

## はじめに

本アプリケーションノートは、RSK+RZT1 に実装されている QSPI フラッシュメモリの設定とプログラムを e<sup>2</sup> studio 環境で行う方法について説明します。

RSK+RZT1 の詳細については、RSK+RZT1 ユーザーズマニュアル (R20UT3551JJ0101) を参照してください。

RZ/T1 の詳細については、RZ/T1 グループユーザーズマニュアル ハードウェア編 (R01UH0483JJ0100) を参照してください。

## 対象製品

RZ/T1

# 目次

1.	概要.....	3
1.1	アドレス空間.....	3
1.2	要約.....	5
1.3	必要なリソース.....	5
2.	QSPI フラッシュメモリからのブート処理.....	7
2.1	Serial モードでの QSPI ブートの開始.....	7
2.2	ローダプログラムの転送.....	7
2.3	QSPI ブートにおけるローダ用パラメータ.....	8
3.	QSPI フラッシュメモリのプログラミング.....	10
3.1	ユーザアプリケーションプログラムの転送.....	10
3.2	ブートルoaderセクション.....	11
4.	QSPI フラッシュメモリからのデータの読み出し.....	12
5.	ロードファイル.....	13
6.	バイナリファイルの作成.....	15
6.1	ユーザアプリケーションプログラムのプログラミング方法.....	15
6.2	QSPI フラッシュメモリからのコード実行方法.....	16
7.	e2 studio を使用した QSPI フラッシュメモリへのユーザアプリケーションプログラムの ロード.....	17
8.	ユーザアプリケーションプログラムのデバッグ構成について.....	19

## 1. 概要

QSPIフラッシュメモリは複数のデータ線をもつシリアルインタフェースデバイスです。データバスは1ビット、2ビット、4ビットを指定できます。

### 1.1 アドレス空間

RZ/T1のアドレス空間の詳細を図1.1に示します。

Cortex-M3 (R-IN Engine搭載製品)		Cortex-R4F		DMAC0/DMAC1		USB	
0000 0000h	Instruction RAM (512KB) (注2)	0000 0000h	ATCM (512KB) (注4)	0000 0000h	ATCM (512KB) (注4)	0000 0000h	ATCM (512KB) (注4)
0008 0000h	予約領域 (注1)	0008 0000h	予約領域 (注1)	0008 0000h	予約領域 (注1)	0008 0000h	予約領域 (注1)
0080 0000h	BTCM (32KB) (注4)	0080 0000h	BTCM (32KB) (注4)	0080 0000h	BTCM (32KB) (注4)	0080 0000h	BTCM (32KB) (注4)
0080 8000h	予約領域 (注1)	0080 8000h	予約領域 (注1)	0080 8000h	予約領域 (注1)	0080 8000h	予約領域 (注1)
0200 0000h	ATCM (512KB) (注4)						
0208 0000h	予約領域 (注1)						
0400 0000h	Instruction RAM (512KB) のミラー領域 (注1)	0400 0000h	Instruction RAM (512KB)	0400 0000h	Instruction RAM (512KB)	0400 0000h	Instruction RAM (512KB)
0408 0000h	予約領域 (注1)	0408 0000h	予約領域 (注1)	0408 0000h	予約領域 (注1)	0408 0000h	予約領域 (注1)
0800 0000h	Buffer RAM (注4) (128MB)	0800 0000h	Buffer RAM (注4) (128MB)	0800 0000h	Buffer RAM (注4) (128MB)	0800 0000h	Buffer RAM (注4) (128MB)
1000 0000h	SPIマスタID0バス空間 (シリアル・フラッシュ) (64MB)	1000 0000h	SPIマスタID0バス空間 (シリアル・フラッシュ) (64MB)	1000 0000h	SPIマスタID0バス空間 (シリアル・フラッシュ) (64MB)	1000 0000h	SPIマスタID0バス空間 (シリアル・フラッシュ) (64MB)
1400 0000h	予約領域 (注1)	1400 0000h	予約領域 (注1)	1400 0000h	予約領域 (注1)	1400 0000h	予約領域 (注1)
2000 0000h	Data RAM (512KB) (注2)	2000 0000h	Data RAM (512KB)	2000 0000h	Data RAM (512KB)	2000 0000h	Data RAM (512KB)
2008 0000h	予約領域 (注1)	2008 0000h	予約領域 (注1)	2008 0000h	予約領域 (注1)	2008 0000h	予約領域 (注1)
2200 0000h	BitBand Alias Area0 (16MB) (注2)	2200 0000h	Data RAM (512KB) のミラー領域 (注1)				
2300 0000h	予約領域 (注1)	2400 0000h	Instruction RAM (512KB) のミラー領域 (注1)				
		2408 0000h	予約領域 (注1)				
		3000 0000h	SPIマスタID0バス空間 (シリアル・フラッシュ) (64MB) のミラー領域 (注1)				
		3400 0000h	予約領域 (注1)				
4000 0000h	周辺IOレジスタ (1MB) のミラー領域 (注1)	4000 0000h	予約領域 (注1)			4000 0000h	予約領域 (注1)
4010 0000h	予約領域 (注1)						
4200 0000h	BitBand Alias Area1 (32MB) (注2)		外部アドレス空間 (CS0) (64MB) のミラー領域 (注1)				
4400 0000h	予約領域 (注1)	4400 0000h	外部アドレス空間 (CS1) (64MB) のミラー領域 (注1)				
		4800 0000h	外部アドレス空間 (CS2) (64MB) のミラー領域 (注1)				
		4C00 0000h	外部アドレス空間 (CS3) (64MB) のミラー領域 (注1)				
		5000 0000h	外部アドレス空間 (CS4) (64MB) のミラー領域 (注1)				
		5400 0000h	外部アドレス空間 (CS5) (64MB) のミラー領域 (注1)				
		5800 0000h	予約領域 (注1)				
6000 0000h	外部アドレス空間 (CS0) (64MB)	6000 0000h	外部アドレス空間 (CS0) (64MB)	6000 0000h	外部アドレス空間 (CS0) (64MB)		
6400 0000h	外部アドレス空間 (CS1) (64MB)	6400 0000h	外部アドレス空間 (CS1) (64MB)	6400 0000h	外部アドレス空間 (CS1) (64MB)		
6800 0000h	外部アドレス空間 (CS2) (64MB)	6800 0000h	外部アドレス空間 (CS2) (64MB)	6800 0000h	外部アドレス空間 (CS2) (64MB)		
6C00 0000h	外部アドレス空間 (CS3) (64MB)	6C00 0000h	外部アドレス空間 (CS3) (64MB)	6C00 0000h	外部アドレス空間 (CS3) (64MB)		
7000 0000h	外部アドレス空間 (CS4) (64MB)	7000 0000h	外部アドレス空間 (CS4) (64MB)	7000 0000h	外部アドレス空間 (CS4) (64MB)		
7400 0000h	外部アドレス空間 (CS5) (64MB)	7400 0000h	外部アドレス空間 (CS5) (64MB)	7400 0000h	外部アドレス空間 (CS5) (64MB)		
7800 0000h	予約領域 (注1)	7800 0000h	予約領域 (注1)	7800 0000h	予約領域 (注1)		
A000 0000h	周辺IOレジスタ (1MB) (注1) (注2)	A000 0000h	周辺IOレジスタ (1MB) (注1) (注2)	A000 0000h	周辺IOレジスタ (1MB) (注1) (注2)		
A010 0000h	予約領域 (注1)	A010 0000h	予約領域 (注1)	A010 0000h	予約領域 (注1)		
E000 0000h	Cortex-M3 Private BUS (1MB) (注2)						
E010 0000h	予約領域 (注1)						
E800 0000h	デバッグ用領域 (64KB)	E800 0000h	デバッグ用領域 (64KB)	E800 0000h	デバッグ用領域 (64KB)		
E801 0000h	予約領域 (注1)	E801 0000h	予約領域 (注1)	E801 0000h	予約領域 (注1)		
FFFF 0000h	ブート専用領域 (32KB)	FFFF 0000h	ブート専用領域 (32KB)	FFFF 0000h	ブート専用領域 (32KB)		
FFFF 8000h	予約領域 (注1)	FFFF 8000h	予約領域 (注1)	FFFF 8000h	予約領域 (注1)		

