

## RX113 グループ

R01AN2537JJ0102

Rev.1.02

2017.03.31

## USB メモリを利用した内蔵フラッシュメモリの プログラム書き換えソリューション

### RX Driver Package Application

#### 要旨

本書は、USB メモリを利用した内蔵フラッシュメモリのプログラム書き換えソリューションのアプリケーションノートです。

本アプリケーションノートには、USB メモリに格納したプログラムを RX113 内蔵フラッシュメモリに書き込み、それを実行させるメインプログラムが含まれています。

本アプリケーションノートのメインプログラムは、RX110、RX111、RX113、RX231 グループ用 RX Driver Package 内の FAT ファイルシステム、USB ドライバ、フラッシュメモリのモジュールを組み合わせで使用します。

#### 動作確認デバイス

RX113 グループ

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。

#### 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。

- Firmware Integration Technology ユーザーズマニュアル(R01AN1833JU)
- ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685JU)
- e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723JU)
- CS+に組み込む方法 Firmware Integration Technology (R01AN1826JJ)

## 目次

1. 概要.....	3
1.1 本アプリケーションノートについて.....	3
1.2 動作環境.....	3
1.3 モジュール構成.....	4
1.4 ファイル構成.....	5
2. 開発環境の入手.....	6
2.1 e <sup>2</sup> studio の入手とインストール方法.....	6
2.2 コンパイラパッケージの入手方法.....	6
3. プロジェクトの構築.....	7
3.1 ワークスペースの作成.....	7
3.2 プロジェクトの作成.....	8
3.3 プロジェクトのインポート.....	10
3.4 変更情報.....	15
3.4.1 コンフィギュレーションの変更.....	15
3.4.2 プロジェクト設定の変更.....	17
4. 動作確認.....	20
4.1 プロジェクトのビルド.....	20
4.2 デバッグの準備.....	21
4.2.1 機器の構成.....	21
4.2.2 RSK の設定.....	22
4.2.3 USBメモリの準備.....	23
4.3 プロジェクトのデバッグ.....	24
5. アプリケーションの概要.....	28
5.1 メモリ構成.....	29
6. メインプログラム仕様.....	30
6.1 ファイル構成.....	30
6.2 モジュール一覧.....	31
6.3 フローチャート.....	32

## 1. 概要

### 1.1 本アプリケーションノートについて

本アプリケーションノートでは、メインプログラム、RX110、RX111、RX113、RX231 グループ用 RX Driver Package 内蔵の M3S-TFAT-Tiny FAT ファイルシステム（以下、TFAT と略す）、USB ドライバ（ホストマスタストレージクラスドライバ USB HMSC と Basic Firmware）、フラッシュメモリ（以下、Flash API と略す）、ボードサポートパッケージ（以下、BSP と略す）の Firmware Integration Technology（以下、FIT と略す）対応モジュール等を組み合わせて評価するまでの手順について説明します。

本アプリケーションノートは、Renesas Starter Kit for RX113（以降、RSK と表記）上で動作します。

また、書き換え後に実行するプログラム（サンプルプログラム）も用意しています。サンプルプログラムは、各プロジェクト内にある「demo」フォルダに格納しています。

### 1.2 動作環境

メインプログラム、サンプルプログラムの動作環境を表 1.1 に示します。

表1.1 動作環境

項目	内容
使用マイコン	RX113 グループ
使用ボード	Renesas Starter Kit for RX113（製品型名: R0K505113S000BE）
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio Version 4.1.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.2.03.00 以降
エミュレータ	E1
エンディアン	リトルエンディアン
RX Driver Package	RX110、RX111、RX113、RX231 グループ RX Driver Package Ver.1.01 (R01AN2670JJ)（注 1）

注 1. 本アプリケーションノートは、上記の RX Driver Package 内蔵のモジュールと組み合わせて動作させます。本アプリケーションノートで使用するモジュールを、別のモジュールと入れ換えた場合については、各自で動作を確認してください。

### 1.3 モジュール構成

図 1.1にメインプログラムのモジュール構成を、表 1.2にメインプログラムに組み込む FIT モジュールの一覧を示します。一部モジュールは、サンプルプログラムにも組み込まれます。

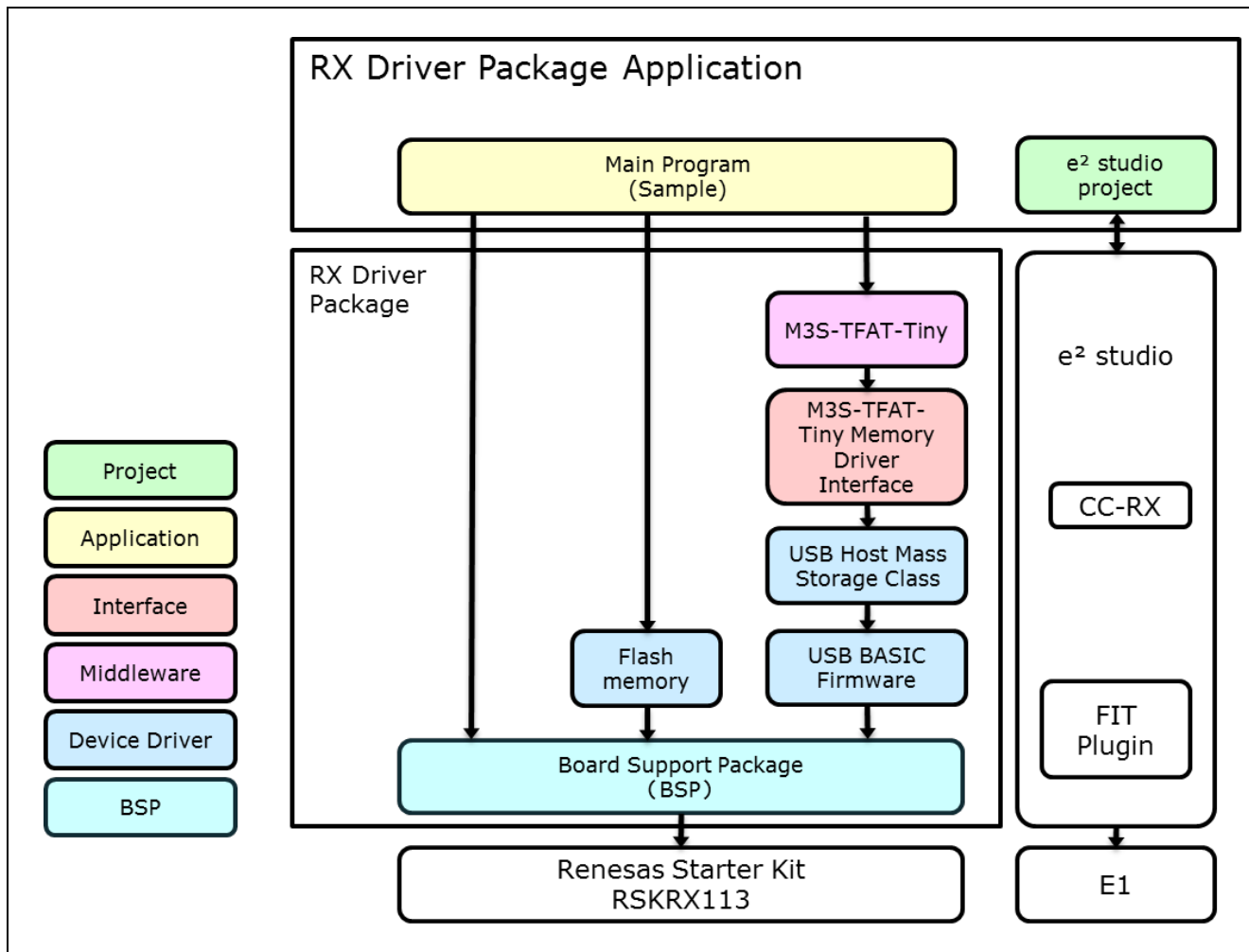


図 1.1 モジュールの構成

表1.2 モジュール一覧

種類	モジュール名	FIT モジュール名	Rev.
BSP	ボードサポートパッケージ(BSP)	r_bsp	3.01
ミドルウェア	M3S-TFAT-Tiny FAT ファイルシステム(TFAT)	r_tfat_rx	3.02
	M3S-TFAT-Tiny メモリドライバインタフェース	r_tfat_driver_rx	1.02
デバイスドライバ	USB Basic Firmware	r_usb_basic_mini	1.01
	USB ホストマスストレージクラス(USB HMSC)	r_usb_hmsc_mini	1.01
	LCD コントローラ/ドライバ(LCDC)	r_lcdc_rx	1.00
	フラッシュメモリ(Flash API)	r_flash_rx	1.30
アプリケーション	メインプログラム(FIT モジュール形式で同梱)	r_flash_writer_rx113	1.00

