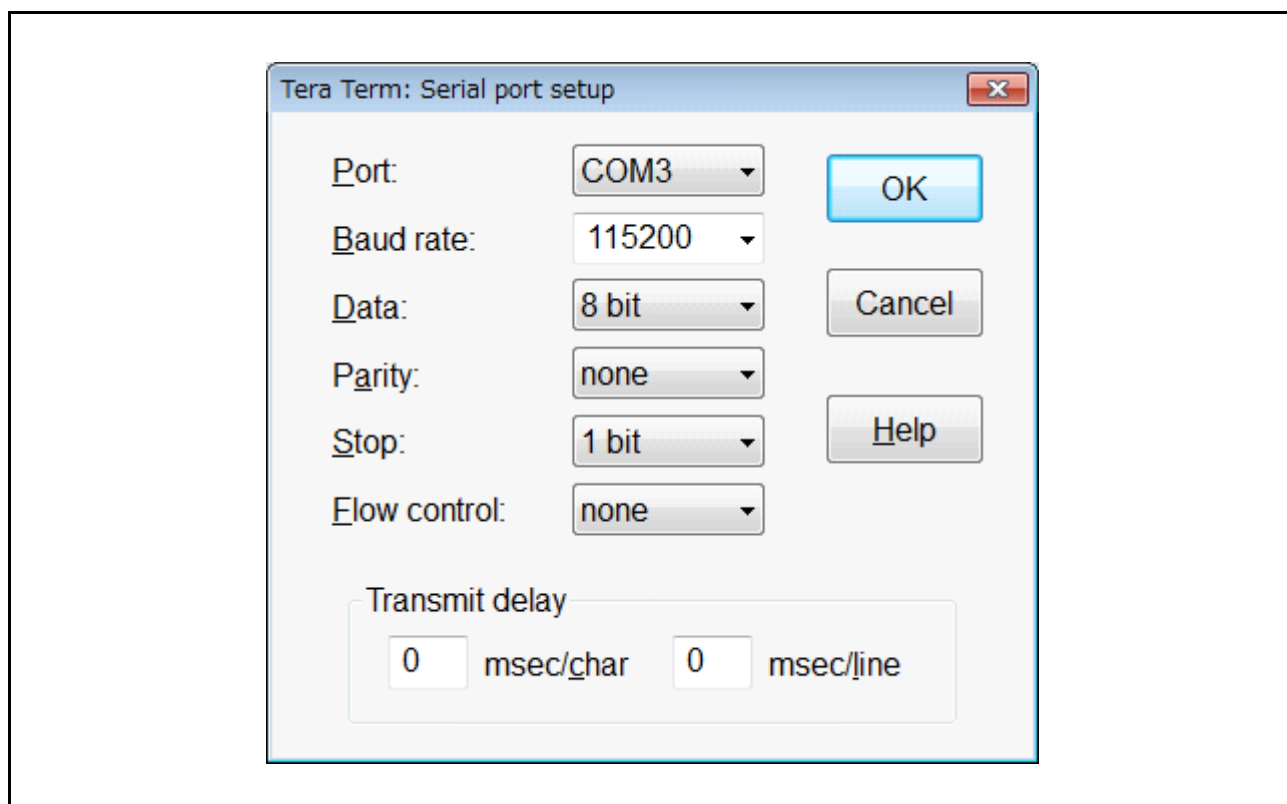


図 6.1 シリアルポートの設定

- (2) また、ターミナルソフトの端末設定では、送受信時とも改行コードが『CR』になるように設定します。

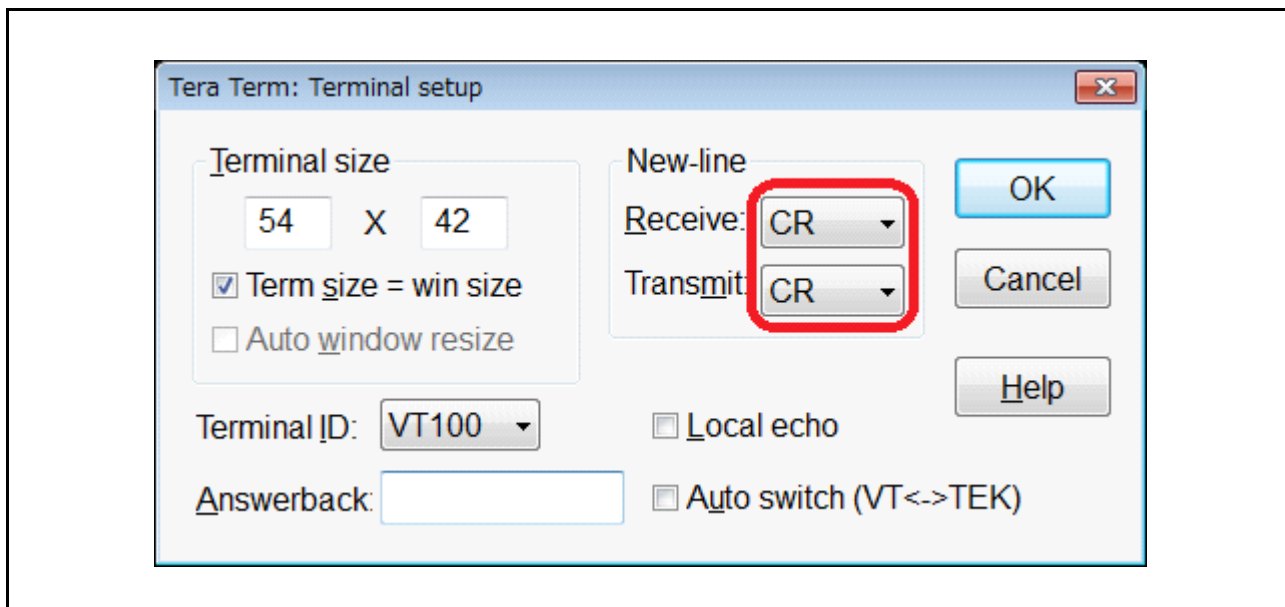


図 6.2 SCIFA 設定後のターミナルソフトの表示

- (3) サンプルプログラムを実行し通信可能状態になると、下図に示すようにサンプルプログラムからの受信データがターミナルソフトに表示されます。

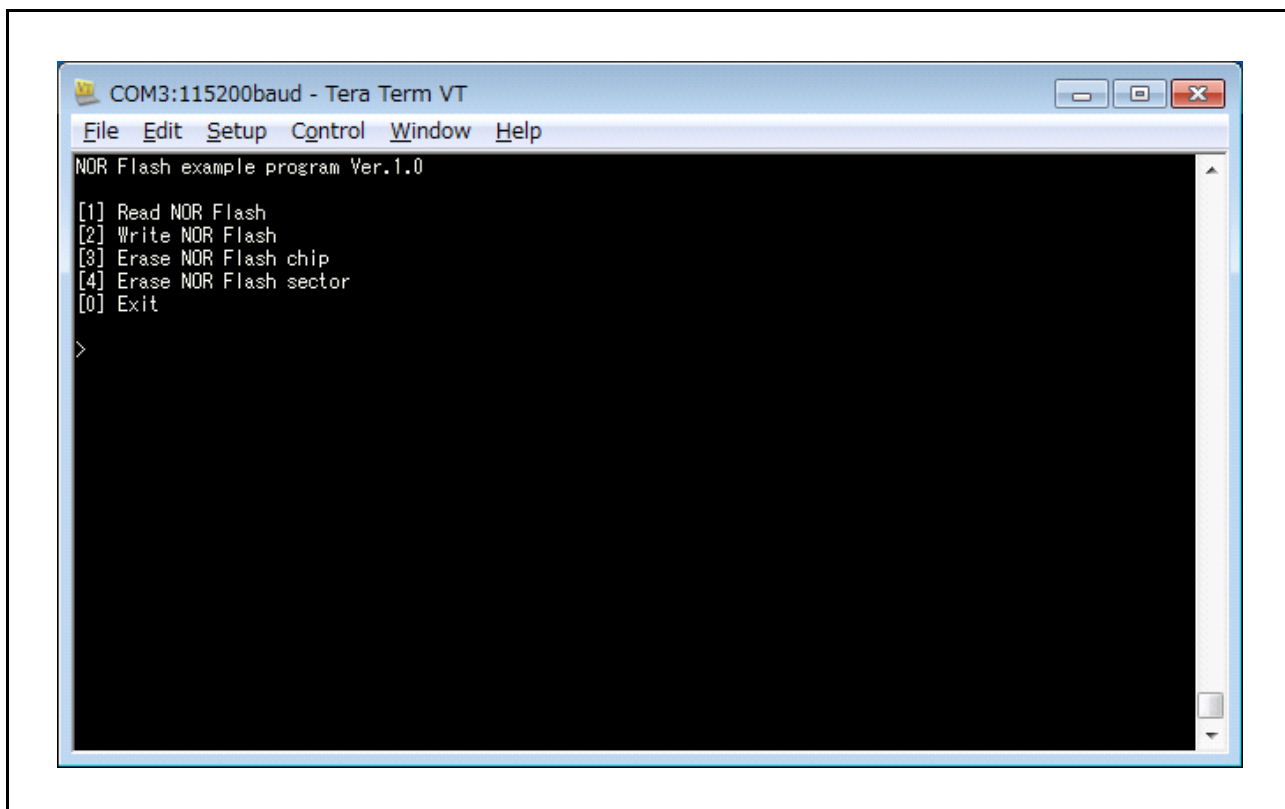


図 6.3 サンプルプログラム実行後のターミナルソフトの表示

6.2 メモリマップ

RZ/T1 グループのアドレス空間と RZ/T1 評価ボードのメモリマッピングについては、RZ/T1 グループ 初期設定アプリケーションノートに記載しています。

6.2.1 サンプルプログラムのセクション配置

サンプルプログラムで使用するセクションおよびサンプルプログラムの初期状態のセクション配置（ロードビュー）、スキップローディング機能を使用後のセクション配置（実行ビュー）は、RZ/T1 グループ 初期設定アプリケーションノートに記載しています。