- 注1. Cache対象 (それ以外の領域は、MPUでキャッシュを有効にしないでください。)
- 注2. Cortex-M3からのみアクセス可能な空間
- 注3. Cortex-M3 SRAM空間 (Extra-RAM)
- 注4. Cortex-R4F密結合メモリ空間
- 注5. 周辺空間
- 注6. A00E 0000h-A00F FFFFhは、予約領域
- 注7. A00E 0000hからA00F FFFFhの空間にはアクセスできません (400E 0000h-400FFFFFhの空間だけアクセスできます)。
- 注8. 詳細は、「28.3.1.3 Buffer Allocator」を参照してください。
- 注. 予約領域は、アクセスしないでください。

図 1.1 RZ/T1 アドレスマップ

## 1.2 要約

ブート機能は、RZ/T1 のシステムリセットに続いて実行されるプログラムです。ローダプログラムのロードや、必要に応じてマイコン内に割り当てられたプログラムメモリ (TCM) にユーザアプリケーションプログラムをコピーし、ユーザアプリケーションプログラムに制御を引き継ぎます。RZ/T1 には QSPI チャンネルが 1 本あり、1 個の QSPI フラッシュメモリに接続できます。

RZ/T1 は 3 種類のブートモードに対応しています。

- SPI ブートモード : SPI マルチ I/O バス空間に接続されたシリアルフラッシュメモリからブート
- 16 ビットバスブートモード : CS0 空間に接続された NOR フラッシュメモリ (バス幅 16 ビット) からブート
- 32 ビットバスブートモード : CS0 空間に接続された NOR フラッシュメモリ (バス幅 32 ビット) からブート

## 1.3 必要なリソース

RSK+RZT1 上の QSPI フラッシュメモリからブートする際に必要な設定は以下のとおりです。

表 1.1 ブートモード端子 SW4 の構成

SW4					
SW4.1*	SW4.2*	SW4.3*	SW4.4	SW4.5	SW4.6
ON	ON	ON	ON	ON	OFF

注. SW4.1はMD0に、SW4.2はMD1に、SW4.3はMD3端子に接続されます。

表 1.2 QSPI 通信端子

QSPI 信号名	RSK+RZT1 信号名	機能
SPBSSL	P60	スレーブセレクト信号出力端子
SPBCLK	P62	クロック出力端子
SPBIO0	P63	データ 0 入出力端子
SPBIO1	P64	データ 1 入出力端子
SPBIO2	P65	データ 2 入出力端子
SPBIO3	P61	データ 3 入出力端子