## 1.4 ファイル構成

図 1.2に本アプリケーションノートのファイル構成を示します。

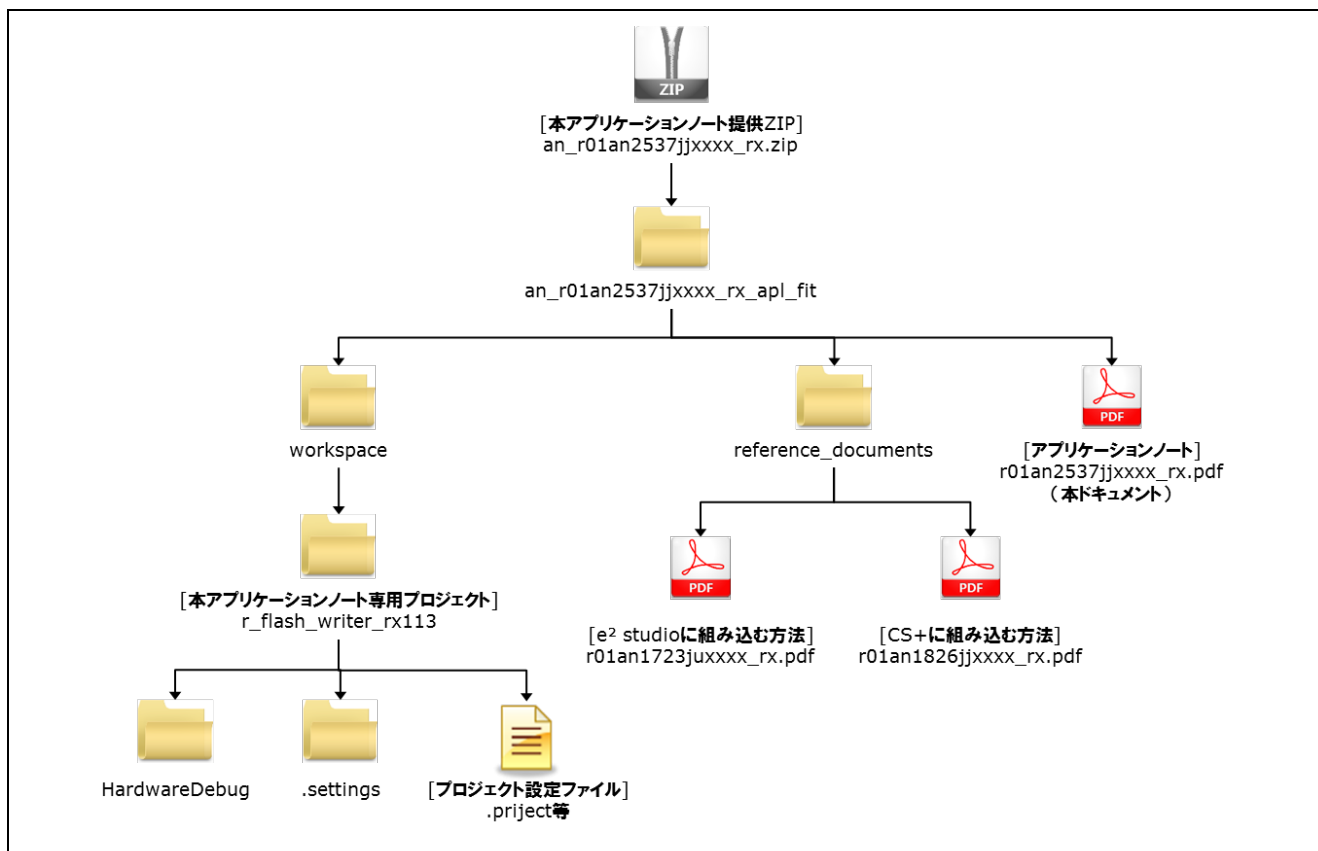


図 1.2 ファイル構成

本アプリケーションノート提供 ZIP ファイルを解凍すると、フォルダの中に以下の各フォルダやファイルが格納されています。

「workspace」フォルダは、「USBメモリを利用した内蔵フラッシュメモリの書き換えアプリケーション」を構築するためのプロジェクトです。e<sup>2</sup> studio を使用する場合は、ワークスペースにインポートして使用します。

「reference\_documents」フォルダの中には、FIT モジュールを各開発環境で使用するための説明書が入っています。「e<sup>2</sup> studio に取り込む方法 (R01AN1723JU)」は、FIT プラグインを使用して、FIT モジュールを e<sup>2</sup> studio のプロジェクトに組み込む方法について説明したドキュメントです。「CS+に取り込む方法 (R01AN1826JJ)」は、FIT モジュールを CS+のプロジェクトに組み込む方法について説明したドキュメントです。

## 2. 開発環境の入手

### 2.1 e<sup>2</sup> studio の入手とインストール方法

以下の URL にアクセスし、e<sup>2</sup> studio をダウンロードしてください。

[http://japan.renesas.com/e2studio\\_download](http://japan.renesas.com/e2studio_download)

なお、本ドキュメントは e<sup>2</sup> studio V4.1.0 以降を使用することを前提としています。V4.1.0 よりも古い Ver. を使用した場合、e<sup>2</sup> studio の一部機能を使用できない可能性があります。ダウンロードする場合、ホームページに掲載されている最新 Ver. の e<sup>2</sup> studio を入手してください。

### 2.2 コンパイラパッケージの入手方法

以下の URL にアクセスし、RX ファミリ用 C/C++コンパイラパッケージをダウンロードしてください。

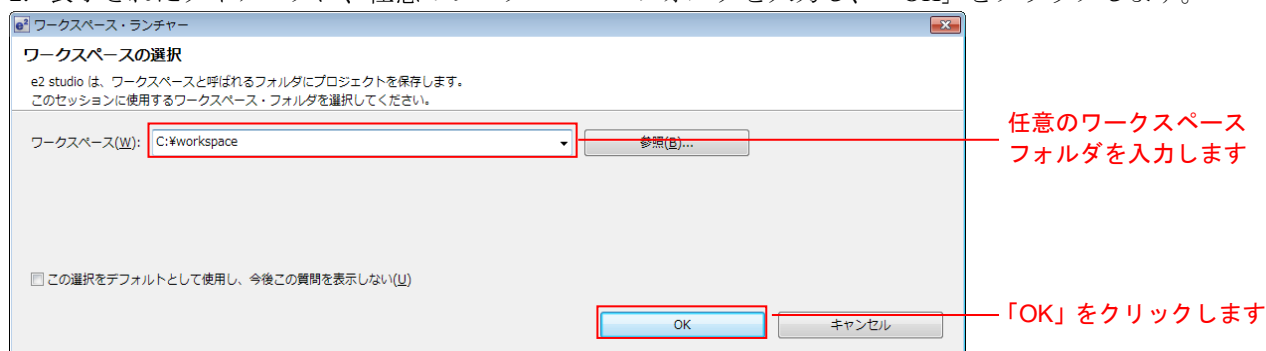
[http://japan.renesas.com/e2studio\\_download](http://japan.renesas.com/e2studio_download)

### 3. プロジェクトの構築

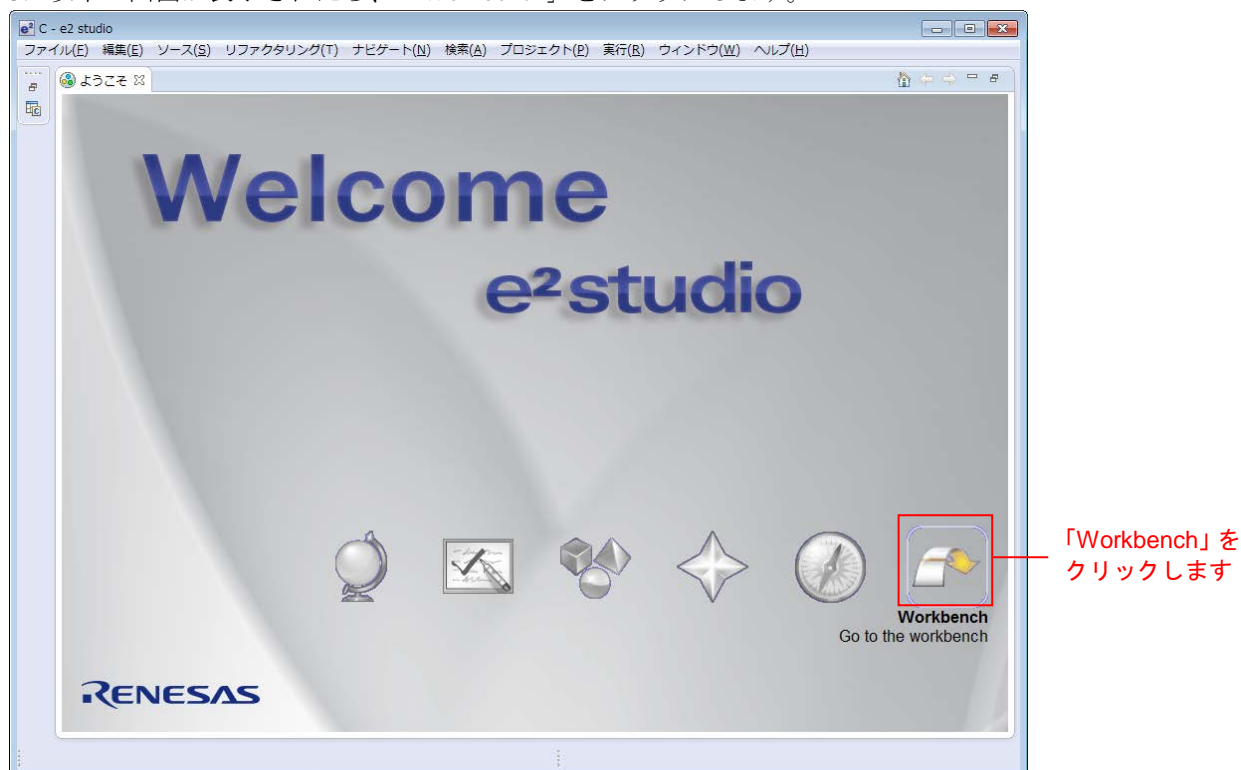
本アプリケーションノートは、環境構築済みのプロジェクトを同梱しています。e2 studio のスマート・ブラウザを用いてプロジェクトをインポートする手順について、以下に説明します。

