RL78 ファミリ

RL78 フラッシュプログラマ (RL78 プロトコル A 編)

要旨

本アプリケーションノートは、RL78 プロトコル A に対応した RL78 マイクロコントローラの内蔵フラッシュ・メモリにプログラムを書き込む方法について説明します。

動作確認デバイス

RL78/G23

本アプリケーションノートを他のマイクロコントローラへ適用する場合、そのマイクロコントローラの仕様に合わせて変更し、十分評価してください。

関連ドキュメント

本アプリケーションノートに関連するドキュメントを以下に示します。あわせて参照してください。

- RL78 マイクロコントローラ (RL78 プロトコル A) シリアルプログラミング編 (R01AN0815)
- RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル (R20UT4830)
- RL78 ファミリ Renesas Flash Driver RL78 Type01 SC 対応仕様 (Code Flash) (R20AN0653)
- RL78 ファミリ Renesas Flash Driver RL78 Type01 SC 対応仕様 (Data Flash) (R20AN0654)
- RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル (R20UT4870)

目次

1. 概要 3
1.1 オンライン・モード 4
1.2 オフライン・モード 4

2. 開発環境 5
2.1 Renesas Flash Driver RL78 Type01 5

3. 外部仕様 6
3.1 ホスト通信仕様 6
3.1.1 コマンド共通仕様 6
3.1.2 setup コマンド 7
3.1.3 lod コマンド 8
3.1.4 ep コマンド 9
3.2 オンライン・モードでの書き込み 11
3.3 オフライン・モードでの書き込み 11
3.3.1 書き込み実行 11
3.4 LED 表示仕様 11
3.5 エラーコード仕様 12
3.6 フローチャート 14
3.6.1 メインループ (main 関数) 14
3.6.2 オンライン・モード書き込みの流れ 15
3.6.3 オフライン・モード書き込みの流れ 16

4. ハードウェア説明 17
4.1 ハードウェア構成例 17
4.2 ターゲットインタフェース仕様 18
4.2.1 1 線 UART 18
4.2.2 2 線 UART 18
4.3 使用端子一覧 19
4.4 RL78/G23-128p Fast Prototyping Board の設定 19

5. ソフトウェア説明 20
5.2 オプション・バイト設定 24
5.3 オンチップ・デバッグ・セキュリティ ID 24
5.4 セクション設定 24
5.4.1 データ・フラッシュ領域 (0xF1000 - 0xF2FFF, 8KB) 26
5.5 スマート・コンフィグレータの設定 27
5.6 関数一覧 29
5.7 関数仕様 30

6. 参考ドキュメント 36

改訂記録 37

1. 概要

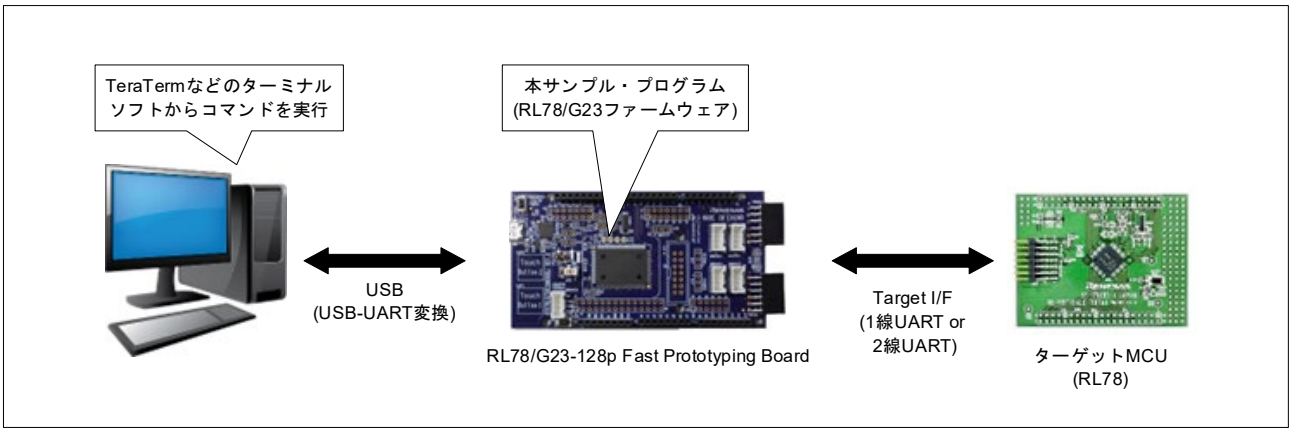
本サンプル・プログラムは、RL78 マイクロコントローラの内蔵フラッシュ・メモリにプログラムを書き込むファームウェアであり、以下の特徴があります。

- 書き込み対象のマイクロコントローラ (ターゲット MCU) は、「RL78 プロトコル A」に対応した RL78 マイクロコントローラです。
- RL78 プロトコル A のシリアルプログラミングにより書き込みを行います。
- プログラムファイル (書き込み用データ) はモトローラ S フォーマットに対応します。
- ホスト PC を使用する「オンライン・モード」と、ホスト PC を使用しない「オフライン・モード」に対応します。
- RL78/G23 を搭載した評価ボード (RL78/G23-128p Fast Prototyping Board) と組み合わせてフラッシュプログラマを実現します。

1.1 オンライン・モード

オンライン・モードは、ホスト PC と評価ボード (RL78/G23-128p Fast Prototyping Board) を USB で接続して、ホスト PC からコマンドを送信することでターゲット MCU に書き込みを行います。

図 1-1 オンライン・モード



1.2 オフライン・モード

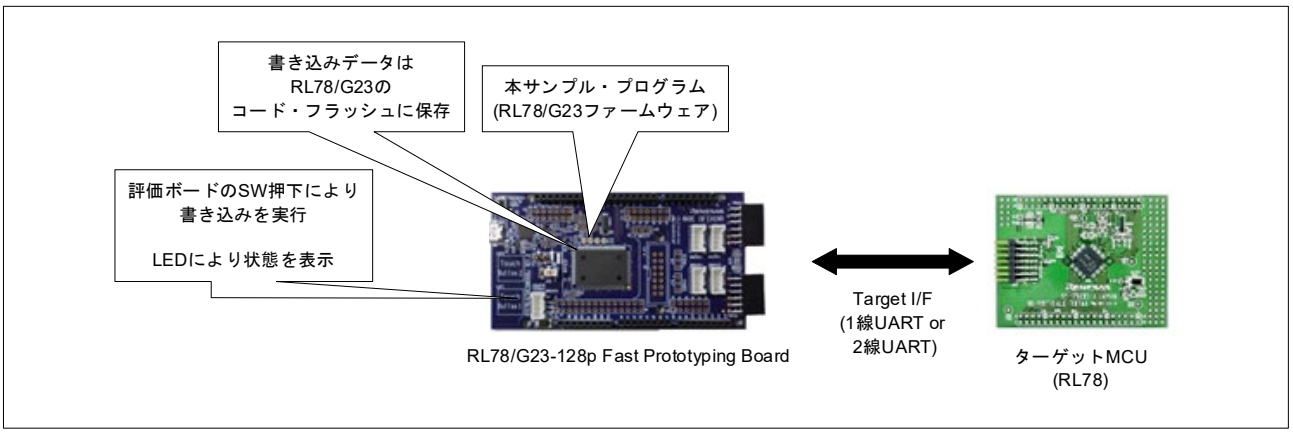
オフライン・モードは、評価ボード (RL78/G23-128p Fast Prototyping Board) 上のプッシュスイッチ (SW1) を押下することで、ターゲット MCU に書き込みを行います。

事前にホスト PC と評価ボードを USB で接続して、コマンドにより設定とプログラムファイルのダウンロードを行う必要があります。

プログラムファイルのデータは、RL78/G23 のコード・フラッシュ上の書き込み用データセクションに保存されます。

オフライン・モードでは対応するコード・フラッシュのメモリサイズは 512KB までとなります。512KB を超えるプログラムを使用する場合は、オンライン・モードを使用してください。

図 1-2 オフライン・モード



2. 開発環境

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条件で動作を確認しています。

表 2-1 動作確認条件

開発ツール		説明
MCU		型番：R7F100GSN2DFB (内蔵メモリ:コード・フラッシュ 768KB, 内蔵 RAM 48KB, データ・フラッシュ 8KB)
評価ボード (簡易プログラマ)		RL78/G23-128p Fast Prototyping Board 型番：RTK7RLG230CSN000BJ 注意:本サンプル・プログラムは、HS モード (高速オンチップ・オシレータ 32MHz) で動作するため、VDD を 1.8V 以上に設定してください。
エミュレータ		本サンプル・プログラムをデバッグする場合は、E2 エミュレータ Lite または E2 エミュレータが必要になります。 RL78/G23-128p Fast Prototyping Board の USB は、ホスト PC と RL78/G23 の通信用に使用されます。そのため、USB を使用したデバッグ (RL78 COM port デバッグ) は使用できません。
CS+	統合開発環境	ルネサス エレクトロニクス製 CS+ for CC V8.09.00
	C コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.12.00
e2studio	統合開発環境	ルネサス エレクトロニクス製 e2studio V2023-01
	C コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.12.00
IAR	統合開発環境	IAR Systems 製 IAR Embedded Workbench for Renesas RL78 V4.21
	C コンパイラ	IAR Systems 製 IAR C/C++ Compiler for Renesas RL78 V 4.21.1
セルフ・プログラミング用ライブラリ		Renesas Flash Driver RL78 Type01

2.1 Renesas Flash Driver RL78 Type01

Renesas Flash Driver RL78 Type01 は、RL78/G2x のフラッシュ・メモリ内のデータを書き換えるためのソフトウェアです。詳細は「Renesas Flash Driver RL78 Type01 ユーザーズマニュアル (R20UT4830)」を参照してください。

Renesas Flash Driver RL78 Type01 は、下記 URL から取得することができます。

https://www.renesas.com/jp/ja/document/scd/renesas-flash-driver-rl78-type-01-rl78g23

備考

RFD には、スマート・コンフィグレータ (SC) 対応仕様の RFD と従来の RFD (上記 URL) の提供形態があります。本アプリケーションノートでは、SC 対応仕様の RFD を使用しています。

