

## RX65N グループ

### Renesas Starter Kit+ for RX65N ブートローダサンプル アプリケーションノート

ルネサス 32 ビットマイクロコンピュータ  
RX ファミリ / RX600 シリーズ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。  
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。



## 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、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。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

# このマニュアルの使い方

## 1. 目的と対象者

このアプリケーションノートは、システムブートローダ (System\_Bootloader) サンプルがどのように機能し、サンプルが RX65N に基づいたシステム上でどのように開発されるかを理解していただくためのドキュメントです。

RSK+RX65N プラットフォーム上でシステム開発するユーザを対象にしています。

RX65N マイクロコントローラに関する詳細情報は、ユーザーズマニュアル ハードウェア編やサンプルコードを参照してください。

RSK+RX65N では次のドキュメントを用意しています。ドキュメントは最新版を使用してください。最新版はルネサスエレクトロニクスのホームページに掲載されています。

ドキュメントの種類	記載内容	資料名	資料番号
ユーザーズマニュアル	RSK ハードウェア仕様の説明	RSK+RX65N ユーザーズマニュアル	R20UT3558JG
チュートリアルマニュアル	RSK および開発環境のセットアップ方法とデバッグ方法の説明	RSK+RX65N チュートリアルマニュアル	R20UT3559JG
クイックスタートガイド	A4 紙一枚の簡単なセットアップガイド	RSK+RX65N クイックスタートガイド	R20UT3560JG
コード生成支援ツール チュートリアルマニュアル	コード生成支援ツールの使用方法の説明	RSK+RX65N コード生成支援ツール チュートリアルマニュアル	R20UT3561JG
回路図	CPU ボードの回路図	RSK+RX65N CPU ボード回路図	R20UT3557EG
ユーザーズマニュアル ハードウェア編	ハードウェアの仕様 (ピン配置、メモリマップ、周辺機能の仕様、電気的特性、タイミング) と動作説明	RX65N グループ ユーザーズマニュアル ハードウェア編	R01UH0590JJ
アプリケーションノート	Firmware Integration Technology (FIT) を使用したフラッシュモジュールの説明	RX Family Flash Module Using Firmware Integration Technology	R01AN2184EU (英文のみ)
アプリケーションノート	Simple Flash API の説明	Using the Simple Flash API for RX without the r_bsp Module	R01AN1890EU (英文のみ)
アプリケーションノート	ブートローダサンプルコードの説明	RSK+RX65N ブートローダ アプリケーションノート	R20AN0392JG (本資料)

## 2. 略語および略称の説明

略語／略称	英語名	備考
API	Application Program Interface	アプリケーションプログラムインタフェース
bps	Bits per second	転送速度を表す単位、ビット/秒
Bootloader	-	本サンプルではアプリケーション更新機能を持つローダ
CGC	Clock Generation Circuit	クロック発生回路
CPU	Central Processing Unit	中央処理装置
CRC	Cyclic Redundancy Check	巡回冗長検査
E1	Renesas On-chip Debugging Emulator	ルネサスオンチップデバッグエミュレータ
FSL	Flash Self-programming Library	フラッシュセルフプログラミングライブラリ
GUI	Graphical User Interface	グラフィカルユーザインタフェース
I <sup>2</sup> C (IIC)	Philips™ Inter-Integrated Circuit Connection Bus	フィリップス社が提唱したシリアル通信方式
IRQ	Interrupt Request	割り込み要求
ISR	Interrupt Service Routine	割り込みサービスルーチン
LCD	Liquid Crystal Display	液晶ディスプレイ
LED	Light Emitting Diode	発光ダイオード
LSB	Least Significant Bit	最下位ビット
MCU	Micro-controller Unit	マイクロコントローラユニット
MTU	Multi-Function Timer Pulse Unit	マルチファンクションタイマパルスユニット
NAC (NACK)	Negative Acknowledgement	否定応答
RSK+	Renesas Starter Kit+	ルネサススタータキット