注． 本サンプルでは、初期設定から以下の点を変更しています。

- 1) RAM 領域に定義されていたヒープ領域を SDRAM 領域での定義に変更
- 2) ヒープ領域のサイズを 0x50000 に変更
- 3) EWARM 環境の NOR ブート版のみ、NOR フラッシュ領域に定義されていた rodata セクションを、RAM 領域での定義に変更

なお、上記3について、下記に記します。

本サンプルプログラムの、EWARM 環境の NOR ブート版（RZ_T1_nor_nor_boot.eww）については、NOR フラッシュの内容を書き換えた際に、rodata セクションの内容を書き換えないために、rodata セクションを ATCM（密結合メモリ）に配置しています。配置の際に、RZ_T1_init_nor_boot.icf を変更しており、その変更内容を以下に示します（太字は追加文字、二重取り消し線は削除文字）。

```
define block USER_PRG_RBLOCK { ro code, section .rodata_init };
define block USER_PRG_WBLOCK { rw code, section .rodata };
:
initialize manually { 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 vector.o,
                    ro code,
                    section .rodata
                    };
:
place in ROM_region { block USER_PRG_RBLOCK, section .rodata };
```

6.2.2 MPU の設定

MPU の設定は、RZ/T1 グループ 初期設定アプリケーションノートに記載しています。

6.2.3 例外処理ベクタテーブル

例外処理のベクタテーブルについては、RZ/T1 グループ 初期設定アプリケーションノートに記載しています。

6.3 固定幅整数一覧

表 6.1 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 6.1 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）