3. 外部仕様

3.1 ホスト通信仕様

ホスト PC とプログラマボード間の通信仕様について説明します。

プログラムボード (RL78/G23-128p Fast Prototyping Board) の USB コネクタ (USB-UART 変換) とホスト PC を接続します。

本サンプル・プログラムではホスト PC とプログラマボード間の通信設定は以下としています。

表 3-1 ホスト通信設定

項目	設定
データ・ビット長 (bit)	8
データ転送方向	LSB ファースト
パリティ設定	なし
転送レート (bps]	115,200
ストップビット (bit)	1
フロー制御	ソフトウェア (Xon/Xoff)

3.1.1 コマンド共通仕様

以下の仕様に従い、コマンド通信を行います。

- プログラマボードは、受信準備が完了すると、プロンプト文字'> 'をホスト PC に送信します。
- プログラマボードは、コマンドの入力時のエコーバックを送信します。
- コマンドを実行する際には、ホスト PC から改行 (CRLF:0x0D,0x0A) を送信してください。
- プログラマボードは、コマンドの実行が成功した場合は、"PASS"を送信します。
- プログラマボードは、コマンドの実行が失敗した場合は、"ERROR:XX"を送信します。
  XX は 2 桁の 16 進数で表示されます。詳しくは表 3-7 と表 3-8 を参照してください。

本サンプル・プログラムで対応するホスト PC のコマンド一覧は、以下となります。

表 3-2 ホスト PC のコマンド一覧

コマンド	説明
setup	ターゲットMCUとの通信方式、通信速度、VDD電圧値、セキュリティIDコード、オフライン・モード時のベリファイを設定します。
lod	オフライン・モードで使用するプログラムファイル (モトローラSフォーマット) を、プログラマボードに保存します。
ep	オンライン・モードで消去と書き込みを実行します。 オプション設定で、ベリファイとチェックサムの実行を追加できます。

3.1.2 setup コマンド

プログラマボードの動作設定を行います。

ターゲット MCU との通信方式、通信速度、VDD 電圧値、オフライン・モード時のベリファイ実行を設定できます。

ターゲット MCU にプログラムを書き込む前に setup コマンドの実行が必要です。本コマンドで設定した内容は、プログラマボード上の RL78/G23 のデータ・フラッシュに保存し記憶されます。

表 3-3 に setup コマンドのオプションの詳細を示し、図 3-1 に setup コマンドの使用例を示します。

表 3-3 setup コマンドのオプション

オプション	設定	説明
-if	uart1	uart1： 1 線 UART (TOOL0) でターゲット MCU と通信を行います。 uart2： 2 線 UART (TOOL0, TOOLTxD, TOOLRxD) でターゲット MCU と通信を行います。 オプション省略： uart1 が選択されます。
	uart2	
-speed	115200	RL78 プロトコル A の Baud Rate Set コマンドで設定する通信速度(bps)を指定します。 オプション省略： 115200 が選択されます。
	250000	
	500000	
	1000000	
-vdd	x.x (10 進数, 整数 1 桁, 少数第 1 位)	RL78 プロトコル A の Baud Rate Set コマンドで設定する VDD 印可電圧値 (V) を指定します。 プログラマボードとターゲット MCU に供給している VDD の電圧値を設定してください。 オプション省略：3.3 が選択されます。
-v		オフライン・モードでベリファイを追加で実行します。 オンライン・モードでは、このオプションは無視されます。オンライン・モードでベリファイを追加する場合は、ep コマンドのオプションを使用してください。

図 3-1 ターミナルソフト画面：setup コマンドの使用例

```
> setup -if uart1 -speed 115200 -vdd 3.3 -v
PASS
>
```

3.1.3 lod コマンド

オフライン・モード用にプログラムファイルのデータをプログラマボードに搭載されている RL78/G23 のコード・フラッシュ上の書き込み用データセクションに保存します。セクションについては、「5.4 セクション設定」を参照ください。

オフライン・モードでターゲット MCU にプログラムを書き込む前に、lod コマンドの実行が必要です。

lod コマンド送信後、プログラマボードから「Please send a motorola-s file.」を受信した後にモトローラ S フォーマットのファイルを送信してください。モトローラ S フォーマットのファイルは、データレコードをアドレス昇順にしてください。

TeraTerm などのターミナルソフトを使用して、ファイルを送信する場合は、バイナリモードで送信してください。

lod コマンドで保存可能なプログラムファイルのコード・フラッシュとデータ・フラッシュのデータ範囲は以下となります。

- コード・フラッシュ：最大 512KB (0x00000 - 0x7FFFF)
- データ・フラッシュ：最大 8KB (0xF1000 - 0xF2FFF)

上記範囲を超える書き込みを行う場合は、オンライン・モード(ep コマンド)を使用してください。

図 3-2 に lod コマンドの使用例を示します。

図 3-2 ターミナルソフト画面：lod コマンドの使用例

```
> lod
Please send a motorola-s file.
<モトローラ S フォーマットファイル送信>
PASS
>
```

3.1.4 ep コマンド

オンライン・モードで書き込みを行うコマンドで、ホスト PC からモトローラ S フォーマットのファイルを受信しながら、ターゲット MCU に対して消去または書き込みを実行します。

オプション設定でベリファイとチェックサムを追加できます。全消去、書き込み/ベリファイ、チェックサムの順番で実行します。書き込みとベリファイは交互に実行されます。

フラッシュ・メモリの消去範囲は、コード・フラッシュとデータ・フラッシュの全領域が対象となります。

フラッシュ・メモリの書き込みとベリファイの範囲は、送信したモトローラ S フォーマットのデータが存在する領域が対象となります。

書き込みやベリファイは、内蔵フラッシュのブロック単位で実行されます。データがブロック単位にならない場合は、不足しているデータを 0xFF で充填します。

フラッシュ・メモリのチェックサムはコード・フラッシュとデータ・フラッシュの全領域が対象です。

ep コマンド送信後、プログラマボードから「Please send a motorola-s file.」,「Connect」を受信した後にモトローラ S フォーマットのファイルを送信してください。TeraTerm などのターミナルソフトを使用して、ファイルを送信する場合は、バイナリモードで送信してください。

プログラマボードに搭載された RL78/G23 は、ホスト PC からのファイル受信をしながら、ターゲット MCU との通信設定処理を実行します。ターゲット MCU との通信設定が完了すると、ホスト PC に”PASS”を送信します。その後、ターゲット MCU のフラッシュ・メモリに対して消去を実行します。消去が完了すると、ホスト PC に”PASS”を送信します。続いて、ターゲット MCU のフラッシュ・メモリに対して書き込みを実行します。書き込みが完了すると、ホスト PC に”PASS”を送信します。

ベリファイのオプションが指定されている場合は、書き込みとベリファイが交互に実行されます。書き込みとベリファイが完了すると、プログラマボードに搭載された RL78/G23 はホスト PC に”PASS”を送信します。

チェックサムのオプションが指定されている場合は、書き込み (ベリファイを含む) 後にチェックサムが実行されます。ターゲット MCU から受信したコード・フラッシュとデータ・フラッシュのチェックサム値をプログラマボードに搭載された RL78/G23 はホスト PC へ送信します。

表 3-4 に ep コマンドのオプションの詳細を示し、図 3-3 に ep コマンドの使用例を示します。

表 3-4 ep コマンドのオプション

オプション	説明
-v	本オプションを指定すると、オンライン・モードでベリファイを追加で実行します。
-s	本オプションを指定すると､オンライン･モードでチェックサムを追加で実行します。

図 3-3 ターミナルソフト画面：ep コマンドの使用例

```
> ep
Please send a motorola-s file.

Connect
<モトローラ S フォーマット ファイル送信>
PASS
Erase
PASS
Program
PASS
> ep -v -s
Please send a motorola-s file.

Connect
<モトローラ S フォーマット ファイル送信>
PASS
Erase
PASS
Program, Verify
PASS
Checksum
Code Flash: 0x1234
Data Flash: 0xABCD
PASS
>
```

3.2 オンライン・モードでの書き込み

ep コマンドの実行により書き込みを行います。

書き込みまでの流れは以下の通りです。

① setup コマンドの実行

② ep コマンドの実行

3.3 オフライン・モードでの書き込み

プログラマボード (RL78/G23-128p Fast Prototyping Board) のプッシュスイッチ (SW1) を使用して、PC レスで書き込みを行います。

書き込みまでの流れは以下の通りです。

① setup コマンドの実行

② lod コマンドの実行

③ SW1 の押下

注意. ①と②は、PC とプログラマボードを接続して行います。

3.3.1 書き込み実行

SW1 押下により消去、書き込み、ベリファイを実行します。ベリファイ実行は、事前に setup コマンドで-v オプションの選択が必要です。

表 3-5 オフライン・モードのフラッシュ操作範囲

フラッシュ操作	操作範囲
消去	コード・フラッシュとデータ・フラッシュの全領域
書き込み	コード・フラッシュとデータ・フラッシュの全領域 モトローラ S フォーマットのプログラムファイルデータにおいてデータが存在しない領域は 0xFF として処理を行います。
ベリファイ	コード・フラッシュとデータ・フラッシュの全領域 モトローラ S フォーマットのプログラムファイルデータにおいてでデータが存在しない領域は 0xFF として処理を行います。

3.4 LED 表示仕様

以下の表 3-6 に LED 表示とプログラマボードの動作状態の関係を示します。

表 3-6 LED 表示と動作状態

LED 表示		動作状態
LED1 (UART1 線/2 線インジケータ)	消灯	UART 動作停止中
	点灯	UART 1 線モードで動作中
	点滅	UART 2 線モードで動作中
LED2 (動作インジケータ)	消灯	書き換え動作停止中
	点灯	書き換え動作中
	点滅	エラー発生

3.5 エラーコード仕様

プログラマボードはコマンドの実行が失敗した際に、ホスト PC へエラーメッセージ (ERROR:XX) を送信します。XX は 2 桁の 16 進数エラーコードです。

表 3-7 と表 3-8 にエラーコードを示します。

表 3-7 エラーコードの説明(1/2)