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。

# 目次

1. 概要 .....	7
1.1 目的 .....	7
1.2 特徴 .....	7
2. はじめに .....	8
3. システムブートローダ .....	9
3.1 メモリマップ-フラッシュ .....	9
3.2 メモリマップ-RAM .....	9
3.3 動作 .....	11
4. ベクタ配置 .....	12
4.1 割り込みベクタテーブル .....	12
4.2 例外ベクタテーブル .....	12
4.3 コード説明 .....	13
4.3.1 X モデム送信実行 .....	14
4.3.2 Flash API と FIT .....	14
4.4 ブートローダのリンクセクションアドレス .....	14
4.5 ブートローダアプリケーションコードについて .....	15
4.5.1 リンクセクションアドレス .....	15
4.5.2 レジスタ設定 .....	15
4.5.3 マイクロコントローラの初期設定値 .....	15
5. 追加情報 .....	16

## 1. 概要

### 1.1 目的

本アプリケーションノートは、RSK+RX65N プラットフォームにおいてシステムブートローダサンプルコードの動作について説明し、同様のアプリケーション開発を支援します。なお、本アプリケーションノートでは、「RX Family Flash Module Using Firmware Integration Technology(FIT)」を使用しています。

### 1.2 特徴

システムブートローダサンプルコードは、SCI 経由で RX65N の内蔵フラッシュメモリに S-レコード・ファイルまたは HEX ファイルフォーマットのアプリケーションコードを配置し、実行させることが可能です。また、CRC 演算とウォッチドッグタイマを使用してアプリケーションコードの正当性を評価します。

CPU ボードはマイクロコントローラの動作に必要な回路を全て備えています。

## 2. はじめに

本アプリケーションノートは、システムブートローダサンプルコードについて説明します。システムブートローダサンプルコードは、同等の機能を持つデバイスまたは PC とシリアル通信を行うことで、マイコンの内蔵フラッシュメモリにアプリケーションコードをダウンロードし、それを実行することができます。

## 3. システムブートローダ

### 3.1 メモリマップフラッシュ

システムブートローダとアプリケーションコードの使用するフラッシュメモリマップを図 3-1 を示します。

システムブートローダはフラッシュライブラリを含み、コードフラッシュ領域のブロック 7 (アドレス 0xFFFF0000 からアドレス 0xFFFF1FFF)内に配置されています。システムブートローダの例外ベクタは、アドレス 0xFFFFFFFF80 からアドレス 0xFFFFFFFFFB に配置されています。リセットベクタは、アドレス 0xFFFFFFFFFC に配置されています。

アプリケーションコードは、コードフラッシュ領域のブロック 1(アドレス 0xFFFFC000 からアドレス 0xFFFFDFFF)内に配置されています。アプリケーションコードの例外ベクタは、アドレス 0xFFFFFEF4 から 0xFFFFF6F に配置されています。リセットベクタは、アドレス 0xFFFFF70 に配置されています。

アプリケーションコードチェックサムバイトをアプリケーションコードチェックサム用アドレス 0xFFFFF7C に配置しています。アプリケーションコードチェックサムバイトは、システムブートローダがアプリケーションコード領域に有効なアプリケーションを含んでいるかを判定するためのデータです。