6.4 定数／エラーコード一覧

表 6.2 にサンプルプログラムで使用する定数、表 6.3 にサンプルプログラムのエラーコードを示します。

表 6.2 サンプルプログラムで使用する定数

定数名	設定値	内容
R_NOR_MEM_CHIP1	1U	チップ1
R_NOR_MEM_CHIP2	2U	チップ2
R_NOR_MEM_TOP1	0x60000000U	実装されているNORフラッシュ 1のトップアドレス
R_NOR_MEM_END1	0x63FFFFFFU	実装されているNORフラッシュ 1のエンドアドレス
R_NOR_MEM_TOP2	0x64000000U	実装されているNORフラッシュ 2のトップアドレス
R_NOR_MEM_END2	0x67FFFFFFU	実装されているNORフラッシュ 2のエンドアドレス
R_NOR_MEM_SIZE	0x04000000U	実装されているNORフラッシュのサイズ
R_NOR_MEM_ADDR_LIMIT	0x68000000U	実装されているNORフラッシュのアドレスの境界値
R_NOR_READ_SIZE	128U	1回にリードできるワード数
R_NOR_WRITE_SIZE	0x04000000U	1回にライトできるバイト数
R_NOR_SECTOR_SIZE	64U * 1024U	1セクタのサイズ（符号なし）
R_NOR_SECTOR_SIZE_INT	64 * 1024	1セクタのサイズ（符号あり）
R_NOR_SECTOR_NO_SIZE	512U	セクタの総数
R_NOR_SECT_ADDR_MSK	0x0001FFFFU	アドレスを0x20000にするためのマスク値
R_NOR_SECT_NO_MSK	0x03FFFFFFU	セクタ番号を算出するためのマスク値

表6.3 サンプルプログラムのエラーコード

定数名	設定値	内容
R_NOR_SUCCESS	0U	関数コールが正常にできた場合
R_NOR_ERROR_PARAM	1U	関数の引数が正しくない
R_NOR_ERROR_FLASH_READ	2U	フラッシュリードの失敗
R_NOR_ERROR_FLASH_WRITE	3U	フラッシュライトの失敗
R_NOR_ERROR_FLASH_ERASE	4U	フラッシュ消去の失敗
R_NOR_ERROR_FLASH_INIT	5U	フラッシュ初期化の失敗

6.5 大域変数一覧

表 6.4 に大域変数一覧を示します。

表6.4 大域変数一覧

型	変数名	内容	使用関数
char	sbuff[16]	ターミナルソフトへの送信データ	main
char	rbuff[16]	ターミナルソフトからの受信データ	main

6.6 関数一覧

表 6.5 に関数一覧を示します。

表6.5 関数一覧

関数名	概要	ページ番号
nor_flash_init	NORフラッシュ初期化用ドライバ	13
nor_flash_write_buf	NORフラッシュライト用ドライバ（複数ワード）	13
nor_flash_write_unit	NORフラッシュライト用ドライバ（1ワード）	14
nor_flash_erase_sector	NORフラッシュセクタ消去用ドライバ	14
nor_flash_erase_chip	NORフラッシュチップ消去用ドライバ	14
R_NOR_FLASH_Open	ドライバAPI関数の開始	15
R_NOR_FLASH_Erase_Chip	NORフラッシュチップ消去API	15
R_NOR_FLASH_Erase_Sector	NORフラッシュセクタ消去API	15
R_NOR_FLASH_Write_Buf	NORフラッシュライトAPI（複数ワード）	16
R_NOR_FLASH_Close	ドライバAPI関数の終了	16
nor_flash_command	コマンドを発行する関数	16
main	サンプルプログラムのメイン関数	17

6.7 関数仕様

サンプルプログラムの関数仕様を示します。

6.7.1 nor_flash_init

nor_flash_init

概要	NORフラッシュ初期化	
ヘッダ	nor_flash.h	
宣言	uint32_t nor_flash_write(uint32_t address, uint32_t* data, uint32_t size)	
説明	NORフラッシュを初期化する関数です。	
引数	uint32_t chip_number	初期化するフラッシュのチップ No. を指定します。
リターン値	R_NOR_SUCCESS :	初期化成功
	R_NOR_ERROR_FLASH_INIT :	初期化失敗
	R_NOR_ERR_PARAM :	引数エラー
補足	なし	