エラーコード (16 進数)	説明
04	コマンド番号エラー ターゲット MCU から RL78 プロトコル A のステータス・コードのコマンド番号エラーを受信した場合のエラーです。
05	パラメータエラー ターゲット MCU から RL78 プロトコル A のステータス・コードのパラメータエラーを受信した場合のエラーです。
07	チェックサムエラー ターゲット MCU から RL78 プロトコル A のステータス・コードのチェックサムエラーを受信した場合のエラーです。
0F	ベリファイエラー ターゲット MCU から RL78 プロトコル A のステータス・コードのベリファイエラーを受信した場合のエラーです。
10	プロテクトエラー ターゲット MCU から RL78 プロトコル A のステータス・コードのプロテクトエラーを受信した場合のエラーです。
15	NACK ターゲット MCU から RL78 プロトコル A のステータス・コードの NACK を受信した場合のエラーです。
1A	消去エラー ターゲット MCU から RL78 プロトコル A のステータス・コードの消去エラーを受信した場合のエラーです。
1B	ブランクエラー ターゲット MCU から RL78 プロトコル A のステータス・コードのブランクエラーを受信した場合のエラーです。
1C	書き込みエラー ターゲット MCU から RL78 プロトコル A のステータス・コードの書き込みエラーを受信した場合のエラーです。

表 3-8 エラーコードの説明(2/2)

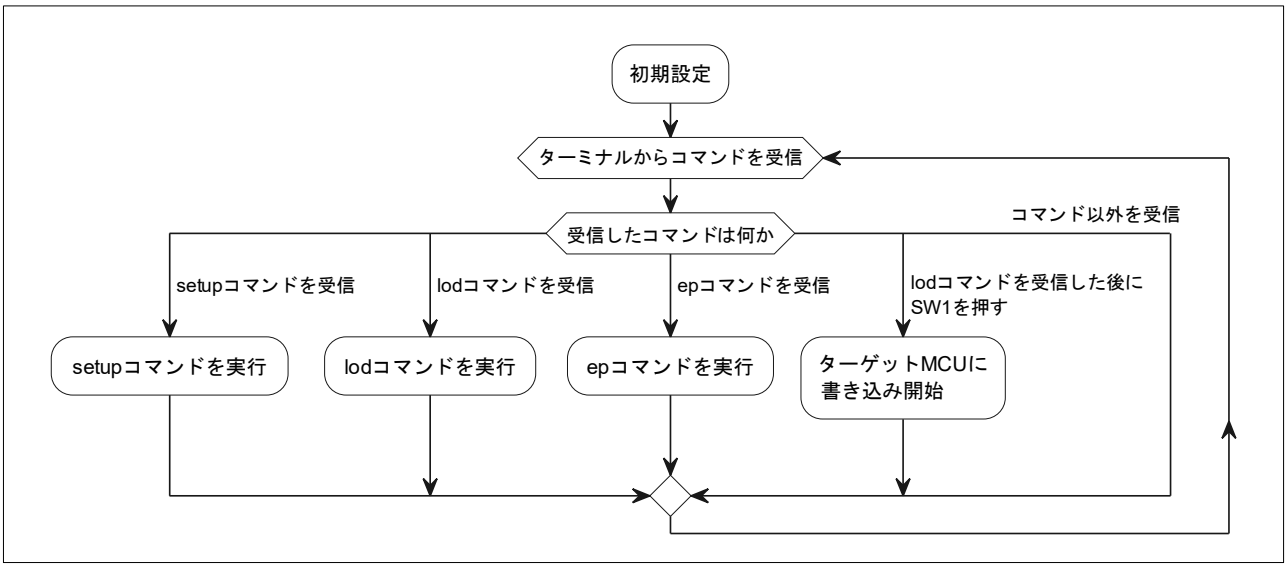
エラーコード (16 進数)	説明
F9	コマンドやパラメータのデータが不正 ホスト PC から送信したコマンドおよびオプションが不正の場合に発生します。
FA	バッファーオーバーラン発生 プログラマボードがホスト PC からのデータを受信時にバッファーオーバーランが発生した場合のエラーです。 ホスト PC との通信でフロー制御がソフトウェア(Xon/Xoff)に設定されているかを確認してください。
FB	モトローラ S フォーマットのデータが不正 lod コマンドや ep コマンドで送信するモトローラ S フォーマットが不正の場合に発生します。 モトローラ S フォーマットのデータレコードがアドレス昇順になっていない場合も本エラーが発生します。
FC	ターゲット MCU 通信タイムアウト発生 プログラマボードとターゲット MCU との通信でタイムアウトが発生した場合に発生します。
FD	モトローラ S フォーマットのファイルのアドレスが非対応 lod コマンドで受信したモトローラ S フォーマットのファイルに含まれるアドレスが範囲外の場合に発生します。
FE	コマンド通信データエラー ターゲット MCU から受信したパケットフォーマットが不正の場合に発生します。
FF	システムエラー 正常にプログラムが動作しなかった場合に発生します。

3.6 フローチャート

3.6.1 メインループ (main 関数)