図 1.2 にブート処理の動作概要を示します。

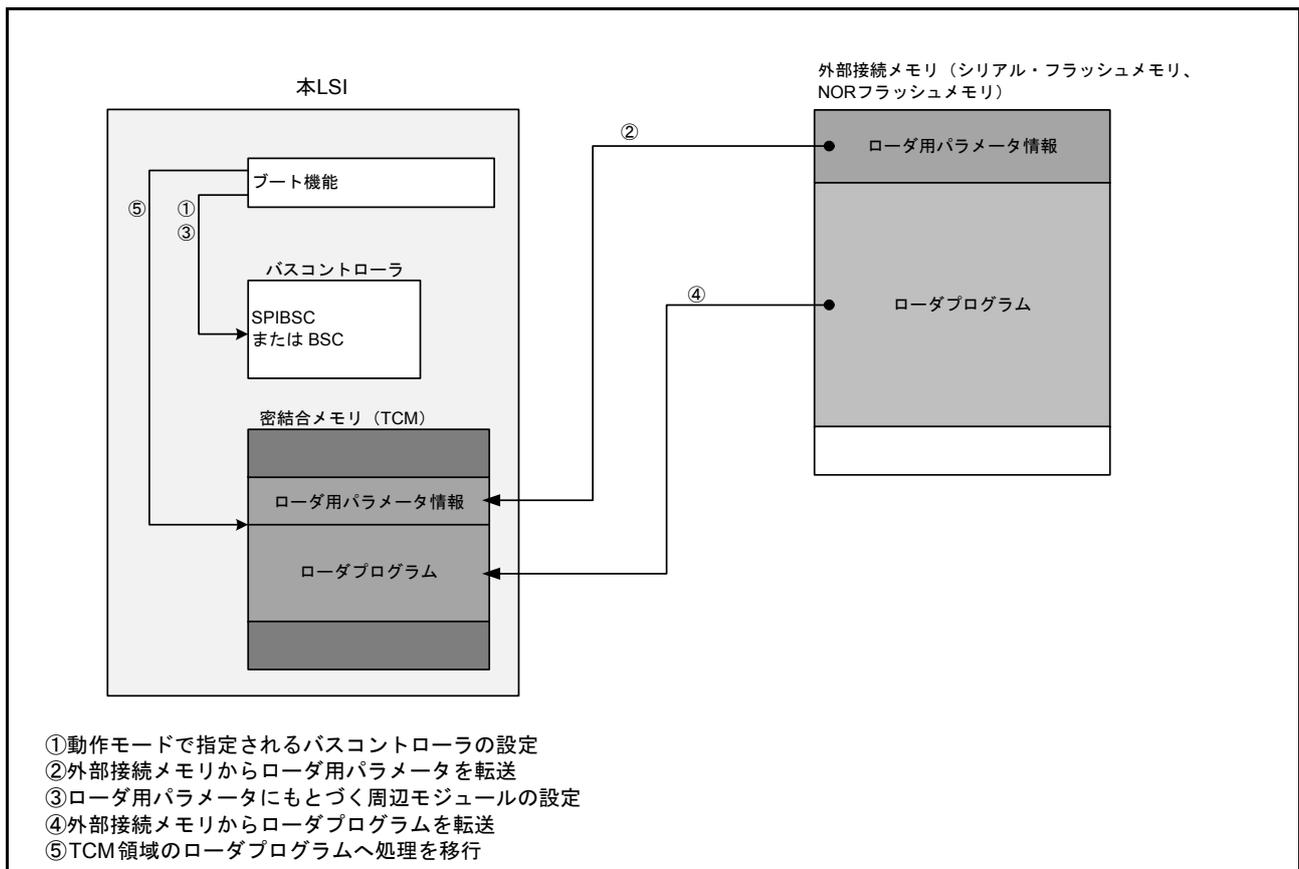


図 1.2 ブート処理の動作概要

## 2. QSPI フラッシュメモリからのブート処理

### 2.1 Serial モードでの QSPI ブートの開始

リセット後、RZ/T1 はハイベクタ 0xFFFF0000 に配置されたブート機能を実行します。これにより、QSPI フラッシュメモリが Serial モードおよび外部アドレス空間リードモードで構成され、接続された QSPI フラッシュメモリから直接データを読み出すことが可能となります。

### 2.2 ローダプログラムの転送

ローダプログラムは、ブート機能により BTCM へとコピーされます。ブート機能はローダ用パラメータを通じて、QSPI フラッシュメモリの先頭 75 バイト (0x4B) に格納されたローダプログラムに関する情報を取得します。

### 2.3 QSPI ブートにおけるローダ用パラメータ

ローダ用パラメータは、ローダプログラム情報、ブート処理を高速化するためのキャッシュ設定、バスステートコントローラ設定 (SPIBSC または BSC) などの、ユーザシステムの構成情報を含みます。ローダ用パラメータは事前に QSPI フラッシュに格納しておく必要があります。RZ/T1 の TCM と QSPI フラッシュメモリ間の関係を図 2.1 に示します。TCM 領域として、ユーザアプリケーションプログラムを配置する ATCM と、ローダプログラムを配置する BTCM の 2 つが存在します。