### 3.2 メモリマップ-RAM

CPU ボード上の RX65N マイクロコントローラは 256K バイトの RAM を内蔵しています。

アプリケーションコードとブートローダコードは、別々に動作して RAM を共有していないため、RAM 使用に制限はありません。

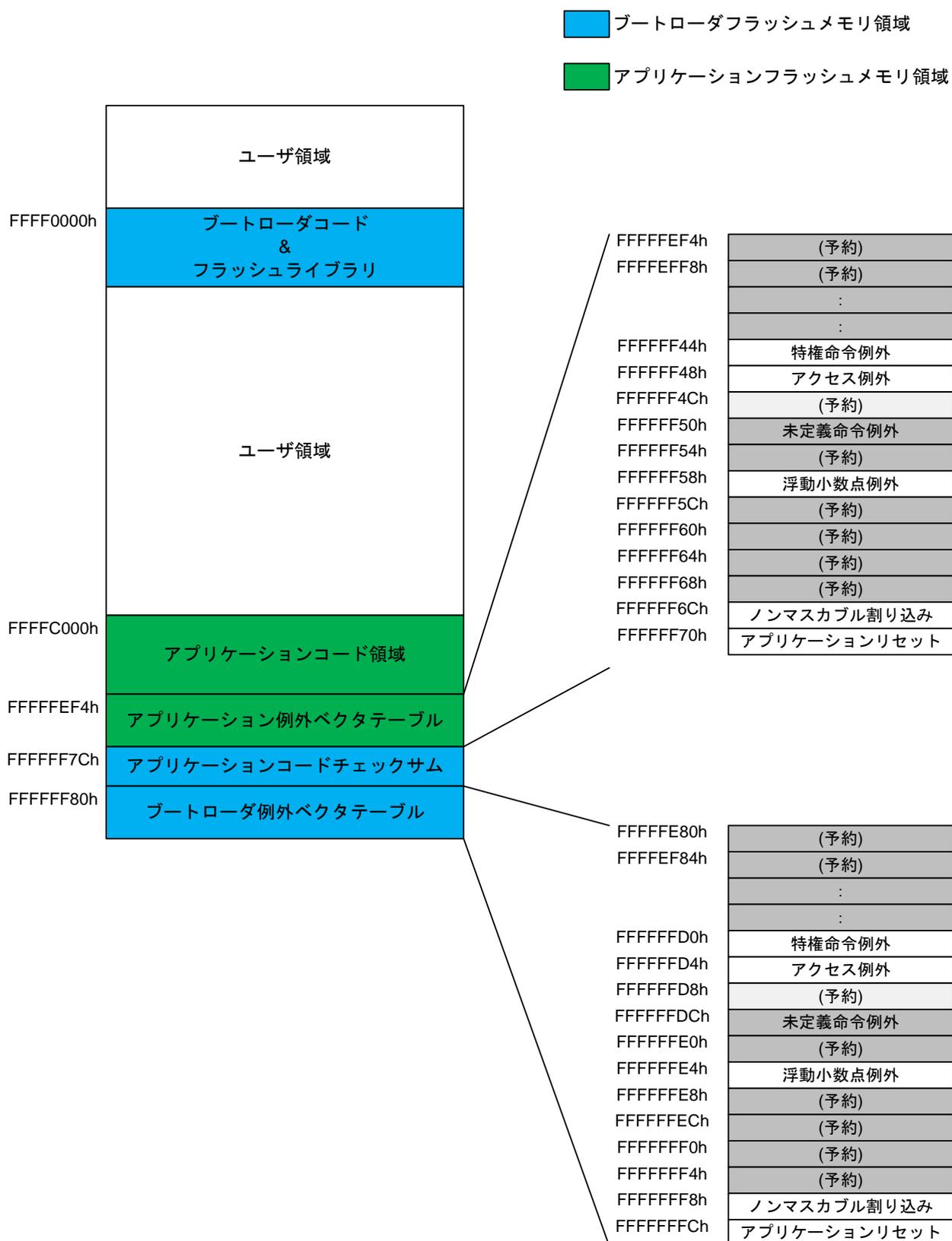


図 3-1: フラッシュメモリマップ

### 3.3 動作

図 3-2 にブートローダの動作フローを示します。

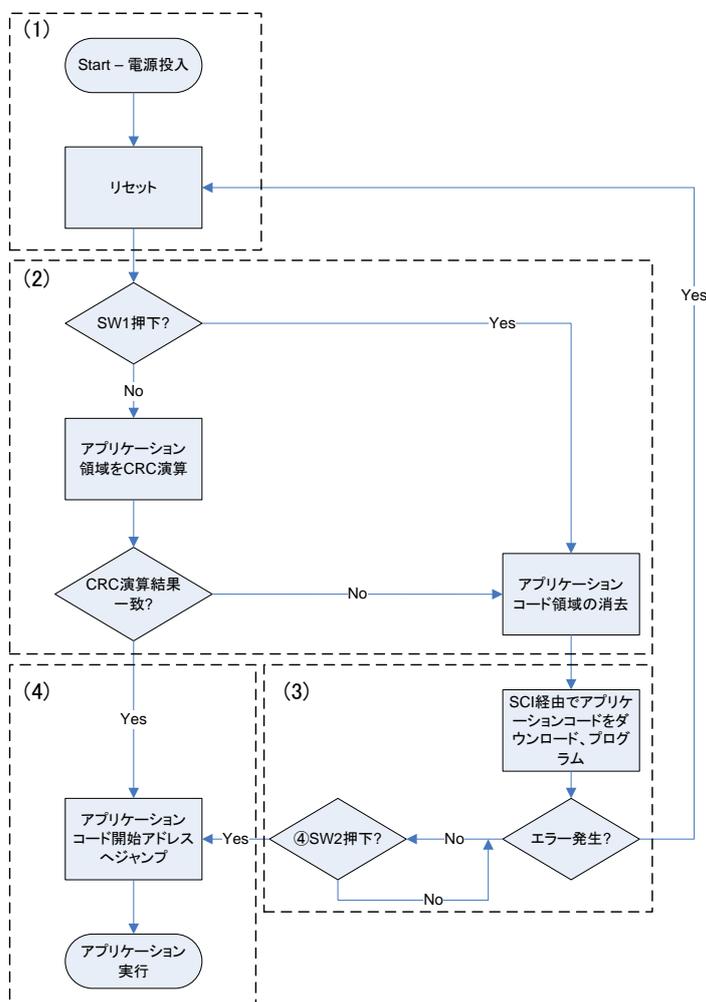


図 3-2: ブートローダ動作フロー

- (1) マイクロコントローラは、システムブートローダのリセットベクタから動作を開始します。
- (2) システムブートローダは動作を開始した際に、以下の各条件でアプリケーションコード領域をイレーズします。
  1. CPU ボード上の SW1 が押されている。
  2. CRC 演算結果がアプリケーションコードチェックサムバイトと一致しない。
- (3) RL78/G1C マイクロコントローラによる USB シリアル変換を介した SCI 接続によってアプリケーションコード領域に書き込みします。書き込みが完了すると、システムブートローダはアプリケーションコード領域に対して CRC 演算を実行し、アプリケーション領域チェックサム用アドレスに演算結果を格納します。エラーやハングアップが発生した場合は CPU ボード上のリセット SW を押して再度書き込みを実行してください。
- (4) システムブートローダは、以下の各条件でアプリケーションコードを実行します。
  1. ③の CRC 演算結果格納後、CPU ボード上の SW2 を押す。
  2. SW1 を開放した状態でのリセット後、CRC 演算結果がアプリケーションコードチェックサムバイトと一致した。

システムブートローダはアプリケーションコードの例外ベクタテーブルのリセットベクタを読み、アプリケーションのプログラム開始アドレスにジャンプします。

## 4. ベクタ配置

RX65N マイクロコントローラは、割り込みベクタと例外ベクタの2種類のベクタを持っています。