図 3-4 にメインループの動作を示します。

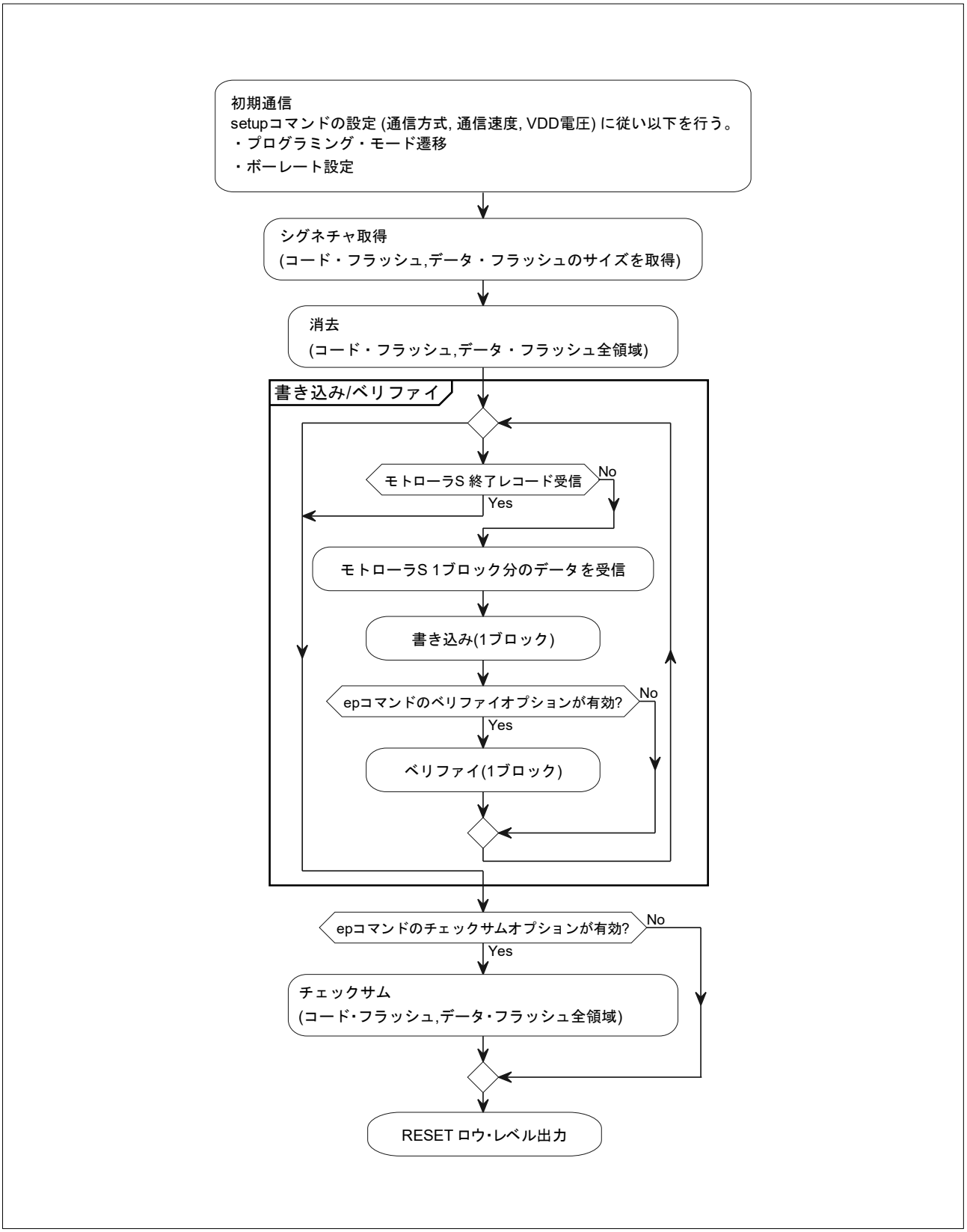
図 3-4 メインループ



3.6.2 オンライン・モード書き込みの流れ

図 3-5 に ep コマンド実行時の書き込み処理の流れを示します。

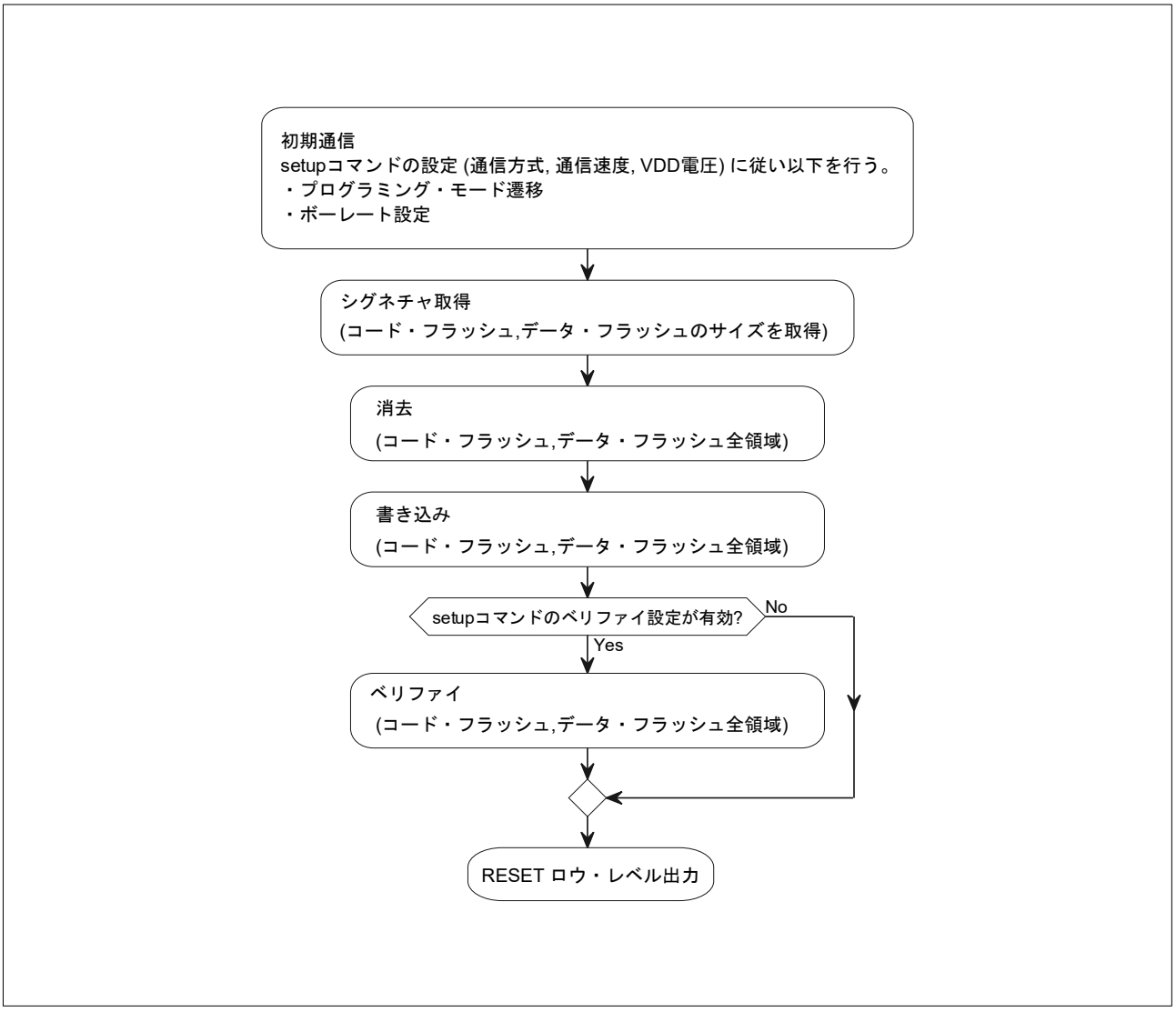
図 3-5 オンライン・モード



3.6.3 オフライン・モード書き込みの流れ

図 3-6 に SW1 押下時に実行される書き込み処理の流れを示します。

図 3-6 オフライン・モード

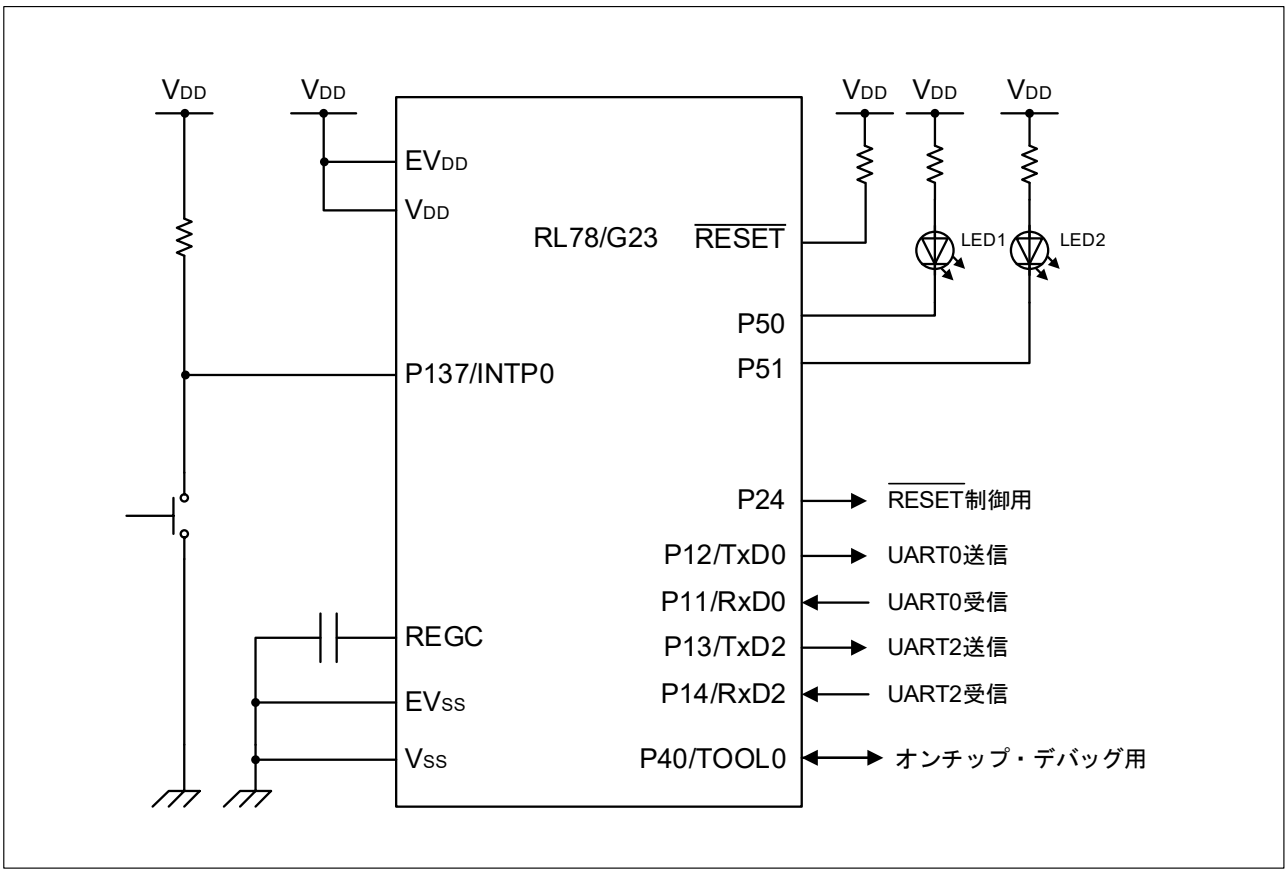


4. ハードウェア説明

4.1 ハードウェア構成例

図 4-1 に本アプリケーションノートで使用するハードウェア構成例を示します。

図 4-1 ハードウェア構成



注意 1. この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください (入力専用ポートは個別に抵抗を介して V_{DD} 又は V_{SS} に接続して下さい) 。

注意 2. EV_{SS} で始まる名前の端子がある場合には V_{SS} に、EV_{DD} で始まる名前の端子がある場合には V_{DD} にそれぞれ接続してください。

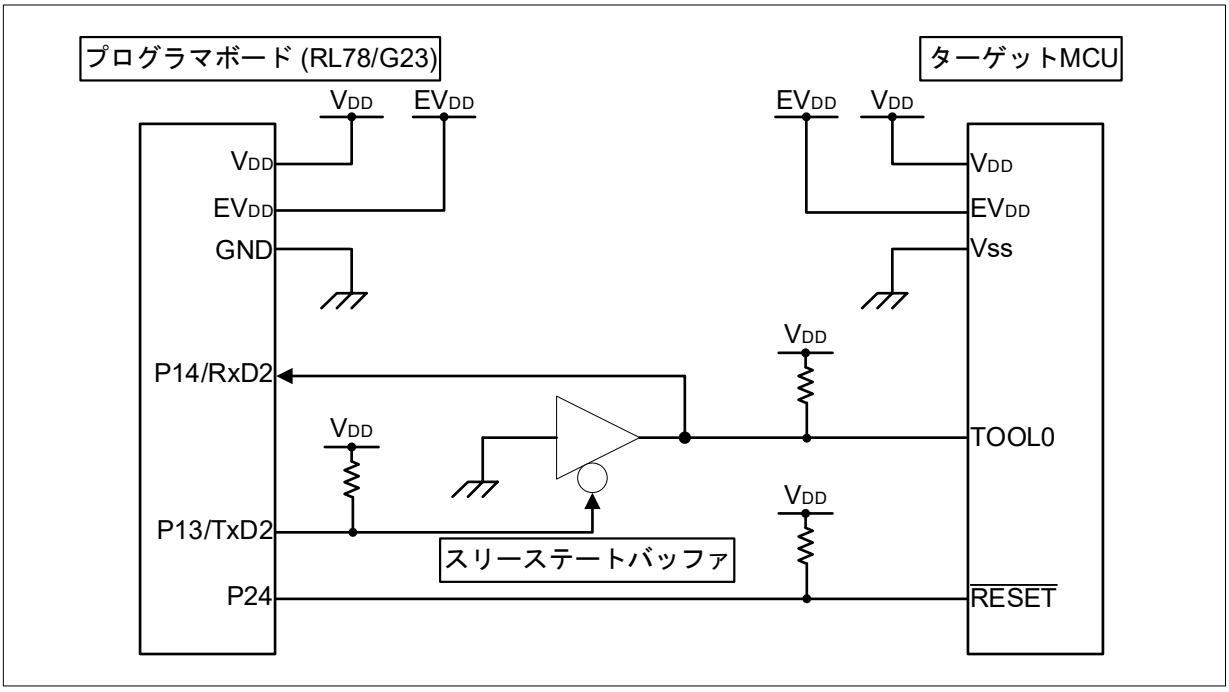
注意 3. V_{DD} は LVD0 にて設定したリセット解除電圧 (V_{LVD0}) 以上にしてください。