#### 3.1 ワークスペースの作成

1. e2 studio を起動します。
2. 表示されたダイアログに、任意のワークスペースフォルダを入力し、「OK」をクリックします。



3. 以下の画面が表示されたら、「Workbench」をクリックします。

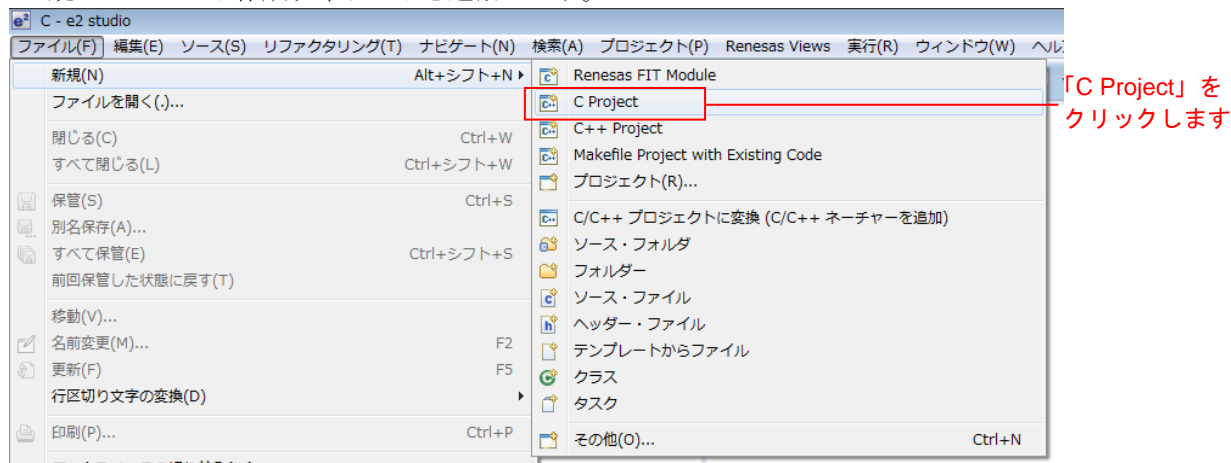


### 3.2 プロジェクトの作成

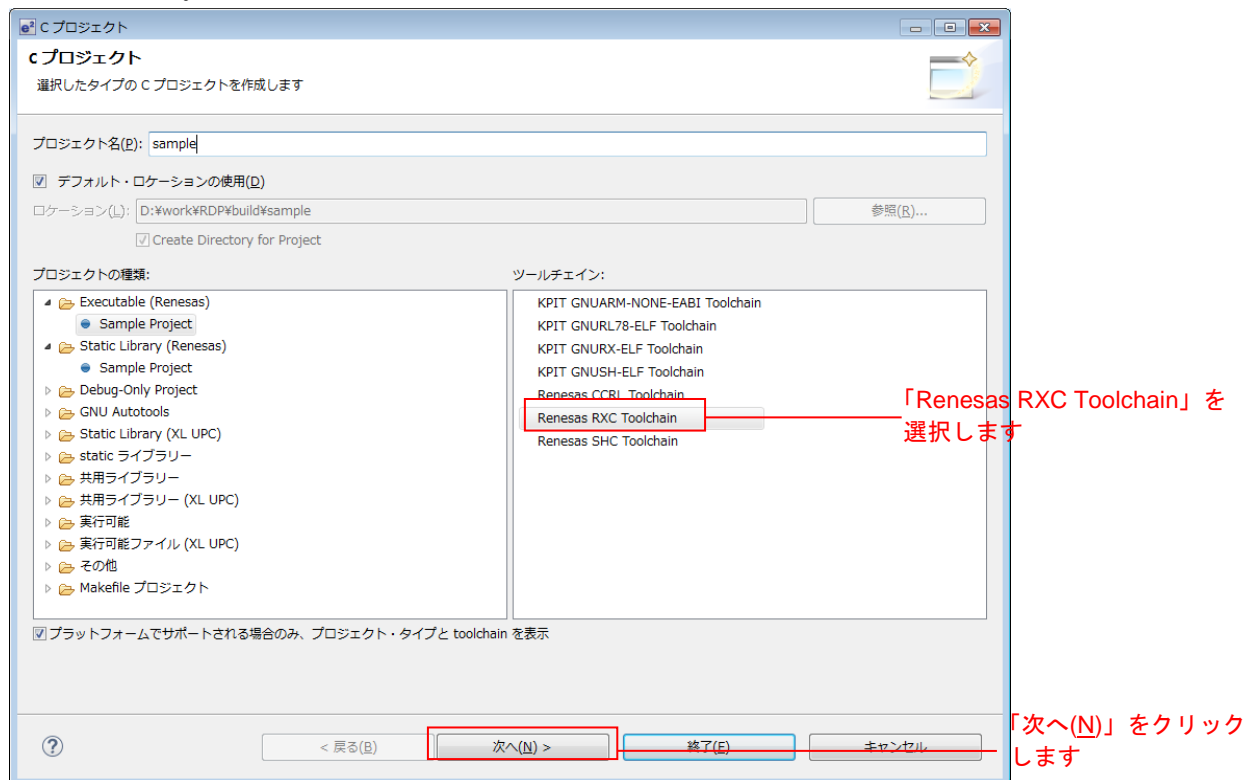
スマート・ブラウザーの機能を使用するときは、対象となるプロジェクトあるいはファイルを選択しておく必要があります。スマート・ブラウザーの機能を使用するために、対象デバイスを指定したプロジェクトを作成します（注1）。

注1：ここで作成するプロジェクトは、スマート・ブラウザーを使用するために作成するためのダミープロジェクトです。

1. [ファイル(F)] → [新規(N)] → [C Project]の順にクリックして新しいCプロジェクトを作成します。新規プロジェクト作成ウィザードを起動します。



2. 任意のプロジェクト名を入力し、“Renesas RXC Toolchain”を選択します。[次へ(N)>] をクリックしてください。

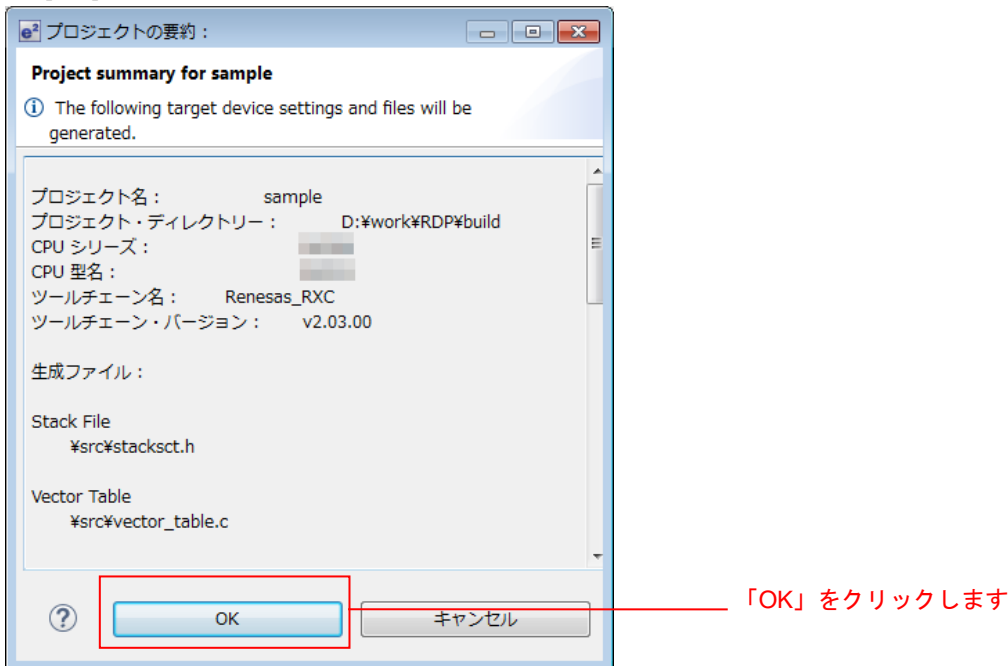




3. 「ターゲットの選択:」を設定します。RX113 100 ピンデバイスの「R5F51138AxFP」としてください。その他の項目は任意の設定で構いません。設定が完了したら[終了(F)]をクリックします。



4. [OK]をクリックします。



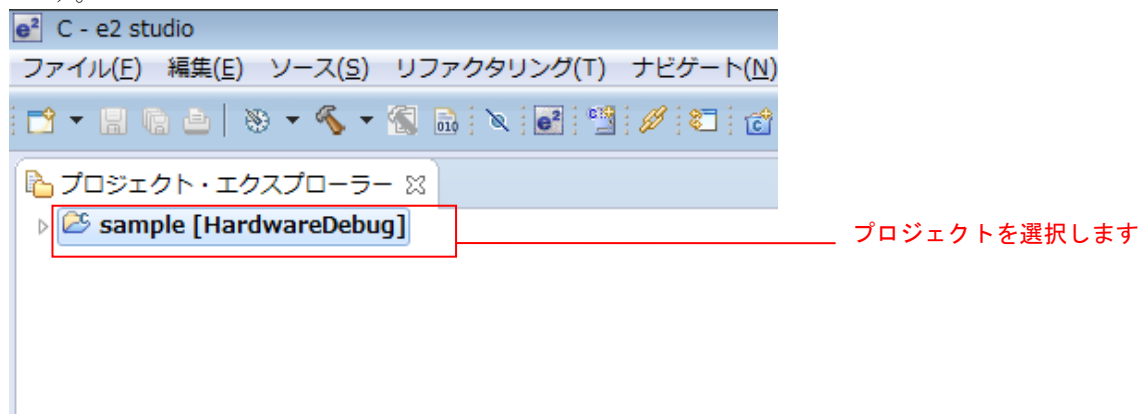
### 3.3 プロジェクトのインポート

メインプログラムのプロジェクトを、作成したワークスペースにインポートします。

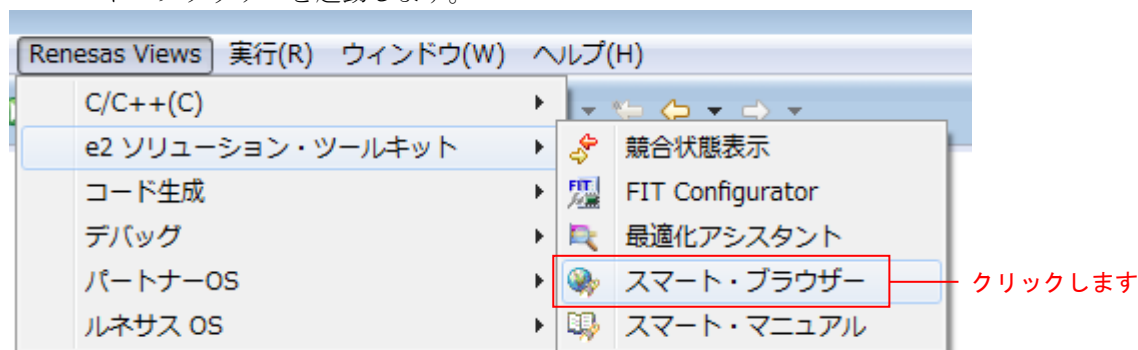
本アプリケーションノートには、以下に示すプロジェクトが含まれます。

- ファイル選択をスイッチで行うプロジェクト

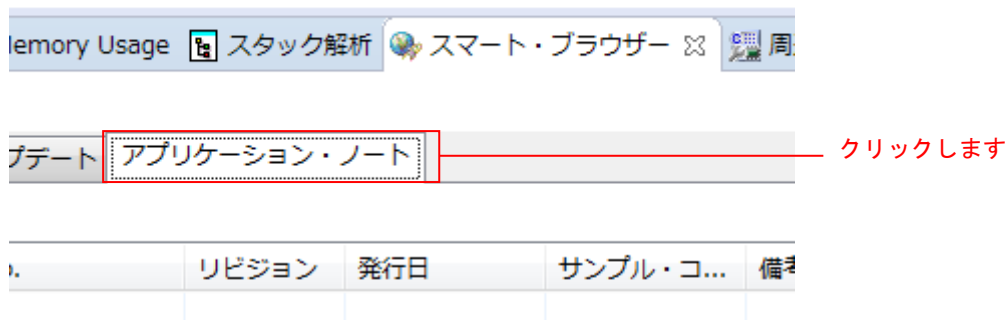
1. プロジェクト・エクスプローラーから「3.2 プロジェクトの作成」で作成したプロジェクトを選択します。