6.7.2 nor_flash_write_buf

nor_flash_write_buf

概要	NORフラッシュライト（複数ワード）	
ヘッダ	r_nor_flash.h	
宣言	uint32_t nor_flash_write_buf(uint32_t address, uint16_t *data, uint32_t cnt);	
説明	NORフラッシュに複数バイトデータを書き込む関数です。	
引数	uint32_t address	書き込みする先頭アドレスを指定します。
	uint16_t *data	書き込みするデータのポインタを指定します。
	uint32_t cnt	書き込みするデータ数を指定します。
リターン値	R_NOR_SUCCESS :	書き込み成功
	R_NOR_ERROR_FLASH_WRITE :	書き込み失敗
	R_NOR_ERR_PARAM :	引数エラー
補足	このコマンドはセクタの消去なし、およびチップ、セクタの境界を意識せず書き込みます。	

6.7.3 nor_flash_write_unit

nor_flash_write_unit

概要	NORフラッシュライト（1ワード）	
ヘッダ	r_nor_flash.h	
宣言	uint32_t nor_flash_write_unit(uint32_t address, uint16_t data)	
説明	NORフラッシュに1ワードデータを書き込む関数です。	
引数	uint32_t address	書き込みする先頭アドレスを指定します。
	uint16_t data	書き込みするデータを指定します。
リターン値	R_NOR_SUCCESS :	書き込み成功
	R_NOR_ERROR_FLASH_WRITE :	書き込み失敗
	R_NOR_ERR_PARAM :	引数エラー
補足	このコマンドを使用する前に該当セクタの消去をしてください。	

6.7.4 nor_flash_erase_sector

nor_flash_erase_sector

概要	NORフラッシュセクタ消去	
ヘッダ	r_nor_flash.h	
宣言	uint32_t nor_flash_erase_sector(uint32_t chip_number, uint32_t sect_no)	
説明	NORフラッシュの1セクタを消去する関数です	
引数	uint8_t chip_number :	消去するチップ No. を指定します。
	uint32_t sect_no :	消去するセクタ No. を指定します。
リターン値	R_NOR_SUCCESS :	消去成功
	R_NOR_ERROR_FLASH_ERASE :	消去失敗
	R_NOR_ERR_PARAM :	引数エラー
補足	なし	

6.7.5 nor_flash_erase_chip

nor_flash_erase_chip

概要	NORフラッシュチップ消去	
ヘッダ	r_nor_flash.h	
宣言	uint32_t nor_flash_erase_chip(uint32_t chip_number)	
説明	NORフラッシュ1個のデータを全部消去します。	
引数	uint8_t chip_number :	NORフラッシュのチップ No. を指定します。
リターン値	R_NOR_SUCCESS :	消去成功
	R_NOR_ERROR_FLASH_ERASE :	消去失敗
	R_NOR_ERR_PARAM :	引数エラー
補足	なし	

6.7.6 R_NOR_FLASH_Open

R_NOR_FLASH_Open

概要	ドライバAPI関数の開始	
ヘッダ	r_nor_flash.h	
宣言	int32_t R_NOR_FLASH_Open(void)	
説明	ドライバAPI関数を開始します。	
引数	なし	
リターン値	R_NOR_SUCCESS :	フラッシュ処理開始成功
	R_NOR_ERROR_FLASH_INIT :	フラッシュ初期化失敗
補足	なし	

6.7.7 R_NOR_FLASH_Erase_Chip

R_NOR_FLASH_Erase_Chip

概要	NORフラッシュチップ消去	
ヘッダ	r_nor_flash.h	
宣言	uint32_t R_NOR_FLASH_Erase_Chip (uint32_t chip_number);	
説明	NORフラッシュのチップ全体を消去します。	
引数	uint32_t chip_number	
リターン値	R_NOR_SUCCESS :	フラッシュ消去成功
	R_NOR_ERROR_FLASH_INIT :	フラッシュ初期化失敗
	R_NOR_ERROR_FLASH_ERASE :	フラッシュ消去失敗
補足	このコマンドは、チップ全体を消去するため数分かかります。	