4.2 ターゲットインタフェース仕様

プログラマボード (RL78/G23) とターゲット MCU の接続方法を以下に示します。

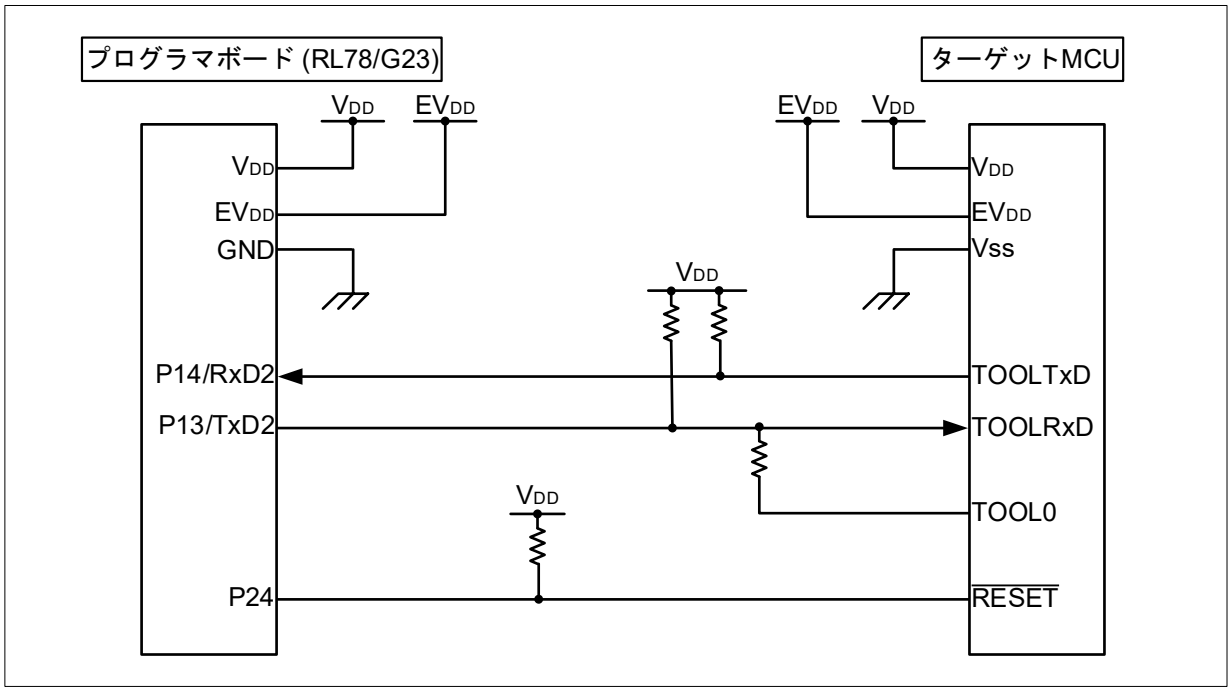
4.2.1 1 線 UART

図 4-2 1 線 UART



4.2.2 2 線 UART

図 4-3 2 線 UART



4.3 使用端子一覧

表 4-1 にサンプル・プログラムで使用する端子と機能を示します。

表 4-1 使用端子一覧

端子名	入出力	機能
P12/TxD0	出力	ホスト PC 通信用送信端子 (UART0)
P11/RxD0	入力	ホスト PC 通信用受信端子 (UART0)
P13/TxD2	出力	ターゲットインタフェース通信用送信端子 (UART2)
P14/RxD2	入力	ターゲットインタフェース通信用受信端子 (UART2)
P50, P51	出力	LED1,LED2 への出力端子
P24	出力	ターゲット MCU の $\overline{\text{RESET}}$ 制御用端子
P137/INTP0	入力	SW1 割込み入力端子

注意 本アプリケーションノートは、使用端子のみを端子処理しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください。

4.4 RL78/G23-128p Fast Prototyping Board の設定

E2 エミュレータ Lite や E2 エミュレータを使用してデバッグを行う場合は RL78/G23-128p Fast Prototyping Board で以下の設定が必要です。

- 「TOOL0_USB」、「RESET」、「T_RESET」のカットパターンをカットしてください。
- 「J15」、「J16」、「J19」のピンヘッダを 2-3 側にショートしてください。

VDD や EVDD の動作電圧の設定は「J20」で行います。

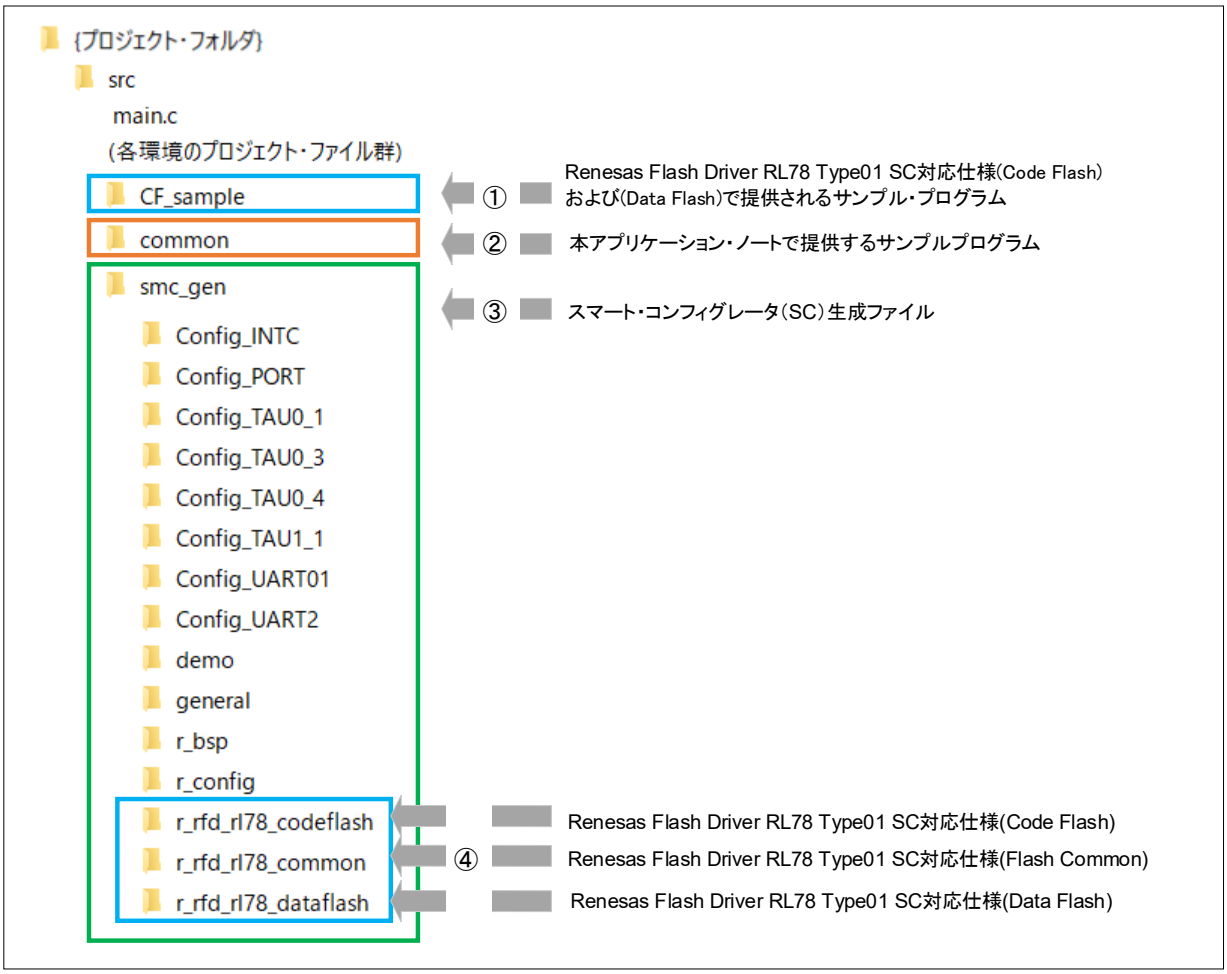
詳細は、「RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル (R20UT4870)」を参照ください。

5. ソフトウェア説明

5.1 フォルダ構成

サンプル・プログラムのフォルダ構成を図 5-1 に示します。

図 5-1 サンプル・プログラムのフォルダ構成



備考

サンプル・プログラムではスマート・コンフィグレータ（SC）が生成するソース・ファイルを使用しています。スマート・コンフィグレータでコードを再生成する場合には、④の Renesas Flash Driver RL78 Type01（SC 対応仕様）をルネサスエレクトロニクスホームページから入手し、スマート・コンフィグレータが参照するフォルダへ置いてください。Renesas Flash Driver RL78 Type01（SC 対応仕様）を置かずにコードを再生成すると、④のフォルダ以下のファイルが削除されます。なお、コードを再生成しない場合にはこの処置は必要ありません。

[手順]

1. ルネサスエレクトロニクスホームページから下記をダウンロードし、任意のフォルダでダウンロードした.zip ファイルを展開します。
   - RL78 Family Renesas Flash Driver RL78 Type 01 SC version (Flash Common) - Sample Code
     https://www.renesas.com/us/en/document/scd/rl78-family-renesas-flash-driver-rl78-type-01-sc-version-flash-common-sample-code?language=en&r=488891
   - RL78 Family Renesas Flash Driver RL78 Type 01 SC version (Code Flash) - Sample Code
     https://www.renesas.com/jp/ja/document/scd/rl78-family-renesas-flash-driver-rl78-type-01-sc-version-code-flash-sample-code?r=488891
   - RL78 Family Renesas Flash Driver RL78 Type 01 SC version (Data Flash) - Sample Code
     https://www.renesas.com/us/en/document/scd/rl78-family-renesas-flash-driver-rl78-type-01-sc-version-data-flash-sample-code?language=en&r=488891