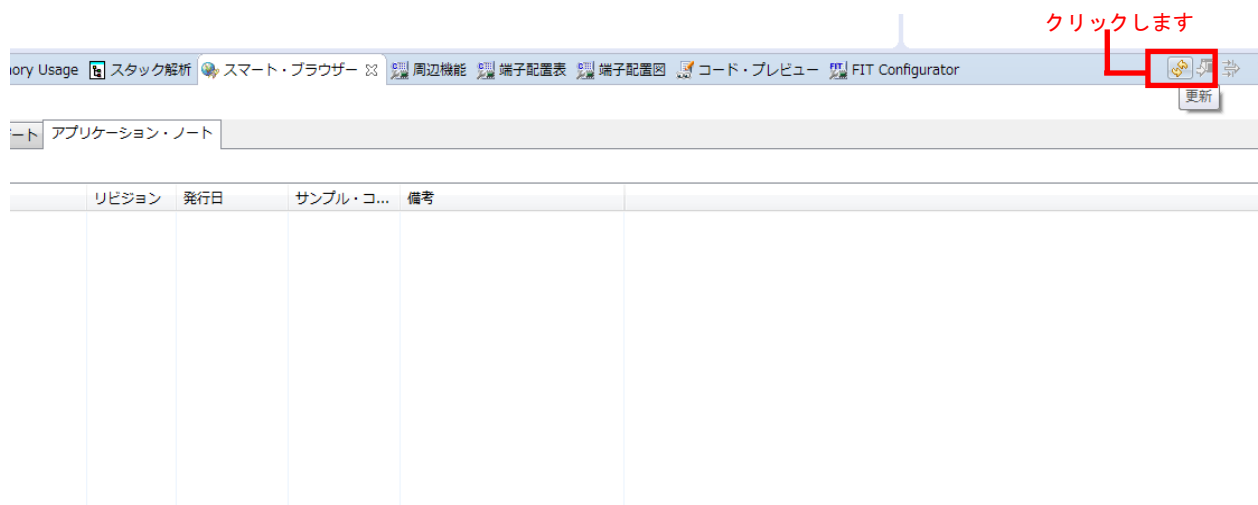
2. [Renesas Views]→[e2 ソリューション・ツールキット]→[スマート・ブラウザー]の順にクリックし、スマート・ブラウザーを起動します。



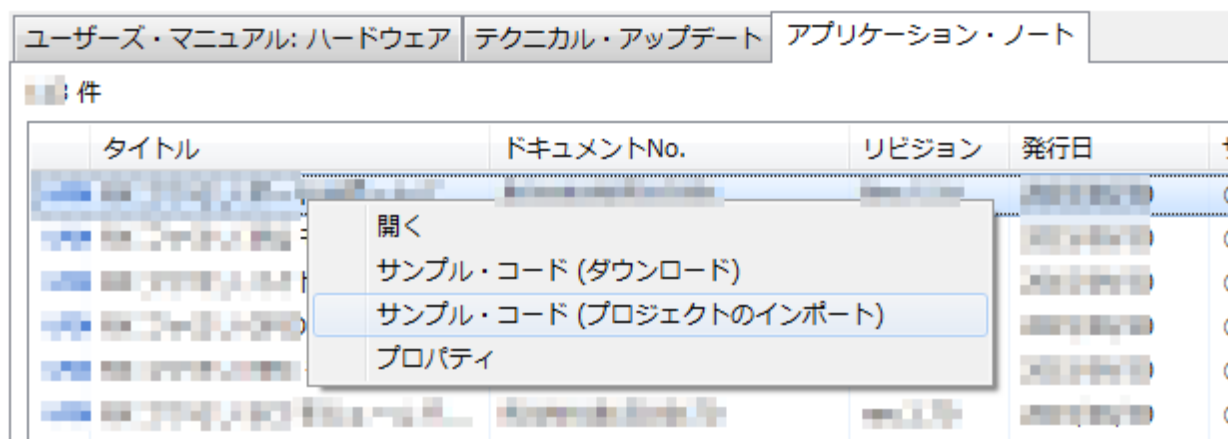
3. [スマート・ブラウザー]タブの[アプリケーション・ノート]タブをクリックします。



4. [更新]  をクリックします。



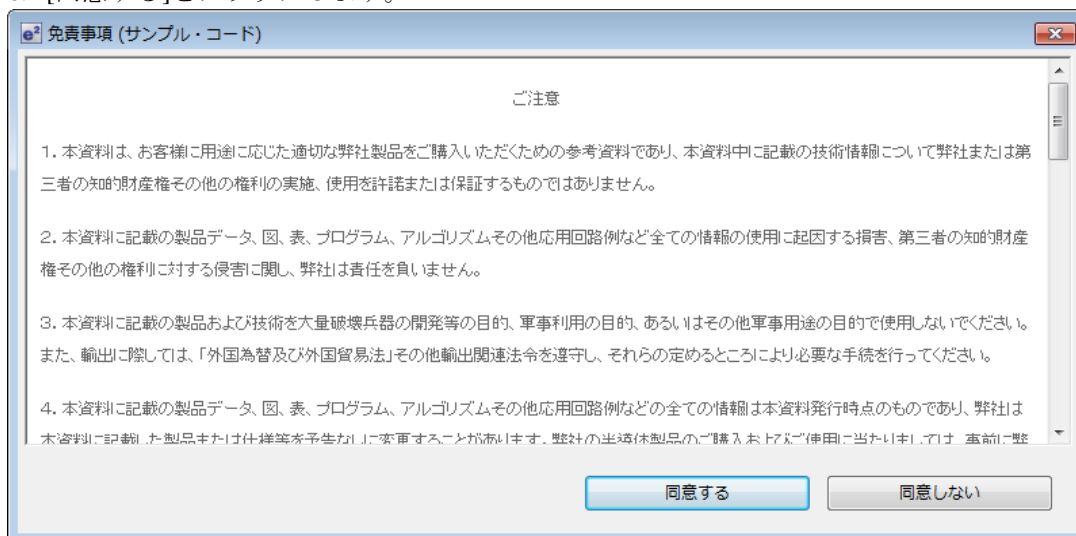
5. 本アプリケーションノートを選択し、右クリックします。コンテキストメニューの[サンプル・コード (プロジェクトのインポート)]をクリックします。



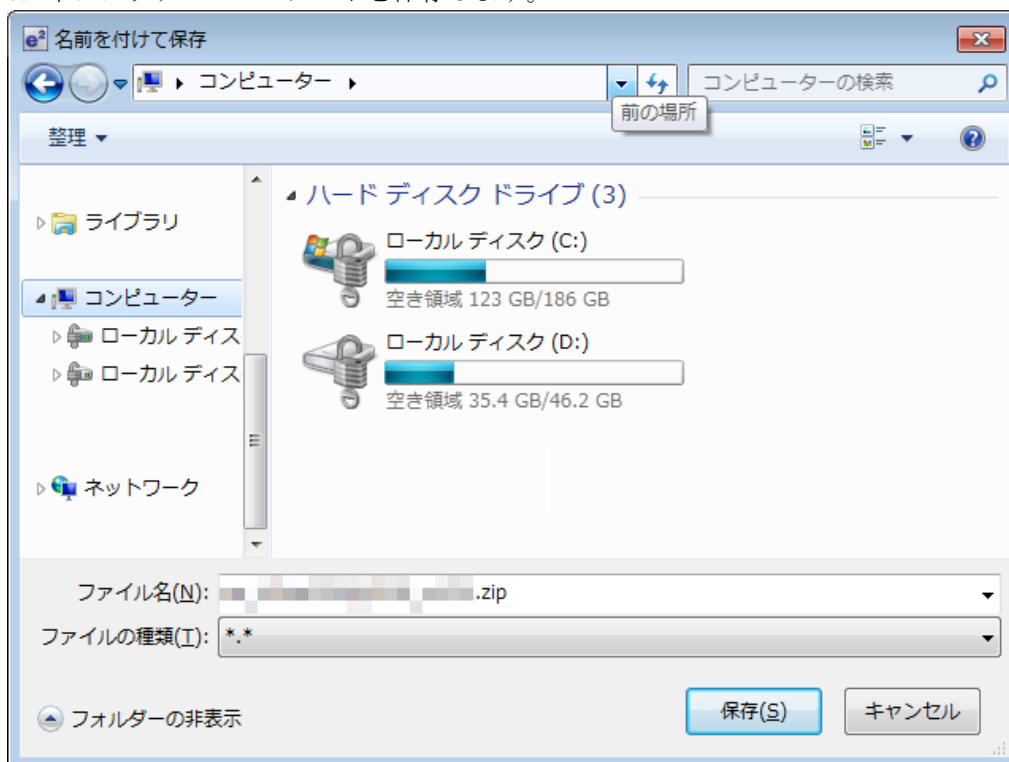
- ※ 一度も My Renesas による認証をしていない場合、ファイルをダウンロードする際に「My Renesas」ダイアログがオープンします。ルネサス Web サイトで登録しているメール・アドレスとパスワードを入力してください。