### 4.1 割り込みベクタテーブル

割り込みベクタテーブルは、テーブルの配置アドレスを変えることができるベクタテーブルです。主に割り込みの各ベクタを格納しています。

ユーザアプリケーションは、アプリケーションの割り込みベクタテーブルの位置に割り込みテーブルレジスタ (INTB) を設定する必要があります。

### 4.2 例外ベクタテーブル

例外ベクタテーブルは、予約ベクタと以下のベクタを含みます：

- リセット
- ノンマスカブル割り込み
- 浮動小数点例外
- 未定義命令例外
- アクセス例外
- 特権命令例

ブートローダの例外ベクタテーブルは 0xFFFFF80 から 0xFFFFFFFF を使用します。このアドレスは、ブートローダ領域に配置されており、ユーザアプリケーションによって書き換えられることはありません。リセット後は、例外テーブルレジスタ (EXTB) を 0xFFFFF80 に設定します。

ユーザアプリケーションの例外ベクタテーブルは、0xFFFFFEF4 から 0xFFFFFFFF73 を使用します。ユーザアプリケーションヘジャンプした後は、例外テーブルレジスタ (EXTB) を 0xFFFFFEF4 に再設定します。リセットを除く例外が発生した場合、アプリケーション側の例外処理が呼び出されます。

### 4.3 コード説明

システムブートローダサンプルの一部はコード生成を使用してコードを生成しています。コード生成は直感的な GUI を使用することで、様々なマイクロコントローラの周辺機能や動作に必要なパラメータを設定することができ、開発工数の大幅な削減が可能です。たとえば、SCI のボーレートを変更する場合などがあります。コード生成によって生成されるファイルは、ファイルの先頭に「r\_cg\_」が付加されます。生成されたファイルにカスタムコードを加える場合、以下に示すコメント文の間にカスタムコードを加えてください。

```
/* Start user code for adding. Do not edit comment generated here */  
/* End user code. Do not edit comment generated here */
```