2. スマート・コンフィグレータを起動し、「コンポーネント」タブをクリックします。

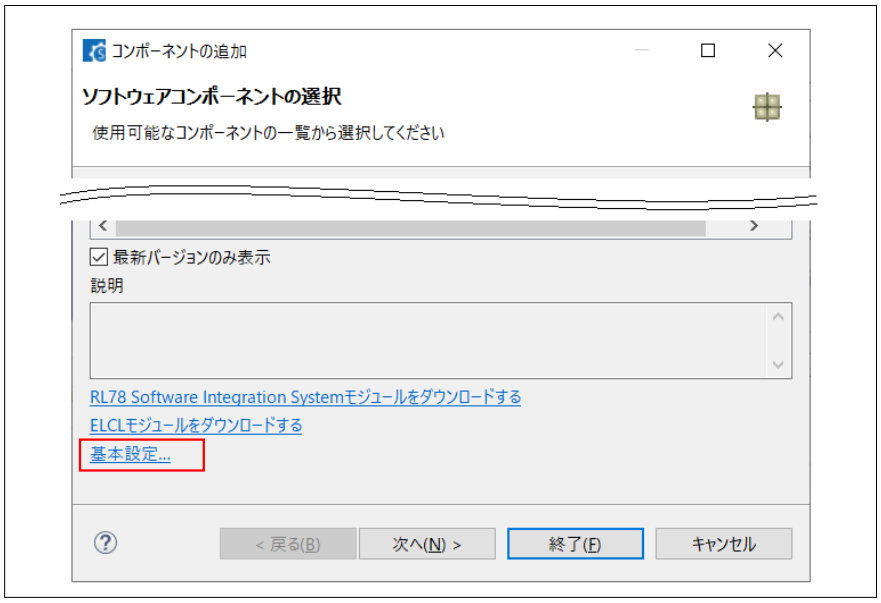
図 5-2 スマート・コンフィグレータ 「コンポーネント」タブの選択

3. 「コンポーネントの追加」ボタンを押し、「コンポーネントの追加」ダイアログを開きます。

図 5-3 スマート・コンフィグレータ 「コンポーネントの追加」ボタン

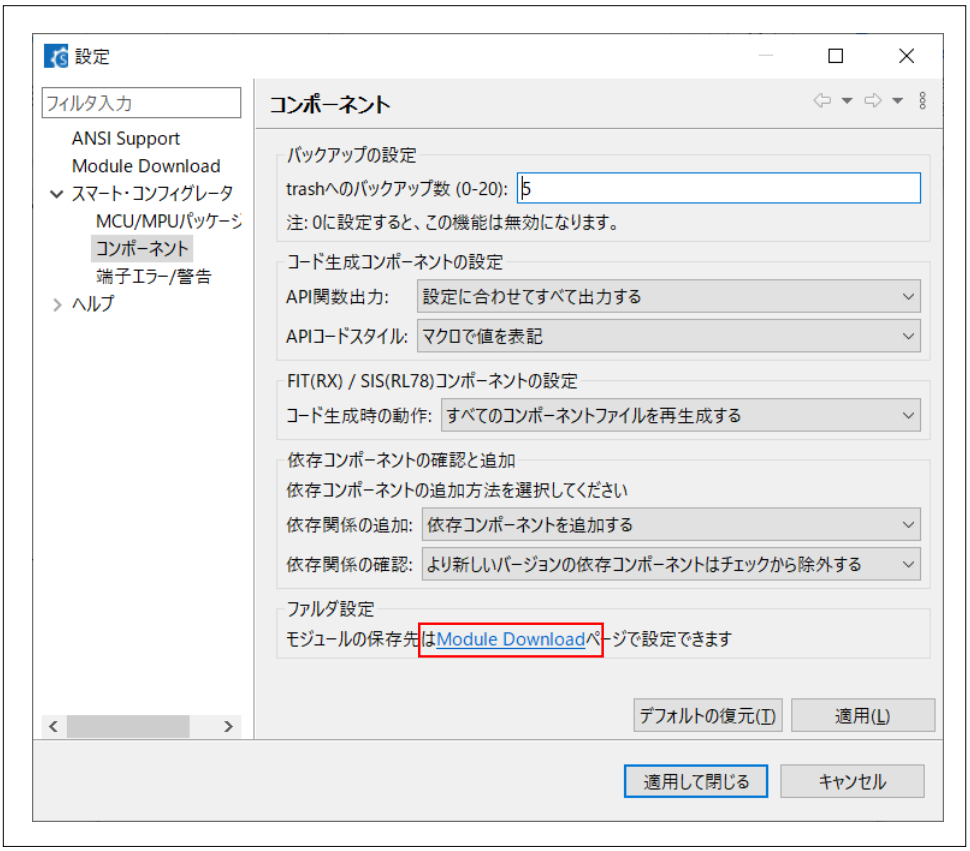
4. 「コンポーネントの追加」ダイアログで、「基本設定」をクリックします。

図 5-4 スマート・コンフィグレータ 「基本設定」



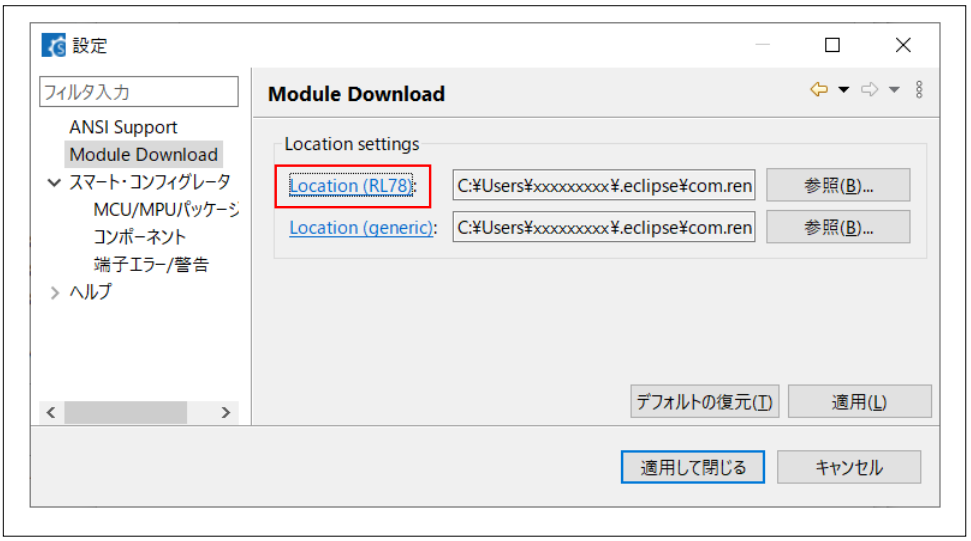
5. 「設定」ダイアログで、「Module Download」をクリックします。

図 5-5 スマート・コンフィグレータ 「設定」ダイアログ（1/2）



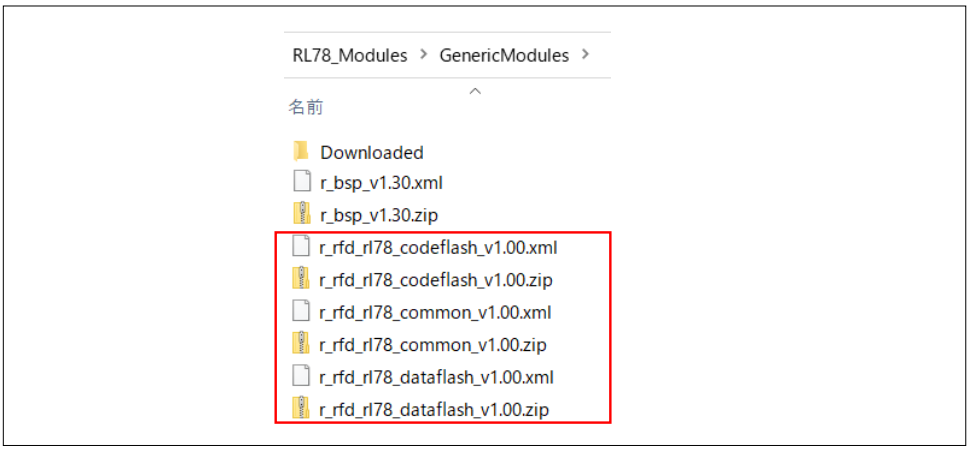
6. 「設定」ダイアログで、「Location(RL78)」をクリックします。

図 5-6 スマート・コンフィグレータ 「設定」ダイアログ（2/2）



7. Windows エクスプローラで表示されたフォルダに、手順 1 で展開した各フォルダから.xml ファイルと.zip ファイルをコピーします。

図 5-7 Renesas Flash Driver RL78 Type01（SC 対応仕様）の配置



8. 「設定」ダイアログで「適用して閉じる」をクリックします。次に、「コンポーネントの追加」ダイアログで「終了」をクリックします。

9. 手順 7 で追加した Renesas Flash Driver RL78 Type01 を認識するために、コード生成前にスマート・コンフィグレータを再起動します。

5.2 オプション・バイト設定

表 5-1 にサンプル・プログラムで使用する、オプション・バイトの設定を示します。

表 5-1 オプション・バイト設定

アドレス	設定値	内容
000C0H/040C0H	11101111B	ウォッチドッグ・タイマ動作停止
000C1H/040C1H	11111110B	LVD 検出電圧 : リセット・モード 立ち上がり時 TYP. 1.90 V (1.84 V ～ 1.95 V) 立ち下がり時 TYP. 1.86 V (1.80 V ～ 1.91 V)
000C2H/040C2H	11101000B	HS モード、 高速オンチップ・オシレータ・クロック : 32MHz
000C3H/040C3H	10000101B	オンチップ・デバッグ許可

RL78/G23 のオプション・バイトは、ユーザ・オプション・バイト (000C0H – 000C2H) とオンチップ・デバッグ・オプション・バイト (000C3H) で構成されています。

5.3 オンチップ・デバッグ・セキュリティ ID

RL78/G23 は、第三者からメモリの内容を読み取られないようにするために、フラッシュ・メモリの 000C4H–000CDH にオンチップ・デバッグ・セキュリティ ID 設定領域を用意しています。

5.4 セクション設定

表 5-2 に CC-RL のセクション設定を示します。

- 「.」から始まるセクションはコンパイラの予約セクションです。
- 「RFD_」「SMP_」から始まるセクションは Renesas Flash Driver RL78 Type01(RFD)とそのサンプル・プログラムの仕様で規定されているセクションです。
- 「FP_DATA」「FP_CODE」はターゲット MCU への書き込み用データを保存するために用意したセクションです。
- IAR は予約セクション名が異なりますが、基本的に同様のセクション構成になっています。