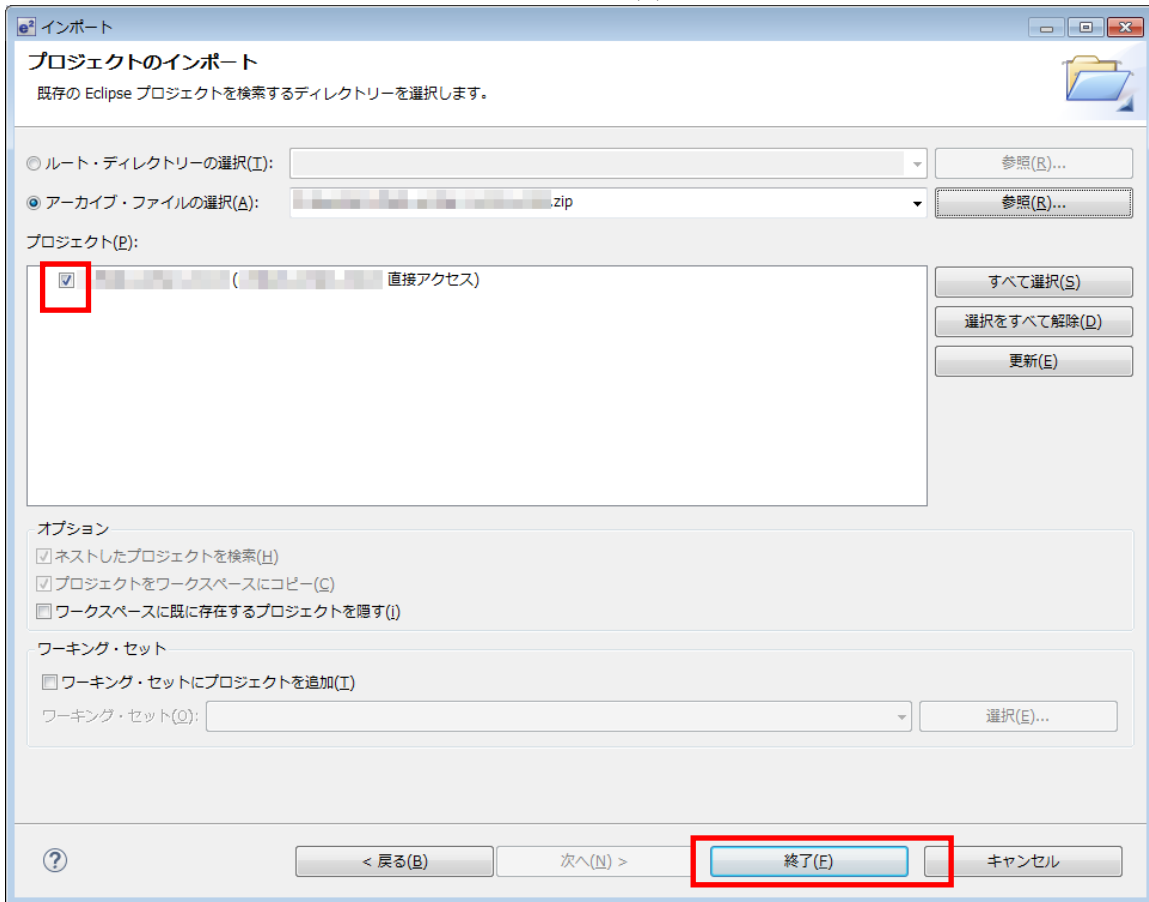
6. [同意する]をクリックします。



7. 本アプリケーションノートを保存します。



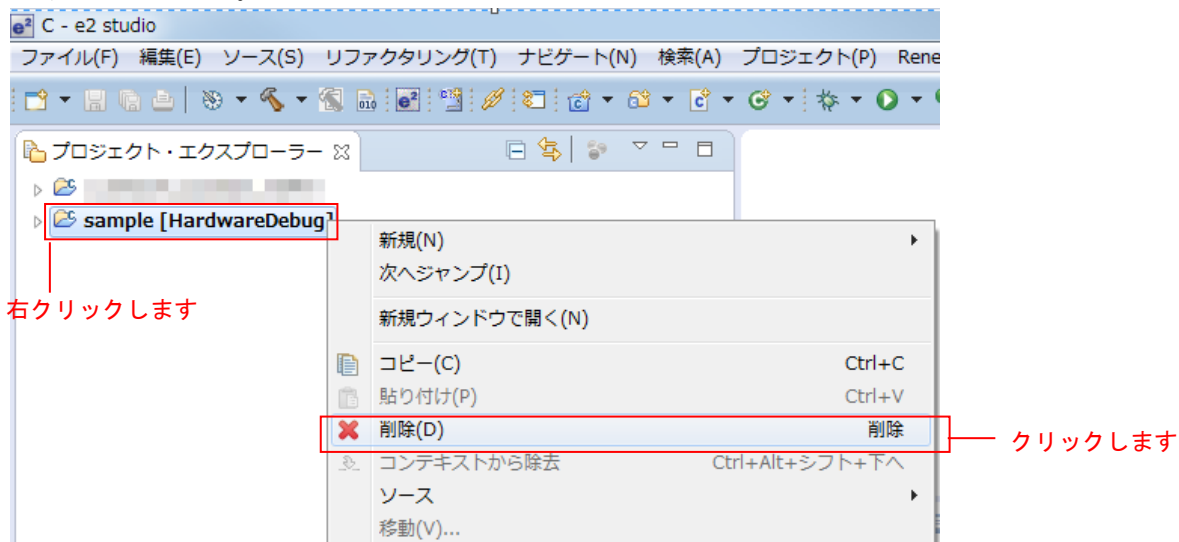
8. インポートするプロジェクトをチェックし「終了(E)」をクリックします。



本アプリケーションノートには以下のプロジェクトが含まれます。

プロジェクト名	内容
r_flash_writer_rx113	スイッチでファイル選択を行うプロジェクト

9. スマート・ブラウザーを使用するために作成したプロジェクト（ここでは sample）は不要ですので削除してください。



### 3.4 変更情報

本プロジェクトでは、本アプリケーションを構成するために各FITモジュールのコンフィグレーションファイル設定とプロジェクト設定を変更しています。以下に詳細を示します。

なお、本変更情報は、新規にプロジェクトを構築する場合に参照してください。インポートしたプロジェクトを使用する場合は「4 動作確認」に進んでください。

#### 3.4.1 コンフィギュレーションの変更

本アプリケーションを構成する各FITモジュールのコンフィギュレーションファイルを変更します。

コンフィギュレーションファイルの項目と設定内容については、各FITモジュールのdocフォルダに入っているマニュアル等を参照してください。

以下にコンフィギュレーションファイルの変更箇所を示します。

##### (1) USB Mini として割り当てるドライブ数の変更

r\_tfat\_driver\_rx のコンフィギュレーションファイルで定義されているUSB Miniとして割り当てるドライブ数を、以下の様に変更してください。

【r\_config/r\_tfat\_driver\_rx\_config.h】

```

/* Number of logical drives to be used.
   Setting to 0      : unused memory
   other            : number of logical drives
   (USB and SDHI can be used together.)
*/
#define TFAT_USB_DRIVE_NUM          (0)
#define TFAT_SDHI_DRIVE_NUM         (0)
#define TFAT_USB_MINI_DRIVE_NUM     (1)

```

##### (2) デバイス割り当ての変更

ドライブ番号に対して、使用するデバイスを割り当てます。本サンプルではドライブ0をUSB Miniを割り当てます。

【r\_config/r\_tfat\_driver\_rx\_config.h】

```

#define TFAT_DRIVE_ALLOC_NUM_0      TFAT_CTRL_USB_MINI
#define TFAT_DRIVE_ALLOC_NUM_1      NULL
#define TFAT_DRIVE_ALLOC_NUM_2      NULL
#define TFAT_DRIVE_ALLOC_NUM_3      NULL
#define TFAT_DRIVE_ALLOC_NUM_4      NULL
#define TFAT_DRIVE_ALLOC_NUM_5      NULL
#define TFAT_DRIVE_ALLOC_NUM_6      NULL
#define TFAT_DRIVE_ALLOC_NUM_7      NULL

```

### (3) DTC 転送設定の変更

r\_usb\_basic\_mini\_config.h には、以下の DTC 定義が記載されています。

サンプルでは DTC 転送を行わない為、” USB\_NOUSE” 定義を有効に変更します。

【r\_config/r\_usb\_basic\_mini\_config.h】

```
/* DTC DEFINE */
#define DTC_USE_PIPE_NUM    USB_NOUSE
// #define DTC_USE_PIPE_NUM    USB_PIPE1
// #define DTC_USE_PIPE_NUM    USB_PIPE2
// #define DTC_USE_PIPE_NUM    USB_PIPE3
// #define DTC_USE_PIPE_NUM    USB_PIPE4
// #define DTC_USE_PIPE_NUM    USB_PIPE5
```

### (4) TFAT 設定の変更

r\_usb\_hmsc\_mini\_config.h には TFAT 定義が記載されています。TFAT を使用するために以下のマクロを有効に変更します。

【r\_config/r\_usb\_hmsc\_mini\_config.h】

```
#define USB_TFAT_USE_PP
```

### (5) Flash API の設定変更

code flash を書き換え可能にしてください。

【r\_config/r\_flash\_rx\_config.h】

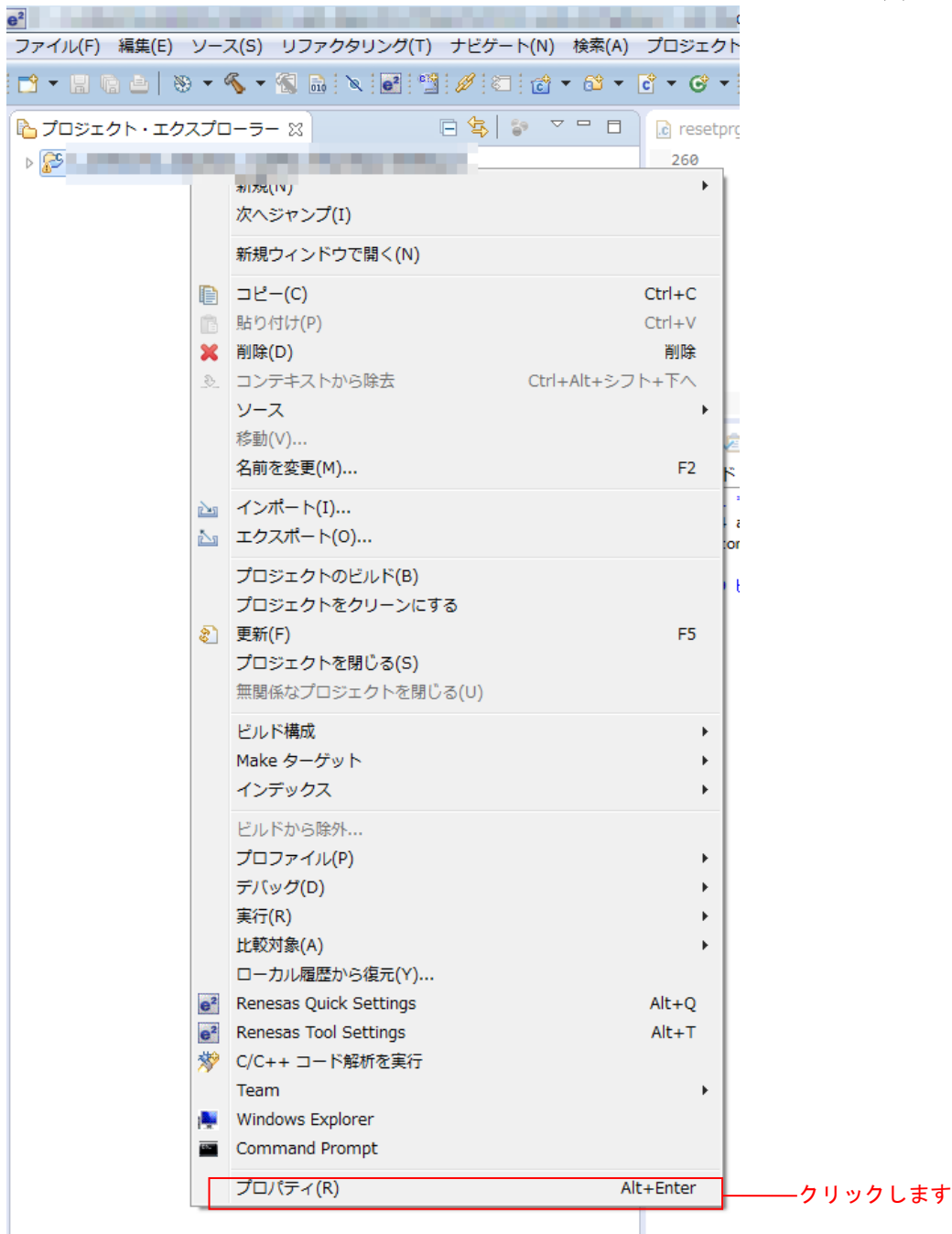
```
#define FLASH_CFG_CODE_FLASH_ENABLE (1)
```