6.7.8 R_NOR_FLASH_Erase_Sector

R_NOR_FLASH_Erase_Sector

概要	NORフラッシュセクタ消去	
ヘッダ	r_nor_flash.h	
宣言	uint32_t R_NOR_FLASH_Erase_Sector(uint32_t chip_number, uint32_t sectorAddr);	
説明	実装されているNORフラッシュの1セクタを消去します。	
引数	uint32_t chip_number :	チップ No. を指定します。
	uint32_t sectorAddr :	セクタ No. を指定します。
リターン値	R_NOR_SUCCESS :	フラッシュ消去成功
	R_NOR_ERROR_FLASH_INIT :	フラッシュ初期化失敗
	R_NOR_ERROR_FLASH_ERASE :	フラッシュ消去失敗
補足	なし	

6.7.9 R_NOR_FLASH_Write_Buf

R_NOR_FLASH_Write_Buf

概要	NORフラッシュライト（複数ワード）	
ヘッダ	r_nor_flash.h	
宣言	int32_t R_NOR_FLASH_Write_Buf(uint32_t address, uint16_t* data, uint32_t size)	
説明	NORフラッシュにデータを複数ワード書き込むAPI関数です。セクタおよびチップの境目を気にせず書き込めます。	
引数	uint32_t address :	書き込みするデータの先頭アドレスを指定します。
	uint16_t *data	書き込みするデータへのポインタを指定します。
	uint32_t size :	書き込みするデータのバイト数を指定します。
リターン値	R_NOR_SUCCESS :	書き込み成功
	R_NOR_ERR_PARAM :	引数エラー
補足	なし	

6.7.10 R_NOR_FLASH_Close

R_NOR_FLASH_Close

概要	ドライバAPI関数の終了	
ヘッダ	r_nor_flash.h	
宣言	void R_NOR_FLASH_Close (void)	
説明	ドライバAPI関数を終了します。	
引数	なし	
リターン値	なし	
補足	なし	

6.7.11 nor_flash_command

nor_flash_command

概要	コマンドを発行する関数	
ヘッダ	r_nor_flash.h	
宣言	void nor_flash_command (void)	
説明	ターミナルソフトにてメニュー形式および対話形式でNORフラッシュの読み書きなどのコマンドをやり取りする関数です。	
引数	なし	
リターン値	なし	
補足	なし	

6.7.12 main

main

概要	サンプルプログラムのメイン関数
ヘッダ	—
宣言	void main(void)
説明	サンプルプログラムのメイン処理です。 処理内容は7.1 メイン処理を参照してください。
引数	なし
リターン値	なし
補足	なし