表 5-2 セクション一覧

セクション名	アドレス	説明
.vect	0x00000	割り込みベクタテーブル
.option_byte	0x000C0	ユーザ・オプション・バイト，およびオンチップ・デバッグ指定専用セクション
.security_id	0x000C4	セキュリティ ID 指定専用セクション
.const	0x03000	ROM データ(near 領域配置)
.text	(コンパイラ自動配置)	コード部用セクション(near 領域配置)
.RLIB	(コンパイラ自動配置)	ランタイム・ライブラリ コード用セクション
.SLIB	(コンパイラ自動配置)	標準ライブラリ コード用セクション
.textf	(コンパイラ自動配置)	コード部用セクション(far 領域配置)
.constf	(コンパイラ自動配置)	ROM データ(far 領域配置)
.data	(コンパイラ自動配置)	near 初期化データ用セクション(初期値あり)
.sdata	(コンパイラ自動配置)	初期化データ用セクション(初期値あり，saddr 配置変数)
RFD_DATA_n	(コンパイラ自動配置)	RFD 用データ
RFD_CMN_f	(コンパイラ自動配置)	RFD 用共通コード
RFD_CF_f	(コンパイラ自動配置)	RFD 用コード・フラッシュ用コード
RFD_DF_f	(コンパイラ自動配置)	RFD 用データ・フラッシュ用コード
SMP_CMN_f	(コンパイラ自動配置)	RFD サンプル 共通コード
SMP_CF_f	(コンパイラ自動配置)	RFD サンプル コード・フラッシュ用コード
.monitor2	0x1FE00	デバッグ用モニタ領域
FP_DATA	0x34000	データ・フラッシュ書き込み用データ領域 モトローラ S フォーマットから取得したバイナリデータを保存
FP_CODE	0x36000	コード・フラッシュ書き込み用データ領域 モトローラ S フォーマットから取得したバイナリデータを保存
FP_CODE_END	0xB5FFF	コード・フラッシュ書き込み用データ終了アドレス
.dataR	0xF3F00	near 初期化データ用セクション(初期値あり) (RAM 展開)
.bss	(コンパイラ自動配置)	データ領域用セクション(初期値なし，near 領域配置)
RFD_DATA_nR	(コンパイラ自動配置)	RFD 用データ(RAM 展開)
RFD_CMN_fR	(コンパイラ自動配置)	RFD 用共通コード(RAM 展開)
RFD_CF_fR	(コンパイラ自動配置)	RFD 用コード・フラッシュ用コード(RAM 展開)
SMP_CMN_fR	(コンパイラ自動配置)	RFD サンプル 共通コード(RAM 展開)
SMP_CF_fR	(コンパイラ自動配置)	RFD サンプル コード・フラッシュ用コード (RAM 展開)
.sdataR	0xFFE20	初期化データ用セクション(初期値あり，saddr 配置変数)(RAM 展開)
.sbss	(コンパイラ自動配置)	データ領域用セクション(初期値なし，saddr 配置変数)

5.4.1

5.4.1 データ・フラッシュ領域 (0xF1000 - 0xF2FFF, 8KB)

setup コマンドで設定した設定情報は、st_command_data_t 構造体としてデータ・フラッシュの開始アドレス(0xF1000)に保存されます。

プログラマボードは、起動時にデータ・フラッシュを読み出し setup コマンドの設定情報を取得します。

表 5-3 に st_command_data_t 構造体の仕様を示します。

表 5-3 st_command_data_t 構造体

メンバ	説明
e_terminal_command_t com	テンポラリ変数 (ホスト PC から受信したコマンドの種類)
e_uart_if_t uart_if	setup コマンド設定 (通信方式) UART_1LINE: 1 線 UART (初期値) UART_2LINE: 2 線 UART
e_uart_speed_t speed	setup コマンド設定 (ターゲット MCU 通信速度) UART_SPEED_DEFAULT: 115200 bps UART_SPEED_250000: 250000 bps UART_SPEED_500000: 500000 bps UART_SPEED_1000000: 1000000 bps
uint16_t vdd	setup コマンド設定 (VDD 印加電圧)
uint8_t verify	setup コマンド設定 (ベリファイ実行オプション)
uint8_t checksum	テンポラリ変数 (ep コマンドで使用)
uint8_t device_frq	テンポラリ変数 (シグネチャ情報)
uint8_t flash_program_mode	テンポラリ変数 (シグネチャ情報)
uint32_t device_code	テンポラリ変数 (シグネチャ情報)
uint8_t device_name[11]	テンポラリ変数 (シグネチャ情報)
uint32_t code_end	テンポラリ変数 (シグネチャ情報)
uint32_t data_end	テンポラリ変数 (シグネチャ情報)
uint32_t firm_version	テンポラリ変数 (シグネチャ情報)

5.5 スマート・コンフィグレータの設定

スマート・コンフィグレータでの設定を以下に示します。

(1) ポートの初期設定を行います。

- P24, P50, P51 を出力モードに設定

(2) シリアル・アレイ・ユニットの初期設定を行います。

- UART0 を使用 (P12 を TXD0 に、P11 を RXD0 に設定)
- 動作クロックを CK00 に、クロック・ソースを fCLK/2 に設定
- 転送モード設定をシングル転送モードに設定
- データ・ビット長設定を 8bit に設定
- データ転送方向設定を LSB に設定
- パリティ設定をパリティ無しに設定
- ストップビット長設定を 1bit
- 送信データ・レベル設定を非反転に設定
- ボーレート設定を 115200bps に設定

- UART2 を使用 (P13 を TXD2 に、P14 を RXD2 に設定)
- 動作クロックを CK10 に、クロック・ソースを fCLK/2 に設定
- 転送モード設定をシングル転送モードに設定
- データ・ビット長設定を 8bit に設定
- データ転送方向設定を LSB に設定
- パリティ設定をパリティ無しに設定
- ストップビット長設定を 2bit (送信),1bit (受信)
- 送信データ・レベル設定を非反転に設定
- ボーレート設定を 115200bps に設定
- 送信完了割り込み設定(INTST2)をレベル 2 に設定
- 受信完了割り込み設定(INTSR2)をレベル 2 に設定

(3) 割り込み処理の初期設定を行います。

- INTP0 端子の有効エッジを立ち下がりエッジに設定

(4) タイマ・アレイ・ユニットの初期設定を行います。

- インターバル・タイマの動作に 8 ビット・カウンタ・モードを選択
- リソースに TAU0_1 を選択
- 動作クロックに CK02 を選択
- クロック・ソースに fCLK/2 を選択
- 動作モード設定に下位 8 ビットを選択
- インターバル時間に 174us を設定
- タイマ・チャネル 1 のカウント完了で割り込み発生にチェックをつける
- 優先順位をレベル 2 に設定

- インターバル・タイマの動作に 8 ビット・カウンタ・モードを選択
- リソースに TAU0_3 を選択
- 動作クロックに CK03 を選択
- クロック・ソースに fCLK/2^8 を選択
- 動作モード設定に下位 8 ビットを選択
- インターバル時間に 1ms を設定
- タイマ・チャネル 3 のカウント完了で割り込み発生にチェックをつける

- インターバル・タイマの動作に 16 ビット・カウンタ・モードを選択
- リソースに TAU0_4 を選択
- 動作クロックに CK00 を選択
- クロック・ソースに fCLK/2^8 を選択
- インターバル時間に 500ms を設定
- タイマ・チャネル 4 のカウント完了で割り込み発生にチェックをつける

- インターバル・タイマの動作に 8 ビット・カウンタ・モードを選択
- リソースに TAU1_1 を選択
- 動作クロックに CK12 を選択
- クロック・ソースに fCLK/2 を選択
- 動作モード設定に下位 8 ビットを選択
- インターバル時間に 1us を設定
- タイマ・チャネル 1 のカウント完了で割り込み発生にチェックをつける

- インターバル・タイマの動作に 16 ビット・カウンタ・モードを選択
- リソースに TAU1_2 を選択
- 動作クロックに CK10 を選択
- クロック・ソースに fCLK/2^5 を選択
- インターバル時間に 1us を選択
- タイマ・チャネル 2 のカウント完了で割り込み発生にチェックを付ける

(5) 電圧検出回路の初期設定を行います。

- リセット・モードを選択
- リセット発生電圧(VLVD0)に 1.86V を選択

5.6 関数一覧

表 5-4 にサンプル・プログラムで使用する主な関数を示します。

表 5-4 関数一覧

関数名	概要
r_Terminal_Init_Flash	Renesas Flash Driver RL78 Type01の初期化
r_Terminal_Command_Init	コマンド設定情報の初期化
r_Terminal_Command_Init_Dev	コマンド設定情報のシグネチャ情報の初期化
r_Terminal_Command_Read	setupコマンド設定情報の取得
r_Terminal_Command_Recieve	ホストPCからコマンドを受信
r_Terminal_Command_SETUP	setupコマンド処理
r_Terminal_Command_LOD	lodコマンド処理
r_Terminal_Command_EP	epコマンド処理
r_Terminal_Command_SW	オフライン・モードの書き込み処理
r_Terminal_Send_Error_Message	エラーメッセージの送信
r_Terminal_Clear_Write_Buffer	書き込み用バッファのクリア
r_Decord_SRecord	モトローラSフォーマットの解析
R_Config_PORT_Dev_Reset	ターゲットMCUのリセット制御
R_Config_UART01_Terminal_Recv_Through	ホストPCの送信データの読み捨て
R_Config_UART01_Terminal_Set_Echo	ホストPC通信 エコーバックの設定
R_Config_UART01_Terminal_Data_Process_Start	ホストPCから1行受信
R_Config_UART01_Terminal_Data_Process_End	ホストPCから1行受信の終了設定
r_FP_Initial_Communication	ターゲットMCUとの初期通信の実行
r_FP_Get_Signature	シグネチャ情報の取得
r_FP_CMD_Baudrate_A	Baud Rate Setコマンドの実行
r_FP_CMD_Reset_A	Resetコマンドの実行
r_FP_CMD_Signature_A	Silicon Signatureコマンドの実行
r_FP_CMD_Erase_A	Block Eraseコマンドの実行
r_FP_CMD_Program_A	Programmingコマンドの実行
r_FP_CMD_Verify_A	Verifyコマンドの実行
r_FP_CMD_Checksum_A	Checksumコマンドの実行