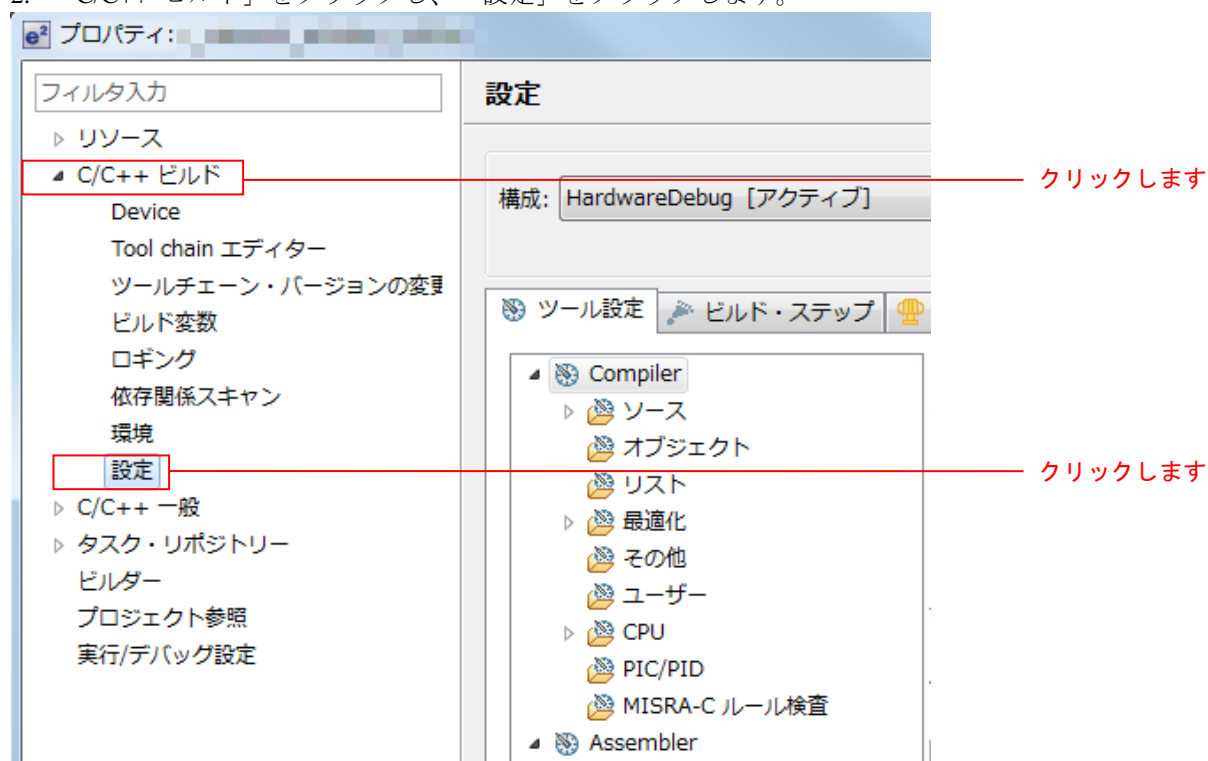
### 3.4.2 プロジェクト設定の変更

プロジェクト設定のデフォルト設定からの変更内容を示します。プロジェクト設定を確認する場合は、以下の手順で行ってください。

1. e<sup>2</sup> studio の対象プロジェクトを選択し右クリックします。その後「プロパティ(R)」をクリックします。



2. 「C/C++ ビルド」をクリックし、「設定」をクリックします。



— メインプログラムのプロジェクト設定

メインプログラムのビルド時の設定はデフォルト設定から表 3.1に示す内容に、デバッグ時の設定はデフォルト設定から表 3.2に示す内容に変更しています。

表3.1 変更したビルド設定

項目	変更内容	説明
Compiler -オブジェクト	「デバッグ情報を生成する」にチェックする。	デバッグ時に必要なデバッグ情報を出力します。
Assembler -オブジェクト	「デバッグ情報を生成する」にチェックする。	リロケータブルファイルにデバッグ情報を出力します。
Linker -入力	"\${workspace_loc}/\${ProjName}/r_tfat_rx/lib/tfat_rx200_little.lib"を追加する(注)。	TFAT を使用する際には設定が必要です。(TFAT を使用する際に必須)
Linker -セクション	PRresetPRG、PIntPRG をセクション定義から除去する(注)。	BSP を使用する際には設定が必要です。(FIT を使用する際には必須)
	P セクションを P*セクションに変更する(注)。	BSP を使用する際には設定が必要です。(FIT を使用する際には必須)
	RPFRAM セクションを R セクションの後ろに追加する(注)。	Flash API が使用する領域を設定が必要です。(Flash API を使用する際に必須)
Linker -出力	ROM から RAM へマップするセクションに PFRAM=RPFRAM を追加する(注)。	Flash API が使用する領域を設定が必要です。(Flash API を使用する際に必須)

注 BSP、TFAT、Flash API の各 FIT モジュールを組み込むプロジェクトを作成する際に必要な設定変更です。この設定については各 FIT モジュールの doc フォルダに入っているマニュアル等を参照してください。

表3.2 変更したデバッグ設定

項目	変更内容	説明
Debugger - デバッグ・ツール 設定	内蔵プログラム ROM を書き換える を「はい」に変更する。	内蔵フラッシュメモリを書き換えるプログラム をデバッグする際には、必須です。

— サンプルプログラムのプロジェクト設定

sample のビルド時のデフォルト設定からの変更点を表 3.3に示します。

表3.3 変更したビルド設定(sample)

項目	変更内容	説明
Linker -セクション	PResetPRG、PIntPRG を セクション定義から除去する（注）。	BSP を使用する際には設定が必要です。 （FIT を使用する際には必須）
	P セクションを P*セクションに変更 する（注）。	BSP を使用する際には設定が必要です。 （FIT を使用する際には必須）
	C_1 セクションのアドレスを “0xFFFF 0000” に変更する。	書き込む先頭のアドレスを指定します。
	FIXEDVECT セクションのアドレスを “0xFFFF BF80” に変更する。	固定ベクタテーブルの先頭アドレスを指定しま す。
Linker -出力	出力ファイル・タイプを “Binary via absolute” に変更する。	USB メモリに書くファイルの形式を設定しま す。

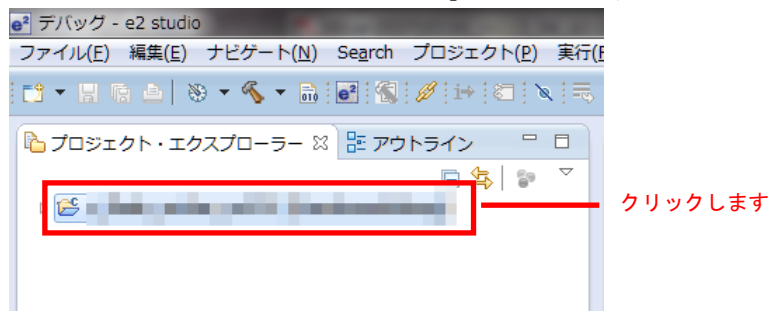
注 BSP FIT モジュールを組み込むプロジェクトを作成する際に必要な設定変更です。この設定については BSP FIT モジュールの doc フォルダに入っているマニュアル等を参照してください。

## 4. 動作確認

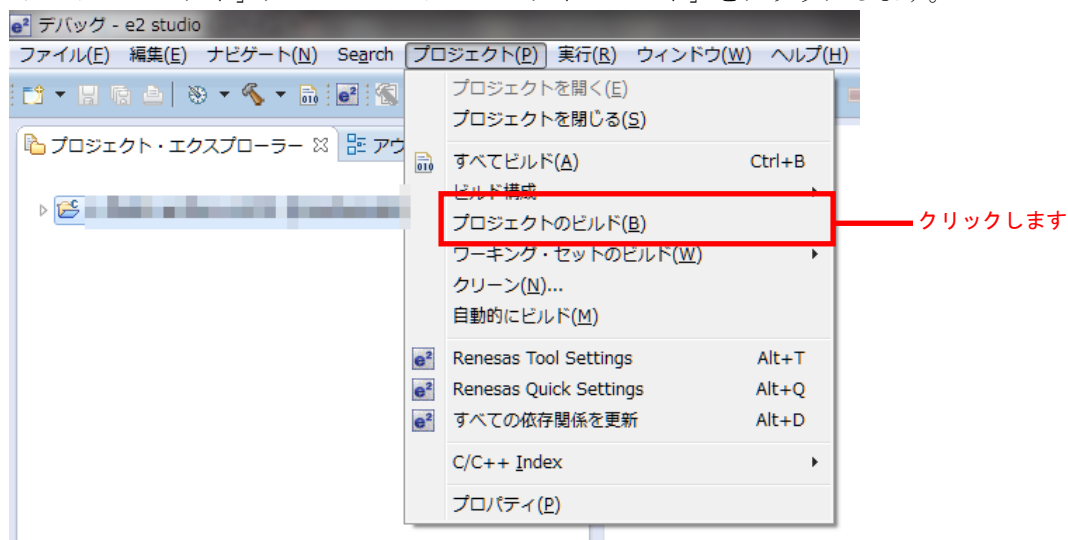
### 4.1 プロジェクトのビルド

以下の手順に従い、プロジェクトをビルドしてロードモジュールを生成します。

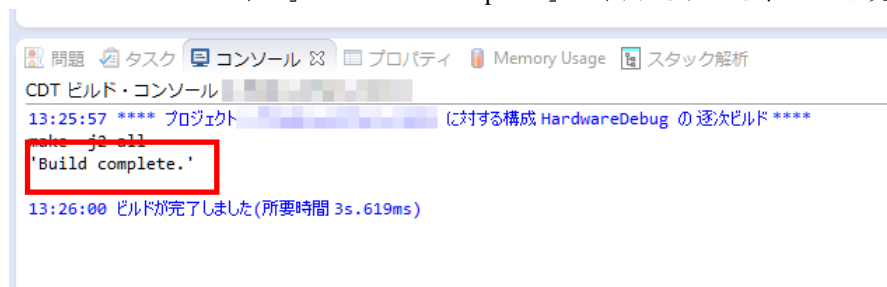
1. 「プロジェクト・エクスプローラ」からビルドするプロジェクトをクリックします。