7. フローチャート

7.1 メイン処理

図 7.1 にサンプルプログラムのメイン処理のフローチャートを示します。

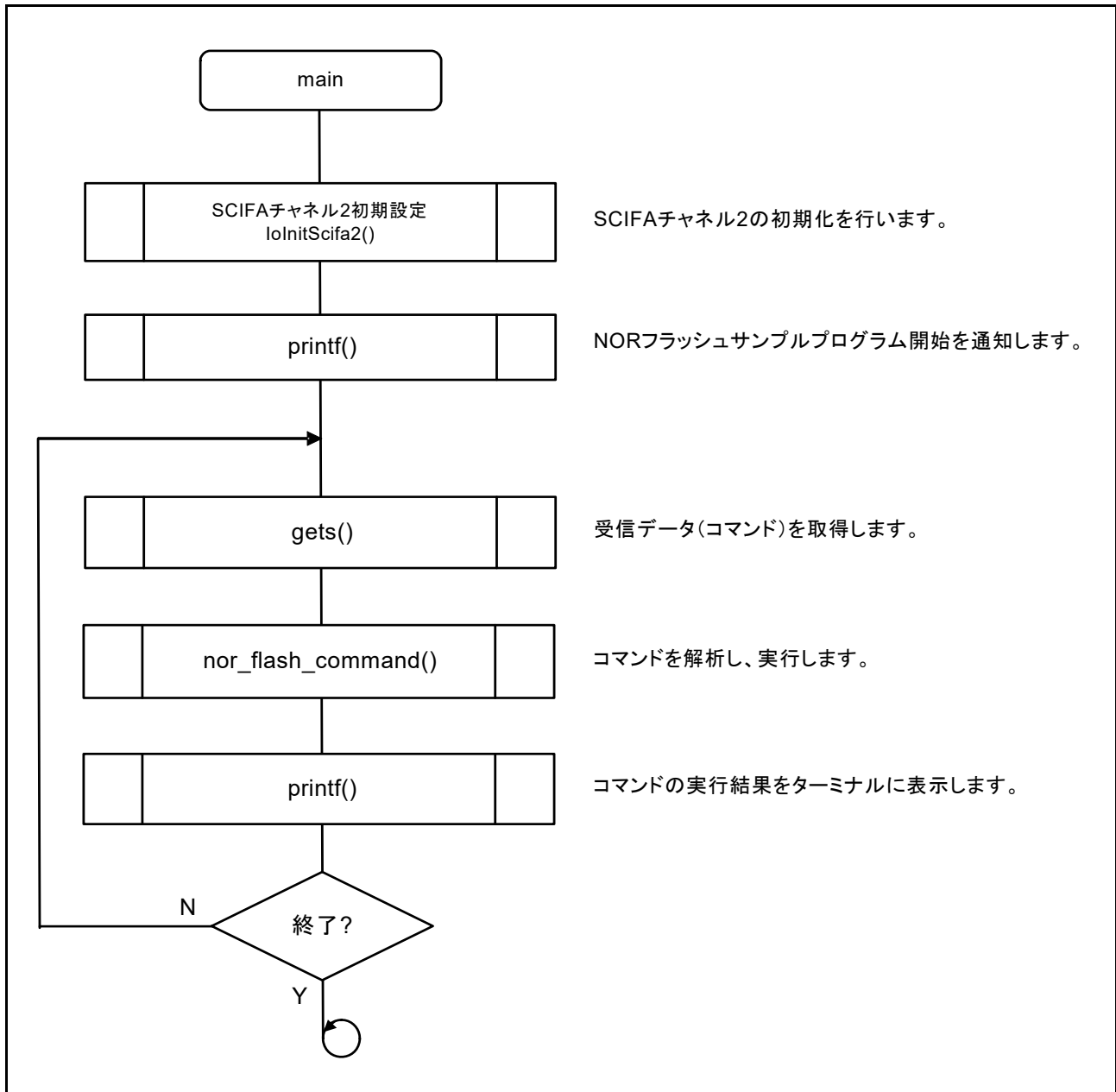


図 7.1 メイン処理

8. サンプルプログラムの動作例

PCのターミナルソフトウェア上でサンプルプログラムがどのように動作するかを示します。

RZ/T1評価ボードの電源ONまたはリセットで以下のメニュー画面が表示されます。

```
NOR Flash example program
```

```
[1] Read NOR Flash  
[2] Write NOR Flash  
[3] Erase NOR Flash Chip  
[4] Erase NOR Flash sector  
[0] Exit
```

```
>
```

[1]を選択すると、アドレスとサイズを入力するよう訊いてきます。サイズは128ワードが最大です。

```
NOR Flash example program
```

```
[1] Read NOR Flash  
[2] Write NOR Flash  
[3] Erase NOR Flash Chip  
[4] Erase NOR Flash sector  
[0] Exit
```

```
>1[Enter]
```

```
Input the top address and data size
```

```
>
```

アドレス：6000F000、サイズ：16を入力しますと、以下のように16ワードのデータが表示されます。

```
Input the top address and data size
```

```
>6000F000 16[Enter]
```

```
    00 02 04 06 08 0A 0C 0E  
6000F000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF  
6000F010 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

```
[1] Read NOR Flash  
[2] Write NOR Flash  
[3] Erase all NOR Flash  
[0] Exit
```

```
>
```

リードデータ表示後、再びメニュー画面が表示されますので、今度は [2] を選択します。

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit
```

```
>2[Enter]
```

```
Input the top address by hex (60000000 - 67FFFFFFE)
```

```
>
```