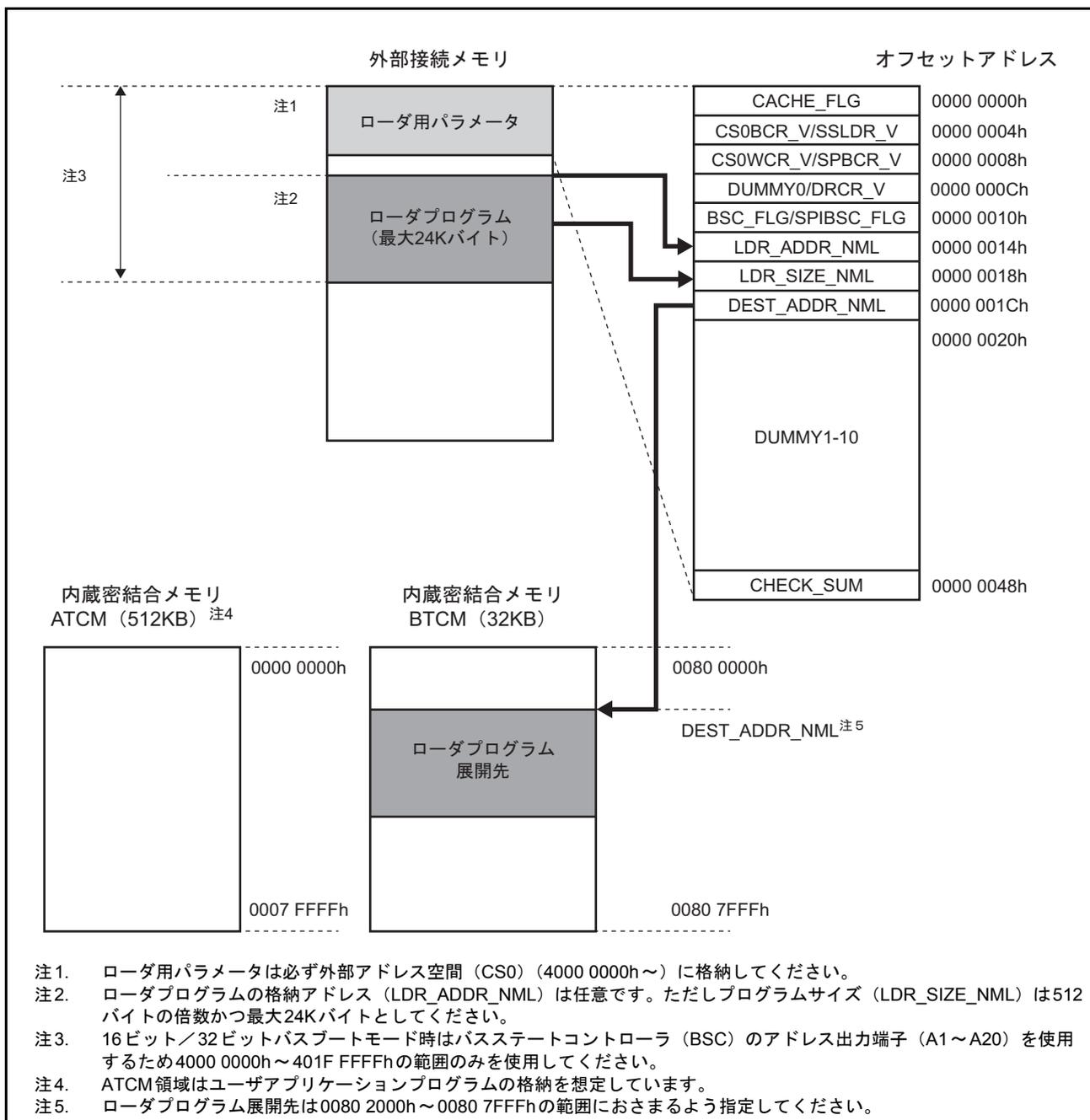


図 2.1 ローダ用パラメータ、ローダプログラムのメモリ配置

ローダ用パラメータは、SPI マルチ I/O バス空間のアドレス 0x30000000 から格納されます。ラベル LDR\_ADDR\_NML はローダプログラムの QSPI フラッシュメモリ内での格納アドレスを指定し、DEST\_ADDR\_NML はローダプログラムをコピーして実行する際の BTCM 内での格納アドレスを指定します。LDR\_ADDR\_NML の設定値は 0x3000004C、DEST\_ADDR\_NML の設定値は 0x00802000 です。

ユーザアプリケーションプログラムを QSPI フラッシュメモリから ATCM へ転送する前に、ローダプログラムは通信速度を最大にするため、QSPI フラッシュメモリのモードを Quad モードに変更します。

注：SPI マルチ IO バス空間用の外部アドレス空間は、全体で 4GB あり、0x10000000 から 0x13FFFFFF (0x30000000 から 0x33FFFFFF のミラー領域) に内部でマッピングされています。

RSK+RZT1 に実装された QSPI フラッシュメモリの総容量は 64MB です。

### 3. QSPI フラッシュメモリのプログラミング

以下の2通りの方法を用いて、RSK+RZT1に実装されたQSPIフラッシュメモリをプログラミングします。インストーラ実行後、ソフトウェアパッケージをユーザPCにコピーして使用してください。

- サンプルコード
  - e<sup>2</sup> studio環境でのサンプルプログラムSystem\_Boot\_Loader\_QSPIを用いたプログラミング及びデバッグ
  - QSPIフラッシュメモリへプログラミングするローダプログラムおよびユーザアプリケーションプログラムも同梱
- バッチファイル
  - ローダプログラムおよびユーザアプリケーションプログラムをQSPIフラッシュメモリへプログラミング
  - 統合開発環境とは独立したファイル

同梱のユーザアプリケーションプログラムは、アドレス0x00000000から始まるATCM内に展開され実行されます。このユーザアプリケーションのバイナリは、RSK+RZT1の製造時にバッチファイルを用いてQSPIフラッシュメモリへ書き込みしています。