2. 「プロジェクト」メニューの「プロジェクトのビルド」をクリックします。



3. 「コンソールパネル」に「Build complete.」と表示されたら、ビルド完了です。



## 4.2 デバッグの準備

### 4.2.1 機器の構成

デバッグを開始する前に、評価環境を準備します。

必要な機器の一覧と構成を以下に示します。

表4.1 必要な機器の一覧

機器	補足
開発 PC	
RSK	評価ボード
E1 エミュレータ	Renesas Starter Kit for RX113 に同梱
USB メモリ	FAT、または FAT32 でフォーマットしたもの。

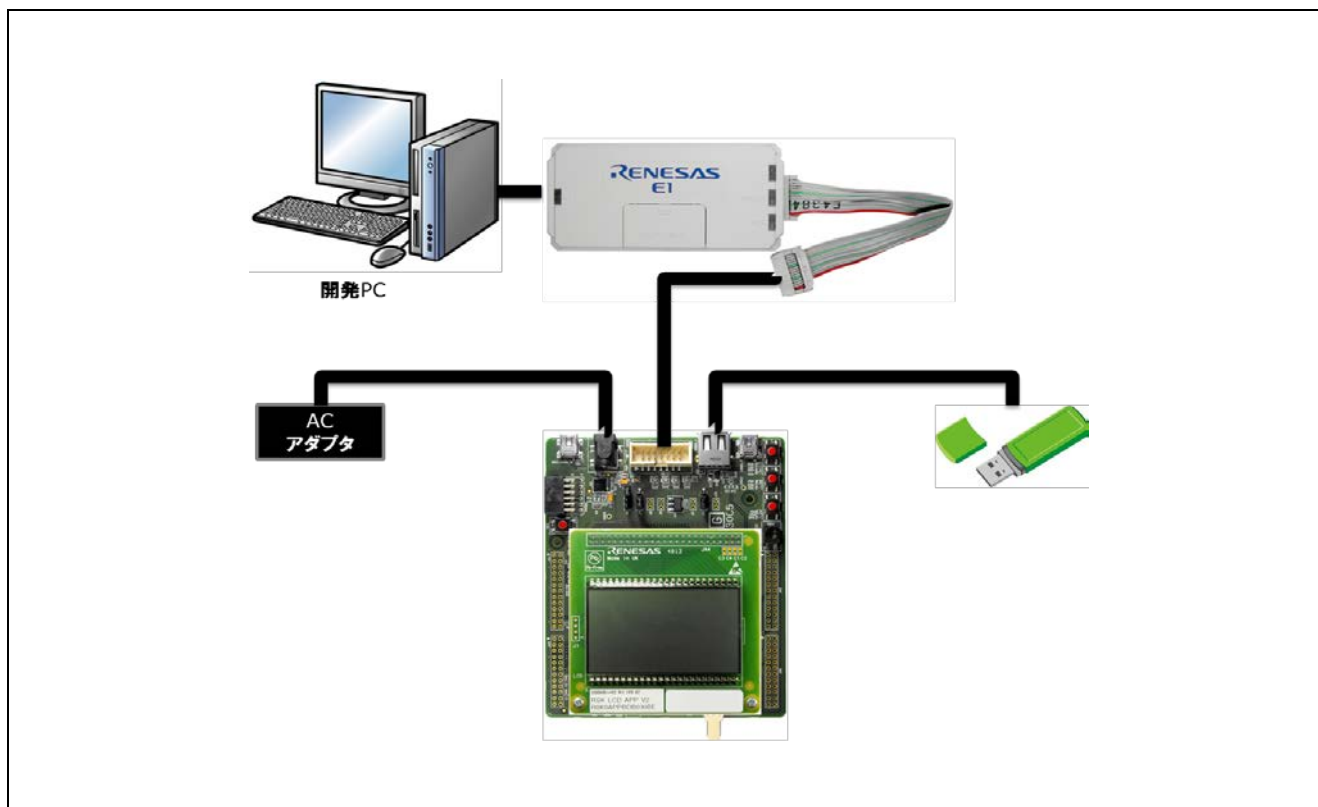


図 4.1 動作環境例

#### 4.2.2 RSK の設定

メインプログラムを動作させるために必要な RSK の設定を以下に示します。

USB のモード (ホスト/ペリフェラル) をジャンパ J15 で設定します。r\_usb\_basic\_mini\_config.h の「USB\_FUNCSEL\_PP」の設定に合わせて設定してください。

表4.2 ジャンパ設定

機器	ジャンパ	設定内容
USB をホストモードで使用する場合 (USB_FUNCSEL_PP = USB_HOST_PP)	J15	1-2 をショート (※今回は、こちらを選択します)
USB をペリフェラルモードで使用する場合 (USB_FUNCSEL_PP = USB_PERI_PP)	J15	2-3 をショート

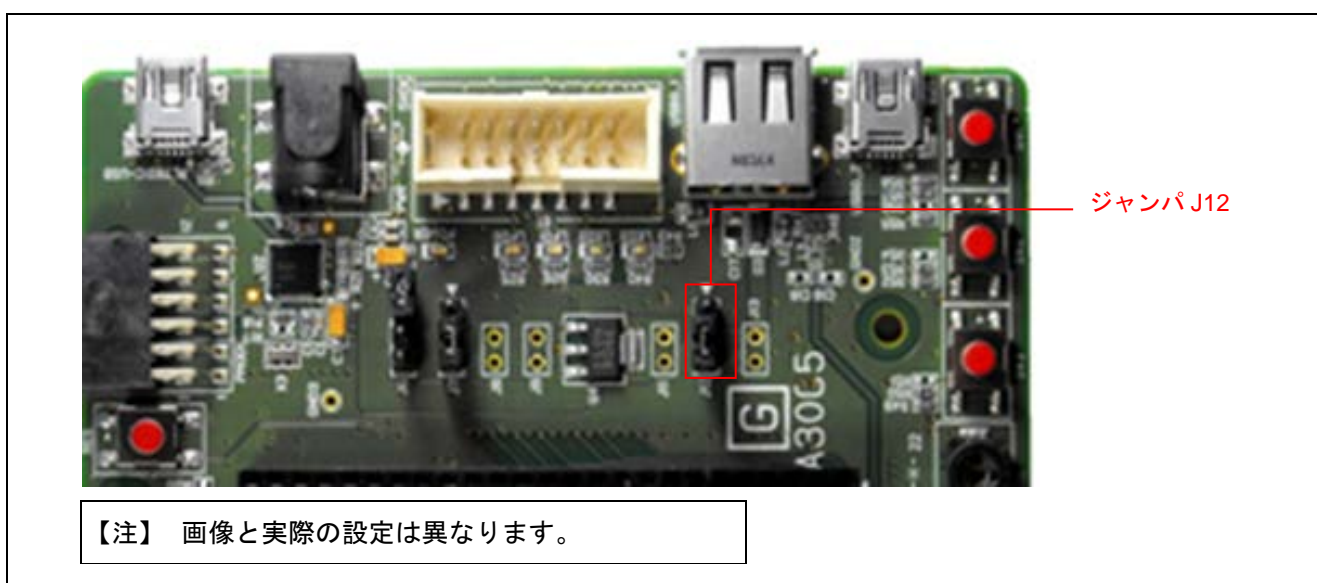
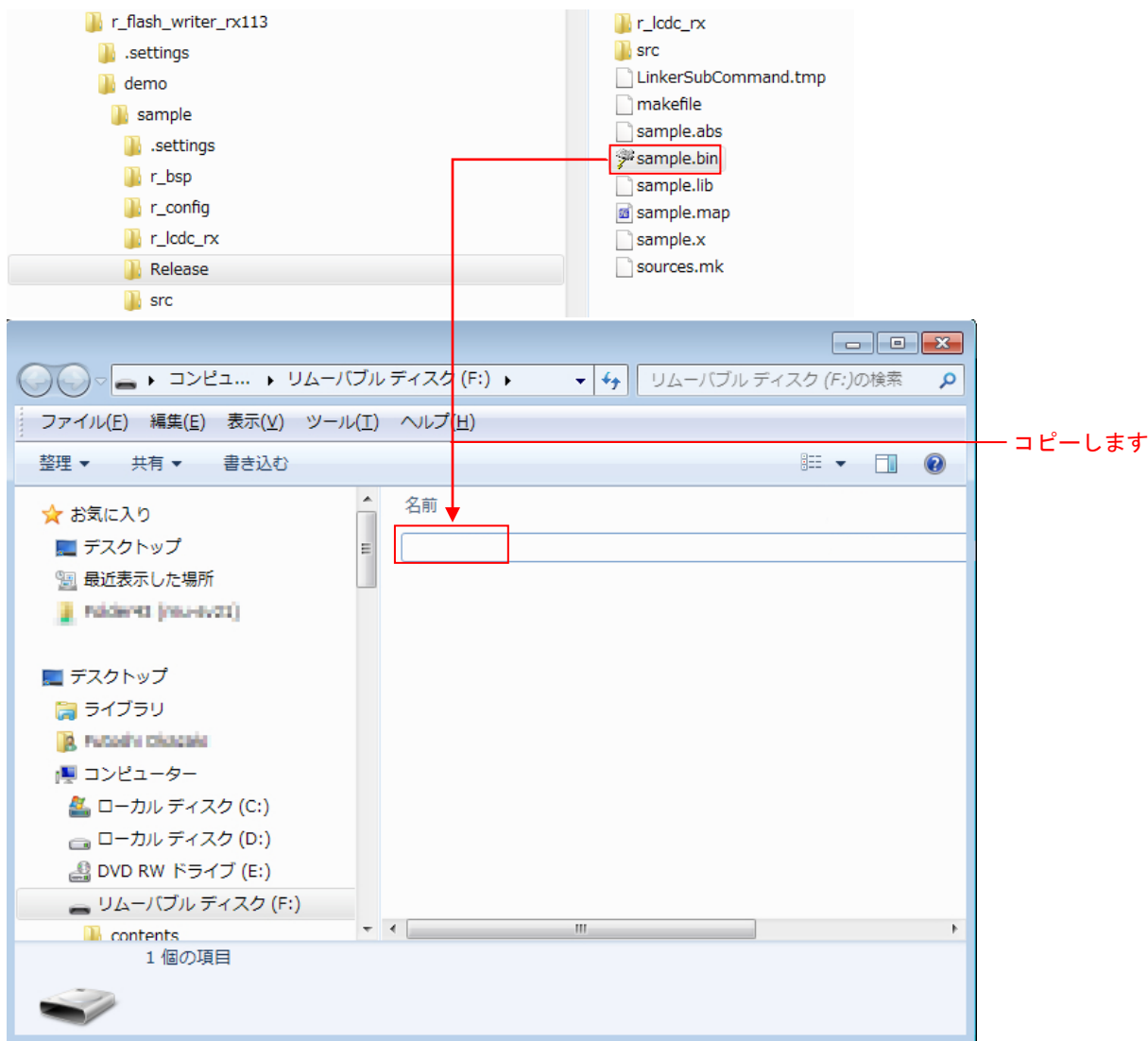