コード生成の GUI 上で設定した内容を変更したい場合等、再度コード生成を行う場合にコード生成はこれらのコメント文を見つけて、コメント文の間に加えられたカスタムコードを保護し、コメント文の外側の内容を上書きします。

コード生成の使用法詳細については、コード生成支援ツールチュートリアルマニュアルを参照してください。

r\_cg\_main.c は、main 関数を含んでいるファイルです。main 関数は、**図 3-2** のフローチャートの処理を行います。

update\_from\_data\_source 関数は、SCI 経由のダウンロードデータの更新方法とプログラムするかどうかを決定します。更新方法の選択は、自動的に送信されてくるデータを適切に処理するためにフォーマット（S-レコード・ファイルまたは HEX ファイル）を判定します。データを完全に受信してからフラッシュへプログラムする必要があるか決定します。

SCI コードは r\_cg\_sci.c にあります。通信設定は、ボーレート 38400bps、8 データビット、1 ストップビット、パリティ機能なしに設定されています。これらの設定はコード生成によって変更可能です。

PC からのデータを制御するために、X モデム転送プロトコルが使用されます。これは、128 バイトパケットベースのプロトコルであり、システムブートローダはデコードおよびプログラム動作中の間、PC からの転送を遅滞させることが可能です。

X モデムパケットは buffer.c 中の制御によってバッファに送り込まれます。このバッファはバッファの最後尾へ到達すると先頭にループする循環的なバッファです。バッファ処理は異なる通信媒体からのデータ受信を容易に置き換えられるように構築されています。

code\_flash.c で、FIT によるプログラムまたはイレーズを実施します。write\_flash 関数は、指定のフラッシュアドレスにデータを書き込みます、また、書き込みサイズが書き込みブロックサイズを超えている場合は、ブロックサイズを変更します。flash\_flush\_buffer 関数は、書き込むブロックを更新し、フラッシュに最終データを書き込むことが可能です。

S-レコードフォーマットファイルのデコードは srec.c において処理されます。S-レコード・ファイル形式の詳細については以下のリンクを参照してください。 [http://en.wikipedia.org/wiki/SREC\\_\(file\\_format\)](http://en.wikipedia.org/wiki/SREC_(file_format))

HEX フォーマットファイルのデコードは hex.c において処理されます。HEX ファイル形式の詳細については以下のリンクを参照してください。 [http://en.wikipedia.org/wiki/Intel\\_HEX](http://en.wikipedia.org/wiki/Intel_HEX)

ブートローダは、システム保全のためにウォッチドッグタイマを使用しています。ウォッチドッグタイマは、r\_cg\_iwdt.c ファイルを通して処理されます。ブートローダ中の R\_IWDT\_Restart 関数は、タイムアウトおよびリセットを防ぐためにウォッチドッグタイマをリセットします。

### 4.3.1 X モデム送信実行

X モデムプロトコルの詳細については、以下のリンクを参照してください。

<http://en.wikipedia.org/wiki/XMODEM>

ブートローダは、標準的な X モデムプロトコルに準拠しています。たとえば、XModem-1K や XModem-CRC で通信することが可能です。

標準プロトコルに従って、第 1 パケットが受け取られる前に CPU ボードから 10 秒毎に NAK 信号が送信されます。これは、ターミナルの送信初期化および第 1 パケットの送信間に 10 秒以内の遅延が起きる可能性があるためです。

X モデム送信中にブートローダが問題を検出した場合、<CAN>フラグをターミナルへ送信されます。これは、PC からの送信を取り消し、PC に表示されるエラーメッセージを許可します。<CAN>フラグによる送信取り消しは、すべてのターミナルソフトウェアでサポートされているわけではありません。

### 4.3.2 Flash API と FIT

ルネサスの Flash API は、フラッシュの消去およびプログラムを実行するために使用されます。本ブートローダではメモリ使用量を少なくするため、ルネサスの Board Support Package (BSP) をカスタムして使用しています。カスタム方法については、「Using the Simple Flash API for RX without the r\_bsp Module」アプリケーションノートを参照してください。