データを書き込む先頭アドレスを入力します。

```
Input the top address by hex (60000000 - 67FFFFFFE)
```

```
>60000000[Enter]
```

```
Input the size by hex (0 - 04000000 words)
```

書き込むデータサイズを入力します。

```
Input the size by hex (0 - 04000000 words)
```

```
>20[Enter]
```

```
Input the pointer address by hex (0 - FFFFFFFE)
```

書き込むデータのポインタアドレスを入力します。

```
Input the pointer address by hex (0 - FFFFFFFE)
```

```
>0[Enter]
```

```
Wrote NOR Flash
```

正常に書き込めたら「Wrote NOR Flash」のメッセージが出ます。

次にメニューの [3] を選択します。本当に消去してよいかを尋ねてきますので、[y] を選択します。

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit

>3[Enter]

Do you really want to erase NOR Flash? <y/n>y[Enter]
```

次に消去するチップ No. を尋ねてきますので、1 を選択します。

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash chip
[4] Erase NOR Flash sector
[0] Exit

>3

Do you really want to erase NOR Flash? <y/n>y

Input the chip number (1 or 2)

>1
```

1 チップを消去するには数分かかります。正常に消去できたら「Erased NOR Flash」の表示が出ます。

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash chip
[4] Erase NOR Flash sector
[0] Exit

>3

Do you really want to erase NOR Flash? <y/n>y

Input the chip number (1 or 2)

>1

Please wait few minutes to erase

Erased NOR Flash
```

次にメニューの [4] を選択します。本当に消去してよいかを尋ねてきますので、[y] を選択します。

```
[1] Read NOR Flash
[2] Write NOR Flash
[3] Erase NOR Flash Chip
[4] Erase NOR Flash sector
[0] Exit
```

```
>4[Enter]
```

```
Do you really want to erase NOR Flash? <y/n>y[Enter]
```

次に消去するチップ No. を尋ねてきますので、1 を選択します。

```
Do you really want to erase NOR Flash? <y/n>y
```

```
Input the chip number (1 or 2)
```

```
>1
```

次に消去するセクタ No. を尋ねてきますので、0 を選択します。

```
Input the chip number (1 or 2)
```

```
>1
```

```
Input the sector number by dec (0 - 511)
```

```
>0
```

正常に消去できたら「Erased NOR Flash sector」の表示が出ます。

```
Input the chip number (1 or 2)
```

```
>1
```

```
Erased NOR Flash sector
```

9. サンプルプログラムの入手先

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。

10. 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 評価ボード 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、IAR ホームページから入手してください。

(最新版を IAR ホームページから入手してください。)

Arm ソフトウェア開発ツール (Arm Compiler toolchain、Arm DS-5 等) に関しては、Arm ホームページから入手してください。

(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 エレクトロニクスソフトウェア開発ツール (e2studio 等) に関しては、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。

(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/contact/>

改訂記録

NORフラッシュサンプルプログラム 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
1.00	2015.12.22	—	新規発行
1.10	2016.06.14	要旨	
		1	制限事項追加
		2. 動作環境	
		5	表2.1 動作環境 統合開発環境 表記一部修正、追加
		6.1.1 プロジェクト設定	
		11	初期設定からの変更情報を追加
		6.2.1 サンプルプログラムのセクション配置	
		13	初期設定からのセクション配置/サイズの変更情報を追加
		6.4 定数/エラーコード一覧	
		16	表 6.3 サンプルプログラムで使用する定数値修正、追加
		10. 参考ドキュメント	
28	動作環境に関する情報入手先を追加		
全体			
—	誤記修正		
1.20	2017.04.05	2. 動作環境	
		4	表2.1 動作環境 統合開発環境の内容変更
		6. ソフトウェア説明	
—	6.2.4 必要メモリサイズ 削除		
1.30	2018.06.07	2. 動作環境	
		4	表2.1 動作環境 統合開発環境の内容変更
		5. ハードウェア説明	
		7	図5.1 ハードウェア構成例 モジュール名変更
		10. 参考ドキュメント	
24	ARM→Armに変更		

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/>