#### 3.1 ユーザアプリケーションプログラムの転送

ローダプログラムは、QSPIフラッシュメモリ内のユーザアプリケーションプログラムを開始アドレス0x30010000から検査します。検査対象は、ユーザアプリケーションプログラムの開始アドレス、終了アドレス、実行アドレス、ローダプログラムのstring変数です。この4項目は、ユーザアプリケーションプログラムのe<sup>2</sup> studioプロジェクトのstart.asmファイル内にあります。このファイルの一部を以下に示します。

```
.text
.code 32

.global start
.func start
start:
    LDR pc, =reset_handler
    LDR pc, =undefined_handler
    LDR pc, =svc_handler
    LDR pc, =prefetch_handler
    LDR pc, =abort_handler
    LDR pc, =reserved_handler
    LDR pc, =irq_handler
    LDR pc, =fiq_handler
code_start:
    .word start
code_end:
    .word end
code_execute:
    .word execute
    .string ".BootLoad_ValidProgramTest."
    .align 4
    .end
```

“.string”は、ローダプログラムがユーザアプリケーションプログラムを検証するためのシグネチャ変数(signature marker)です。

“start”は、ユーザアプリケーションプログラムのベクタテーブルをロードするアドレスを指定します。ラ

ベル“code\_start”および“code\_end”には、ユーザアプリケーションプログラム全体の開始アドレス、終了アドレスおよびそのベクタアドレスを指定する変数が含まれています。開始アドレスと終了アドレスは、以下のラベルを用いてユーザアプリケーションプログラムのリンカファイル(.ld)内で指定されます。

```
ENTRY(start)
```

```
PROVIDE(end = .);
```

ラベル“code\_execute”には実行開始アドレスを示す変数“execute”が含まれます。

“execute”ラベルはリンカファイルにも以下のように指定されます。

```
execute = .;
```

位置を示す定数や .string 変数が、上記のように start.asm ファイルの上記ベクタテーブルの直後に存在しない場合、構成は無効とみなされます。このとき、RSK+RZT1 上の LED1 が長いパルス（2 秒）で 1 回点滅したあと、短いパルス（0.5 秒）で 1 回点滅し、エラーを知らせます。構成が有効の場合、ローダプログラムは ATCM 内のユーザアプリケーションプログラムの開始アドレスを確認します。

リンカファイルは、RSK+RZT1 のサンプルプロジェクトごとの“complier\_specific”ディレクトリに含まれています。

転送が完了すると、ユーザアプリケーションプログラムは実行アドレス 0x00000000 から実行します。

GNU リンカファイルの詳細は e<sup>2</sup> studio のヘルプ>検索からキーワード“GNU linker”を入力して参照してください。

ブート処理に関する詳細は、RZ/T1 グループ ユーザーズマニュアル ハードウェア編（R01UH0483JJ）の『動作モード』の章を参照してください。

## 3.2 ブートローダセクション

ローダプログラムは、ユーザアプリケーションプログラムとは別のセクションに配置されます。リンカファイル中の主要なセクションを以下に説明します。

.loader\_param セクション：

このセクションにはローダ用パラメータが含まれます。QSPI フラッシュメモリへロードされますが、BTCM へのコピーはされません。ロードメモリアドレスは 0x30000000 です。

.reset セクション：

このセクションにはベクタテーブルが含まれます。QSPI フラッシュメモリのアドレス 0x3000004C にロードされ、実行アドレス 0x00802000 に再配置されます。

注． ローダプログラムは、サンプルプロジェクト System\_Boot\_Loader\_QSPI を HardwareDebug 構成またはリリース構成でビルドし、プロジェクトのルートディレクトリにある Program\_QSPI\_Loader.bat を実行することでプログラムできます。RSK+RZT1 は、ローダプログラムを事前にプログラムした状態で出荷されます。

### 4. QSPI フラッシュメモリからのデータの読み出し

QSPI フラッシュメモリへのデータアクセスは1ビット、2ビット、4ビットモードに設定可能です。各モードの違いを以下の図に示します。データサイズが2倍になると、アクセスサイクルは2分の1に減少します。