## 4.4 ブートローダのリンクセクションアドレス

以下はブートローダのリンクセクションのための設定です。メモリマップ詳細については、セクション 3.1 を参照してください。

アドレス	セクション	備考
0x00001000	SU	ブートローダのワーク RAM 開始アドレス
	SI	
	B_1	
	R_1	
	B_2	
	R_2	
	B	
	R	
	RPFRAM	
0xFFFF0000	PRResetPRG	ブートローダコードの開始アドレス
	C_1	
	C_2	
	C	
	C\$*	
	D*	
	W*	
	L	
	PIntPRG	ブートローダの割り込みベクタテーブル
	P	
	PFRAM	
0xFFFFFFFF80	EXCEPTVECT	ブートローダの例外ベクタテーブル
0xFFFFFFFFFC	RESETVECT	リセット

## 4.5 ブートローダアプリケーションコードについて

### 4.5.1 リンクセクションアドレス

ブートローダアプリケーションサンプル (System\_Bootloader\_Application) のコードとベクタテーブルは、以下または、**図 3-1** の通りに設定する必要があります。

アドレス	セクション	備考
0x00004400	SU	アプリケーションのワーク RAM 開始アドレス
	SI	
	B_1	
	R_1	
	B_2	
	R_2	
	B	
	R	
0xFFFFC000	PResetPRG	アプリケーションコードの開始アドレス
	C_1	
	C_2	
	C	
	C\$*	
	D*	
	W*	
	L	
	PIntPRG	アプリケーションの割り込みベクタテーブル
	P	
0xFFFFFEF4	EXCEPTVECT	アプリケーションの例外ベクタテーブル
0xFFFFF70	RESETVECT	アプリケーションのリセット

**図 3-1** に示されているブートローダアプリケーションのリンカマップ出力ファイルを参照して正しく設定してください。本サンプルコードは参考例のため、配置を変更した場合、正しく動作しないことがあります。必要に応じて処理を追加してください。

### 4.5.2 レジスタ設定

RX65N マイクロコントローラは不揮発性の設定情報を格納するいくつかのレジスタを持っています。

- a) エンディアン選択レジスタ (MDE)
- b) オプション機能選択レジスタ (OFS0、OFS1)
- c) ID コード

これらのレジスタは、コード生成が出力する `r_cg_vecttbl.c` ファイルで設定されています。

### 4.5.3 マイクロコントローラの初期設定値

アプリケーションが動作する前にブートローダが動作し、クロック発生回路、SCI およびウォッチドッグタイマを含む様々な周辺が設定されます。そのため、アプリケーションの動作開始前に一部の周辺機能を停止しています。アプリケーション動作開始時に、使用する周辺機能を再設定します。

## 5. 追加情報

### サポート

統合開発環境の使用方法等の詳細情報は、統合開発環境のヘルプメニューを参照してください。

RX65N グループ マイクロコントローラに関する詳細情報は、RX65N グループ、RX651 グループ ユーザーズ マニュアルハードウェア編を参照してください。

アセンブリ言語に関する詳細情報は、RX ファミリユーザーズマニュアルソフトウェア編を参照してください。

オンラインの技術サポート、情報等は <https://www.renesas.com/rskrx65n> より入手可能です：

### オンライン技術サポート

技術関連の問合せは、<https://www.renesas.com/support/contact.html> を通じてお願いいたします。

ルネサスのマイクロコントローラに関する総合情報は、<https://www.renesas.com> より入手可能です：

### 商標

本書で使用する商標名または製品名は、各々の企業、組織の商標または登録商標です。

### 著作権

本書の内容の一部または全てを予告無しに変更することがあります。

本書の著作権はルネサス エレクトロニクス株式会社にあります。ルネサス エレクトロニクス株式会社の書面での承諾無しに、本書の一部または全てを複製することを禁じます。

© 2016 Renesas Electronics Europe Limited. All rights reserved.

© 2016 Renesas Electronics Corporation. All rights reserved.

© 2016 Renesas System Design Co., Ltd. All rights reserved.

改訂記録	RSK+RX65N 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	2016.6.30	－	初版発行

---

RSK+RX65N ブートローダ アプリケーションノート

発行年月日 2016年6月30日 Rev.1.00

発行 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)

---

---

RSK+RX65N ブートローダ アプリケーションノート

発行年月日 2016年6月30日 Rev.1.00

発行 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)

---

RX65N グループ