5.7 関数仕様

サンプル・プログラムで使用する主な関数の仕様を示します。

r_Terminal_Init_Flash	
概　要	Renesas Flash Driver RL78 Type01 の初期化
書　式	int r_Terminal_Init_Flash(void)
引　数	なし
戻り値	RFD ステータス・コード
説　明	Renesas Flash Driver RL78 Type01(RFD)の初期化を行います。

r_Terminal_Command_Init	
概　要	コマンド設定情報の初期化
書　式	void r_Terminal_Command_Init(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	なし
説　明	引数の com_data を初期化します。

r_Terminal_Command_Init_Dev	
概　要	コマンド設定情報のシグネチャ情報の初期化
書　式	void r_Terminal_Command_Init_Dev(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	なし
説　明	引数の com_data のシグネチャ情報を初期化します。

r_Terminal_Command_Read	
概　要	setup コマンド設定情報の取得
書　式	void r_Terminal_Command_Read(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	なし
説　明	データ・フラッシュから setup コマンドの設定情報を取得し、引数の com_data に設定します。

r_Terminal_Command_Recieve	
概　要	ホスト PC からコマンドを受信
書　式	void r_Terminal_Command_Recieve(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	なし
説　明	ホスト PC から送信されたコマンド行を受信し、引数の com_data にコマンド情報を設定します。

r_Terminal_Command_SETUP	
概　要	setup コマンド処理
書　式	void r_Terminal_Command_SETUP(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	なし
説　明	setup コマンドで指定された設定情報をデータ・フラッシュに保存します。

r_Terminal_Command_LOD	
概　要	lod コマンド処理
書　式	int r_Terminal_Command_LOD(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	0: 正常終了 1: 異常終了
説　明	コード・フラッシュの FP_CODE, FP_DATA セクションを消去後、ホスト PC からモトローラ S フォーマットを受信・解析しながら、取得したバイナリデータを FP_CODE, FP_DATA セクションに保存します。

r_Terminal_Command_EP	
概　要	ep コマンド処理
書　式	int r_Terminal_Command_EP(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	0: 正常終了 1: 異常終了
説　明	ep コマンド処理を実行します。 処理の流れは、「3.6.2 オンライン・モード書き込みの流れ」を参照してください。

r_Terminal_Command_SW	
概　要	オフライン・モードの書き込み処理
書　式	int r_Terminal_Command_SW(st_command_data_t * com_data)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	0: 正常終了 1: 異常終了
説　明	オフライン・モードでの書き込み処理を実行します。 処理の流れは、「3.6.3 オフライン・モード書き込みの流れ」を参照してください。

r_Terminal_Send_Error_Message	
概　要	エラーメッセージの送信
書　式	void r_Terminal_Send_Error_Message(uint8_t err)
引　数	uint8_t err: エラーコード
戻り値	なし
説　明	ホスト PC へエラーメッセージを送信します。

r_Terminal_Clear_Write_Buffer	
概　要	書き込み用バッファのクリア
書　式	void r_Terminal_Clear_Write_Buffer(void)
引　数	なし
戻り値	なし
説　明	lod コマンドや ep コマンドで使用する書き込み用バッファ(グローバル変数)を 0xFF で初期化します。

r_Decord_SRecord	
概　要	モトローラ S フォーマットの解析
書　式	void r_Decord_SRecord(uint8_t * srecord_str, uint16_t len, st_srecord_data_t * srecord_data)
引　数	uint8_t * srecord_str: レコードデータ格納バッファ uint16_t len: レコードデータの長さ st_srecord_data_t * srecord_data: レコードデータ情報
戻り値	なし
説　明	モトローラ S フォーマットの 1 行分のレコードデータを解析して、レコードデータ情報としてレコードタイプ、アドレス、データ、データサイズを取得します。

R_Config_PORT_Dev_Reset	
概　要	ターゲット MCU のリセット制御
書　式	void R_Config_PORT_Dev_Reset(RESET_CONTROL reset)
戻り値	なし
引　数	RESET_CONTROL reset: リセット制御 RESET_ENABLE: $\overline{\text{RESET}}$ アクティブ RESET_DISABLE: $\overline{\text{RESET}}$ インアクティブ
説　明	ターゲット MCU のリセットを制御します。

R_Config_UART01_Terminal_Recv_Through	
概　要	ホスト PC の送信データの読み捨て
書　式	void R_Config_UART01_Terminal_Recv_Through(const uint16_t timeout)
引　数	const uint16_t timeout: タイムアウト時間 [ms]
戻り値	なし
説　明	指定タイムアウト時間の間、ホスト PC から送信されるデータを読み捨てます。 lod コマンドや ep コマンドで、異常終了になった場合にモトローラ S フォーマットのデータを読み捨てる用途で使用します。

R_Config_UART01_Terminal_Set_Echo	
概　要	ホスト PC 通信 エコーバックの設定
書　式	void R_Config_UART01_Terminal_Set_Echo(uint8_t echo)
引　数	uint8_t echo: エコーバック設定(有効=1，無効=0)
戻り値	なし
説　明	ホスト PC 通信でのエコーバックの有効・無効を設定します。

R_Config_UART01_Terminal_Data_Process_Start	
概　要	ホスト PC から 1 行受信
書　式	MD_STATUS R_Config_UART01_Terminal_Data_Process_Start(uint8_t ** rx_buf, uint16_t * rx_num, uint8_t * sw)
引　数	uint8_t ** rx_buf: 受信バッファ uint16_t * rx_num: 受信文字数 uint8_t * sw: SW1 押下フラグ(押下あり=1，押下なし=0)
戻り値	MD_OK: 正常終了
説　明	ホスト PC から 1 行分のデータを受信します。 次行のデータを受信する場合は、一度 R_Config_UART01_Terminal_Data_Process_End()を実行してください。

R_Config_UART01_Terminal_Data_Process_End	
概　要	ホスト PC から 1 行受信の終了設定
書　式	void R_Config_UART01_Terminal_Data_Process_End(void)
引　数	なし
戻り値	なし
説　明	R_Config_UART01_Terminal_Data_Process_Start()とペアで使用します。 R_Config_UART01_Terminal_Data_Process_Start()を実行した後に、本関数を実行してください。

r_FP_Initial_Communication	
概　要	ターゲット MCU との初期通信の実行
書　式	uint8_t r_FP_Initial_Communication(st_command_data_t * command)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A に従い、ターゲット MCU との初期通信を行います。 リセット解除、モード情報送信、Baud Rate Set コマンド、Reset コマンドを実行します。

r_FP_Get_Signature	
概　要	シグネチャ情報の取得
書　式	uint8_t r_FP_Get_Signature(st_command_data_t * command)
引　数	st_command_data_t * com_data: コマンド設定情報
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	ターゲット MCU のシグネチャ情報を取得し、引数の com_data に設定します。

r_FP_CMD_Baudrate_A	
概　要	Baud Rate Set コマンドの実行
書　式	uint8_t r_FP_CMD_Baudrate_A(const e_uart_speed_t baudrate, const uint16_t vdd, uint8_t * frq, uint8_t * fpm)
引　数	const e_uart_speed_t baudrate: 通信ボーレート UART_SPEED_DEFAULT: 115200 bps UART_SPEED_250000: 250000 bps UART_SPEED_500000: 500000 bps UART_SPEED_1000000: 1000000 bps const uint16_t vdd: VDD 印可電圧 [100 mV] uint8_t * frq: CPU 動作周波数 [MHz] (ターゲット MCU から取得) uint8_t * fpm: フラッシュ書き換えモード(ターゲット MCU から取得)
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Baud Rate Set コマンドを実行します。

r_FP_CMD_Reset_A	
概　要	Reset コマンドの実行
書　式	uint8_t r_FP_CMD_Reset_A(void)
引　数	なし
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Reset コマンドを実行します。

r_FP_CMD_Signature_A	
概　要	Silicon Signature コマンドの実行
書　式	uint8_t r_FP_CMD_Signature_A(uint8_t * dvc, uint8_t * dev, uint32_t * cfe, uint32_t * dfe, uint8_t * fwv)
引　数	uint8_t * dvc: デバイス機能コード(ターゲット MCU から取得) uint8_t * dev: デバイス名(ターゲット MCU から取得) uint32_t * cfe: コード・フラッシュ領域最終アドレス(ターゲット MCU から取得) uint32_t * dfe: データ・フラッシュ領域最終アドレス(ターゲット MCU から取得) uint8_t * fwv: ブートファームウェアバージョン(ターゲット MCU から取得)
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Silicon Signature コマンドを実行します。

r_FP_CMD_Erase_A	
概　要	Block Erase コマンドの実行
書　式	uint8_t r_FP_CMD_Erase_A(const uint32_t addr)
引　数	const uint32_t addr: 消去ブロックアドレス
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Block Erase コマンドを実行します。

r_FP_CMD_Program_A	
概　要	Programming コマンドの実行
書　式	uint8_t r_FP_CMD_Program_A(const uint32_t start, const uint32_t end, const uint8_t __far * data)
引　数	const uint32_t start: 書き込み開始アドレス const uint32_t end: 書き込み終了アドレス const uint8_t __far * data: 書き込みデータ
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Programming コマンドを実行します。

r_FP_CMD_Verify_A	
概　要	Verify コマンドの実行
書　式	uint8_t r_FP_CMD_Verify_A(const uint32_t start, const uint32_t end, const uint8_t __far * data)
引　数	const uint32_t start: ベリファイ開始アドレス const uint32_t end: ベリファイ終了アドレス const uint8_t __far * data: ベリファイ比較用データ
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Verify コマンドを実行します。

r_FP_CMD_Checksum_A	
概　要	Checksum コマンドの実行
書　式	uint8_t r_FP_CMD_Checksum_A(const uint32_t start, const uint32_t end, uint16_t * checksum)
引　数	const uint32_t start: チェックサム開始アドレス const uint32_t end: チェックサム終了アドレス const uint16_t * checksum: 取得したチェックサムデータ
戻り値	0: 正常終了 0 以外: 異常終了 (「3.5 エラーコード仕様」を参照)
説　明	RL78 プロトコル A の Checksum コマンドを実行します。

6. 参考ドキュメント

RL78/G23 ユーザーズマニュア ルハードウェア編 (R01UH0896)

RL78 マイクロコントローラ (RL78 プロトコル A) シリアルプログラミング編 (R01AN0815)

RL78 ファミリ Renesas Flash Driver RL78 Type01 ユーザーズマニュアル (R20UT4830)

RL78 ファミリ Renesas Flash Driver RL78 Type01 SC 対応仕様 (Code Flash) (R20AN0653)

RL78 ファミリ Renesas Flash Driver RL78 Type01 SC 対応仕様 (Data Flash) (R20AN0654)

RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル (R20UT4870)

(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)

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	2023.02.16	—	初版発行

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. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}(Max.)から V_{IH}(Min.)までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}(Max.)から V_{IH}(Min.)までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス(予約領域)のアクセス禁止
   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、将来の拡張機能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。
8. 製品間の相違について
   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うと、フラッシュメモリ、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、ノイズ幅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。

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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害(当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。)から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為(「脆弱性問題」といいます。)によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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。
14. 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.5.0-1　2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24(豊洲フォレシア)

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。

© 2023 Renesas Electronics Corporation. All rights reserved.