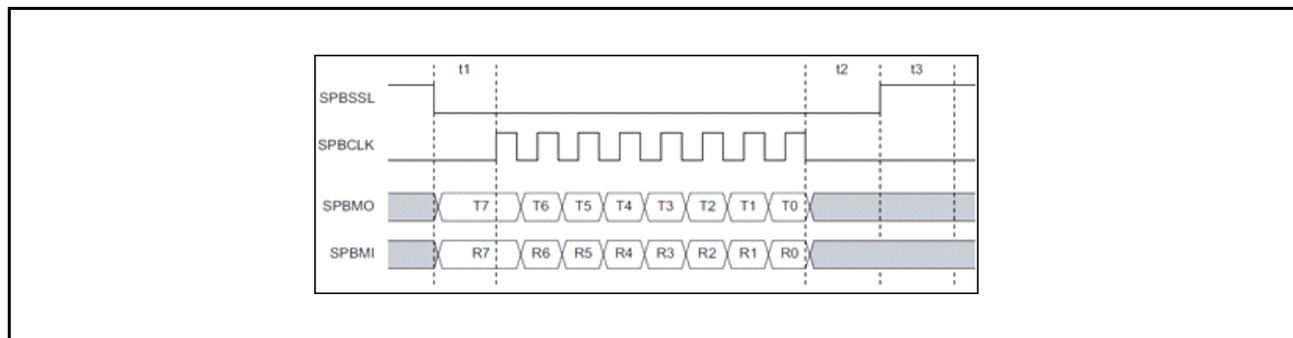


図 4.1 1ビットデータの転送フォーマット例

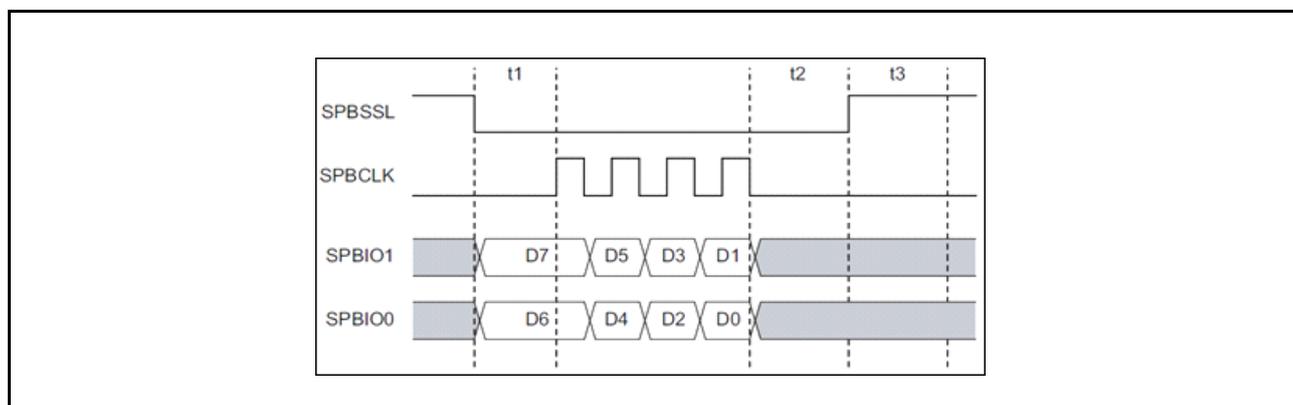


図 4.2 2ビットデータの転送フォーマット例

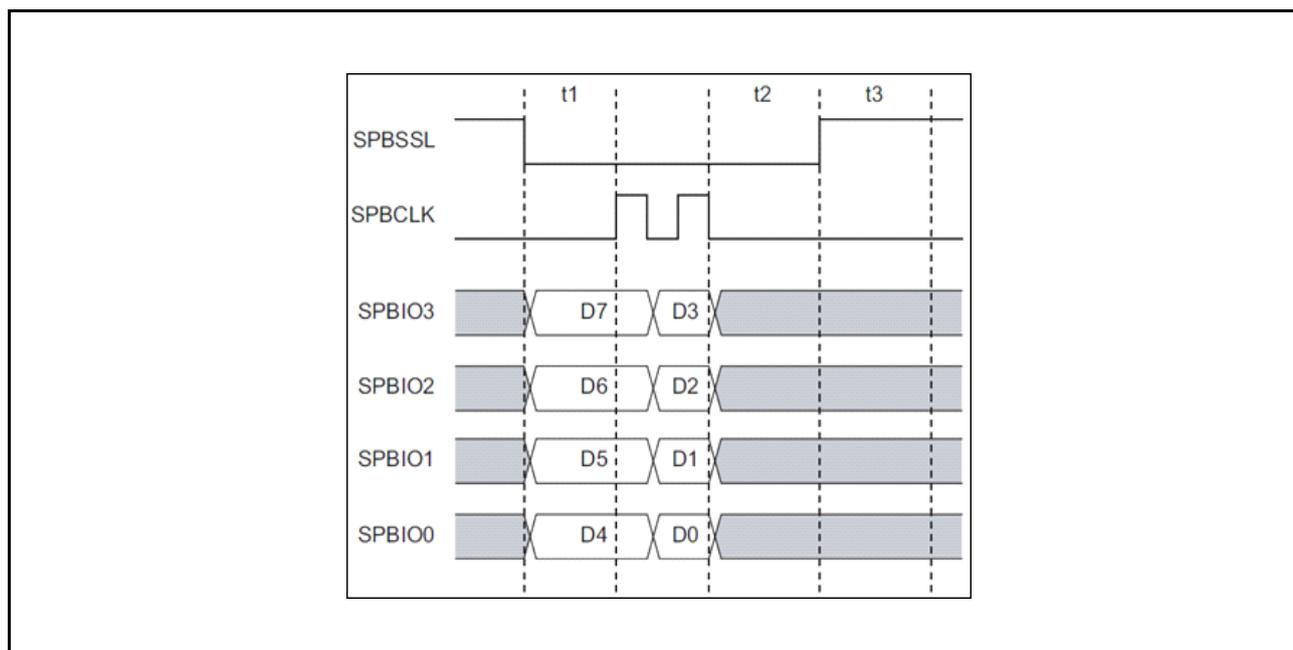


図 4.3 4ビットデータの転送フォーマット例

## 5. ロードファイル

サンプルプロジェクト System\_Boot\_Loader\_QSPI には、2つのリンカファイルまたはロードファイルがあり、ローダプログラムの保存場所と実行場所の指定に用います。これらのファイルは BTCM で実行されます。QSPI フラッシュメモリへロードされたプログラムは、起動時に RZ/T1 のブート機能によって BTCM へとコピーされ、BTCM から実行されます。ロードファイルは「プロジェクト・エクスプローラ」のツリービューのプロジェクトフォルダにあります。

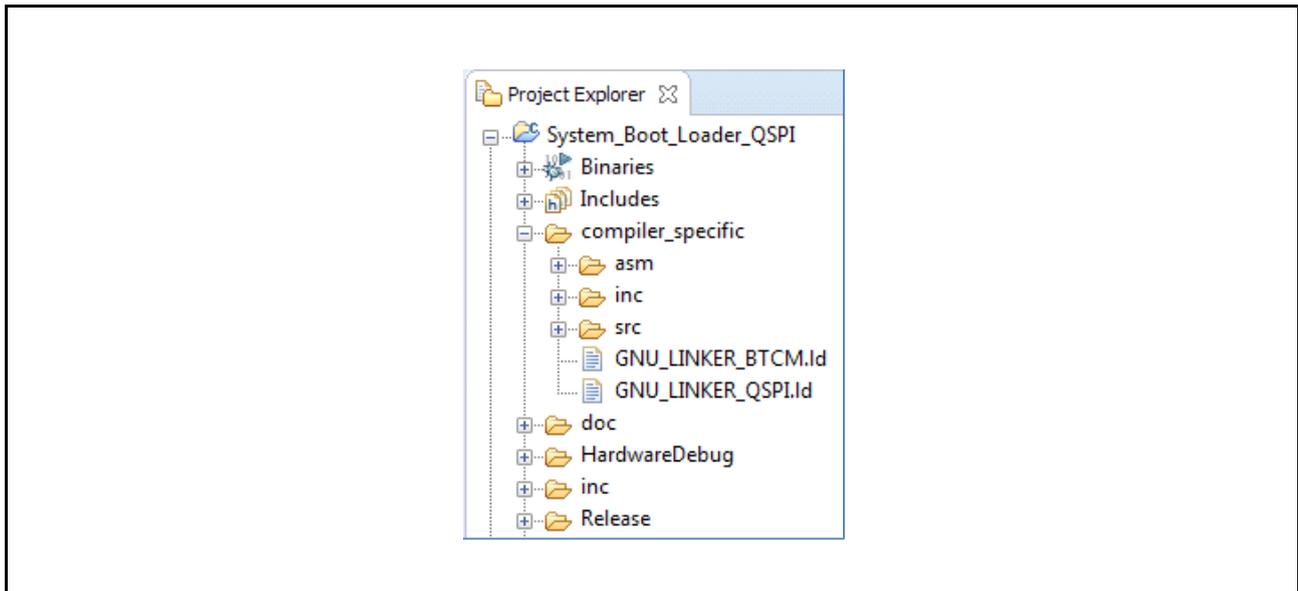


図 5.1 ロードファイル

ロードファイル GNU\_LINKER\_QSPI.Id および GNU\_LINKER\_BTCM.Id には以下の行が含まれます。

```
LDR_PARAM_ROM      = 0x30000000;
LDR_EXEC_BASE      = 0x00802000;
USER_APP_ROM       = 0x30010000;
```

- LDR\_PARAM\_ROM はローダ用パラメータが格納される QSPI フラッシュメモリのアドレスを指定。
- LDR\_EXEC\_BASE はローダプログラムが実行される BTCM メモリのアドレスを指定。
- USER\_APP\_ROM はユーザアプリケーションプログラムが格納される QSPI フラッシュメモリのアドレスを指定。

ロードファイルを使用して、ユーザはローダプログラムのロードアドレス、プログラムの QSPI フラッシュメモリ内の格納アドレス、プログラムが実行される再配置先のアドレスを指定することができます。QSPI フラッシュメモリに格納されたユーザアプリケーションプログラムを BTCM へコピーする再配置命令の例を以下に示します。

```
.reset LDR_EXEC_BASE : AT (LDR_PARAM_ROM + SIZEOF(.loader_param) )
```

ロードファイル GNU\_LINKER\_BTCM.Id は、LDR\_EXEC\_BASE へとロードされ実行されるため、ロードベースアドレスを含みません。また、パラメータのセクションも含みません。ロードファイル GNU\_LINKER\_ATCM.Id は、ローダプログラム以外の全てのサンプルプロジェクトにあります。

ロードファイルを変更するには、「プロジェクト・エクスプローラ」ツリービュー下のプロジェクトフォ

ルダをクリックし、ファイル>プロパティ> C/C++ ビルド>設定> Linker >その他を選択してください。



図 5.2 ロードファイルの設定

## 6. バイナリファイルの作成

ユーザアプリケーションプログラムをプログラミングするためには、バイナリ形式 (.bin) のプログラムファイルが必要です。HardwareDebug ビルドおよび Release ビルドは、初期状態では拡張子 .x および .mot を付けた ELF ファイルを生成するよう構成されています。これらのビルド構成を、拡張子 .bin を付けたバイナリファイルを生成するよう変更します。設定は e<sup>2</sup> studio のビルド後のステップによって行われます。

1. e<sup>2</sup> studio 内の「プロジェクト・エクスプローラー」ツリービュー下の対象プロジェクトをクリックする。
2. メニューバーから、「ファイル>プロパティ」の順に選択する。
3. 「プロパティ」ダイアログから、「C/C++ ビルド>設定」の順に選択する。
4. 目的のビルドの構成（「HardwareDebug」または「Release」）を選択する。
5. 「ビルド・ステップ」タブを選択する。
6. 以下のように「ビルド後のステップ」を変更する。  
arm-none-eabi-objcopy -O binary \${ProjName}.x \${ProjName}.bin&
7. 「OK」をクリックする。

### 6.1 ユーザアプリケーションプログラムのプログラミング方法

ユーザアプリケーションプログラムまたは提供されたサンプルコードの中から所望のコードを構築します。以下のファイルを System\_Boot\_Loader\_QSPI フォルダからユーザアプリケーションまたはサンプルコードのルートフォルダにコピーしてください。

```
Program_QSPI_Loader.bat  
Program_QSPI_Loader_HDB.Command  
Program_QSPI_Loader_REL.Command
```

Program\_QSPI\_Loader.bat および Program\_QSPI\_Loader\_HDB.command、  
Program\_QSPI\_Loader\_REL.command ファイル中の以下を変更します。この時、行番号を変更しないよう注意してください。

“Tutorial” の記述箇所をユーザアプリケーションの名前(“ ”なし)に置き換える

バッチファイル Program\_QSPI\_Loader.bat を実行します。

J-Link を用いてプログラムする際は、ロードされるローダプログラムが、QSPI フラッシュメモリの既存のローダプログラムと一致するかどうかを確認します。一致する場合、“loadbin” 命令は、プログラミングをスキップします。ユーザアプリケーションプログラムのロードが完了した後、RSK+RZT1 の再起動(電源の切断・再投入)が必要となります。

## 6.2 QSPI フラッシュメモリからのコード実行方法

RSK+RZT1 の起動時に QSPI フラッシュメモリにプログラムされているユーザアプリケーションプログラムを実行することができます。この機能を使用するには、QSPI フラッシュメモリにローダプログラムがプログラムされている必要があります。出荷時点ではローダプログラムがプログラムされていますが、これに変更を加えた場合は、Program\_QSPI\_Loader.bat を使用して再プログラムしてください。

起動時、ローダプログラムは Tutorial プログラムの存在を確認し、実行するため ATCM へと転送します。コードが存在しない場合、ローダプログラムは LED1 を継続的に点滅させます。プログラムのプログラム方法は以下のとおりです。

- e<sup>2</sup> studio のデバッグセッションが接続中であれば切断する。
- Windows エクスプローラを開き、System\_Boot\_Loader\_QSPI プロジェクトのディレクトリを選択する。
- Program\_QSPI\_Loader.bat ファイルを実行する。
- 「Option 1 – HardwareDebug」もしくは「Option 2 – Release」を選択する。これにより QSPI フラッシュメモリにローダプログラムがプログラムされます。
- RSK+RZT1 を再起動 (電源の切断・再投入) すると、Tutorial プログラムが実行されます。

## 7. QSPI フラッシュメモリへのユーザアプリケーションプログラムのロード

RSK+RZT1 で用いるユーザアプリケーションプログラムのビルド構成の初期値は ATCM に直接ロードする構成となっています。QSPI フラッシュメモリへロードするためには、追加設定が必要となります。

プロジェクトがバイナリファイルを生成するように構成されていることを確認します。

以下の順に操作してください。

プロジェクトのプロパティ > C/C++ ビルド > 設定 > 構成 : Release > 「ビルド・ステップ」タブを選択し、「ビルド後のステップ コマンド:」を指定してください。



何も指定されていない場合、以下のようにコマンドを追加してください。

**arm-none-eabi-objcopy -O binary \${ProjName}.x \${ProjName}.bin&**

「適用」をクリックします。

「OK」をクリックします。

プロジェクトをビルドします。

以下の手順は、ユーザアプリケーションプログラムを QSPI にプログラムする方法の一つです。この操作は、0x30010000 に書きこまれている既存のアプリケーションプログラムを上書きします。

「デバッグの構成」を開きます。

目的の Release 構成（以下では、Tutorial Release を例に説明します）をクリックします。

「Startup」タブをクリックします。

「コマンドを実行」フィールドに命令を追加します。

以下の命令を先頭に追加します。

**restore Release/Tutorial.bin binary 0x30010000**

別のユーザアプリケーションプログラムを使用する場合は、「Tutorial」を該当のプログラム名に置き換えてください。

「適用」をクリックし、変更を保存します。

「デバッグ」をクリックします。

Segger J-Link のプログレスバーが短時間表示され、ユーザアプリケーションプログラムが QSPI フラッシュメモリにプログラムされたことを示します。

e2 studio の接続を解除します。

リセットスイッチ (RESET) を押下します。

ローダプログラムによってプログラムされたユーザアプリケーションプログラムが ATCM へロードされ、実行されます。

対象製品の接続やユーザアプリケーションプログラムのダウンロードに関する情報はチュートリアルマニュアル (R20UT3243JJ) を参照してください。

## 8. ユーザアプリケーションプログラムのデバッグ構成について

デバッグ構成内の各コマンドについて、Tutorial サンプルを用いて以下に説明します。

-exec-interrupt

CPU が現在実行中の動作を停止するための停止信号を送信します。

-data-write-memory-bytes --thread 1 0 "FF" 524288

524288 バイトのデータ (0xFF) をアドレス 0x0 から書き込みます。

-data-write-memory-bytes --thread 1 8396800 "FF" 22784

22784 バイトのデータ (0xFF) をアドレス 0x802000 (8396800) から書き込みます。

load applications/RAM\_DEBUG\_JUMP.x

RAM\_DEBUG\_JUMP.x は 0x802000 にロードされた小容量のプログラムです。“jump-to” 命令を行い、プログラムカウンタ (PC) を、Tutorial などのサンプルを実行開始する ATCM の先頭アドレスにセットします。

load Release/Tutorial.x

Tutorial サンプルプログラムをロードします。

print /x \$pc

e<sup>2</sup> studio のコンソールウィンドウ内に PC の現在のアドレスを示します。

set \$pc = 0xFFFF0000

プログラムカウンタをハイベクタ領域 (リセットベクタアドレス) にセットします。

stepi 6

ステップ命令を 6 回実行します。これにより、Cortex R4F の初期命令の一部が実行可能となります。

-exec-interrupt

CPU が現在実行中の動作を停止するための停止信号を送信します。

set \$pc = 0x802000

プログラムカウンタのポインタを、アドレス 0x802000 のプログラム RAM\_DEBUG\_JUMP.x にセットします。

stepi 6

ステップ命令を 6 回実行します。これにより、プログラムカウンタが ATCM(アドレス 0x0) の開始点にセットされます。

-exec-interrupt

CPU が現在実行中の動作を停止するための停止信号を送信します。

## 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	QSPIフラッシュブートローダ 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	2016.09.16	—	初版発行

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、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い合わせください。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/>