図 4.2 RSK でのジャンパの位置

### 4.2.3 USBメモリの準備

USBメモリにサンプルプログラムのバイナリファイルを格納しておきます。

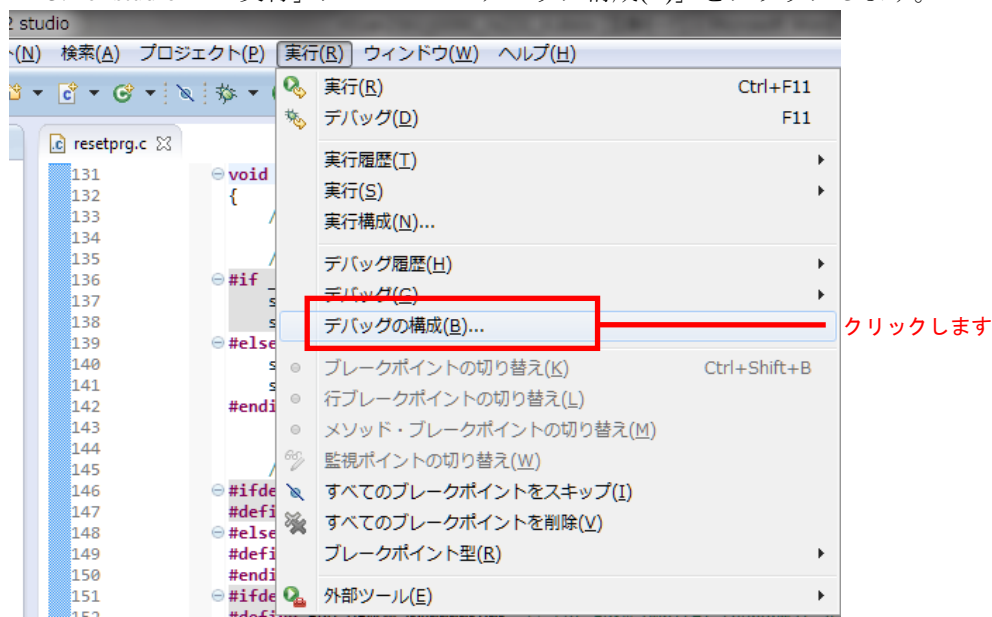
メインプログラムのプロジェクト内にある「demo」フォルダを開き、その中にある「sample.zip」ファイルを任意のフォルダに解凍します。解凍したフォルダ「sample\release」フォルダに入っている「sample.bin」ファイルをUSBメモリにコピーします。



### 4.3 プロジェクトのデバッグ

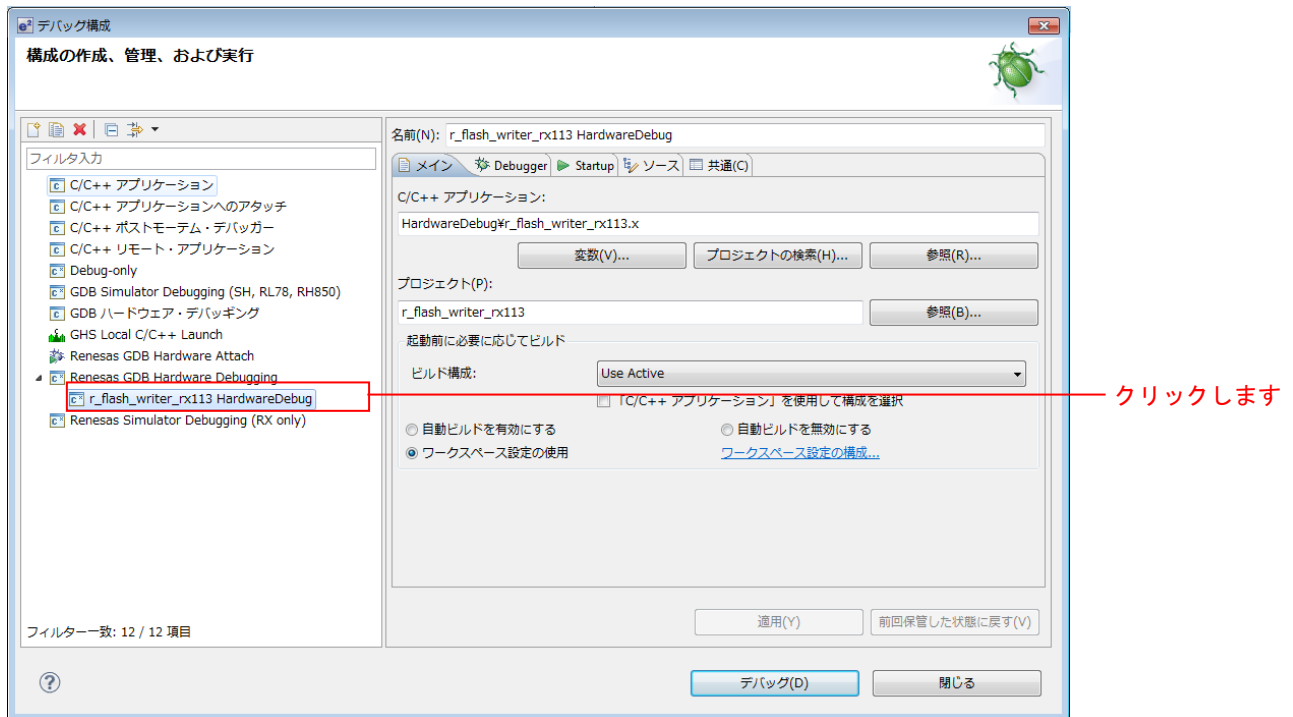
以下の手順に従い、プロジェクトのデバッグを開始します。

1. 開発PCとE1エミュレータをUSBケーブルで、E1エミュレータとRSKをユーザシステムインタフェースケーブルで接続します。
2. RSKとACアダプタを接続して、電源を入れます。
3. e<sup>2</sup> studioの「実行」メニューの「デバッグ構成(B)」をクリックします。

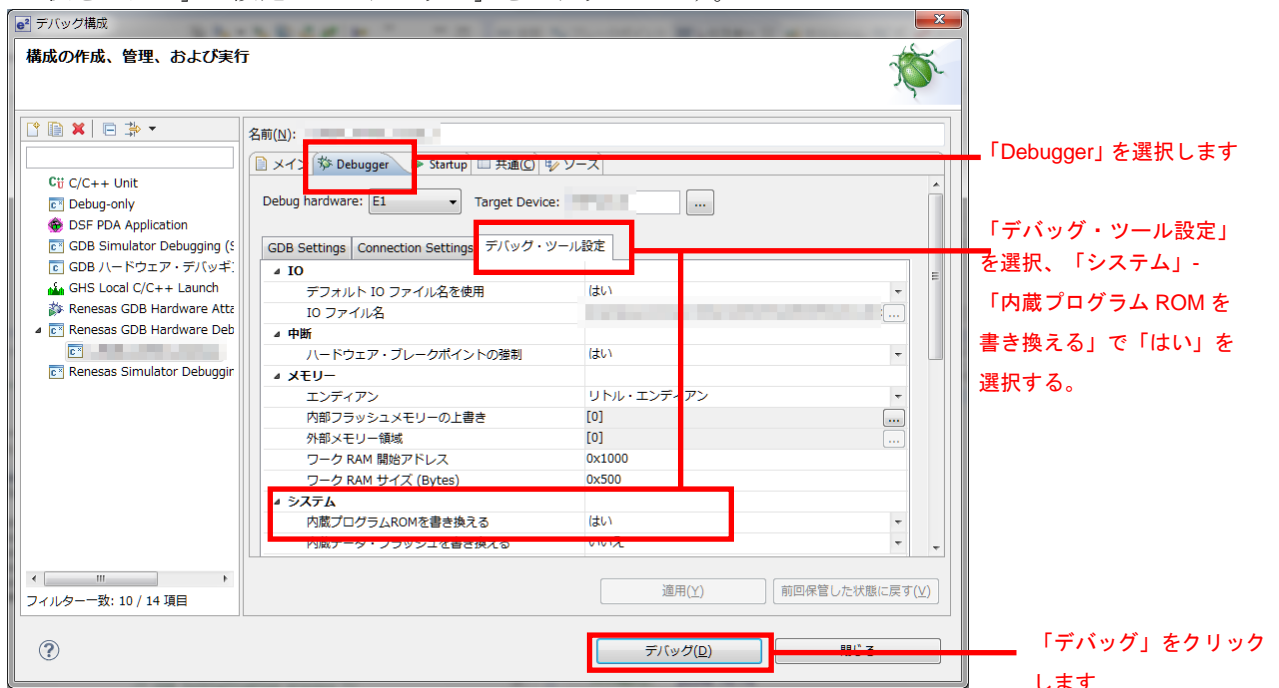




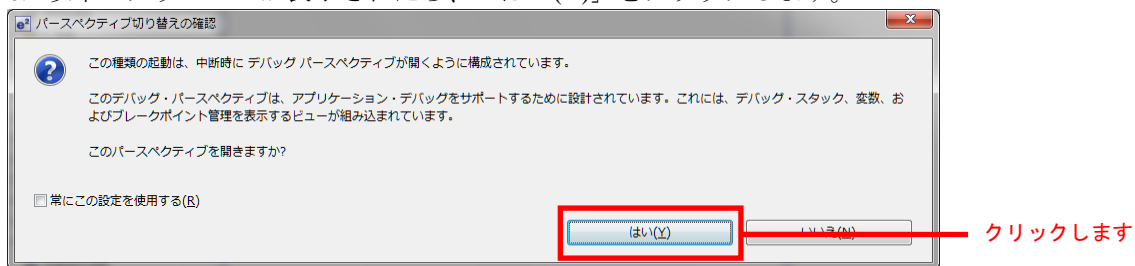
4. 「Renesas GDB Hardware Debugging」の「r\_flash\_writer\_rx113.x」をクリックし、「デバッグ(D)」をクリックします。



5. 「デバッグ・ツール設定」をクリックし、「システム」—「内蔵プログラムROMを書き換える」の選択を「はい」に設定して「デバッグ」をクリックします。

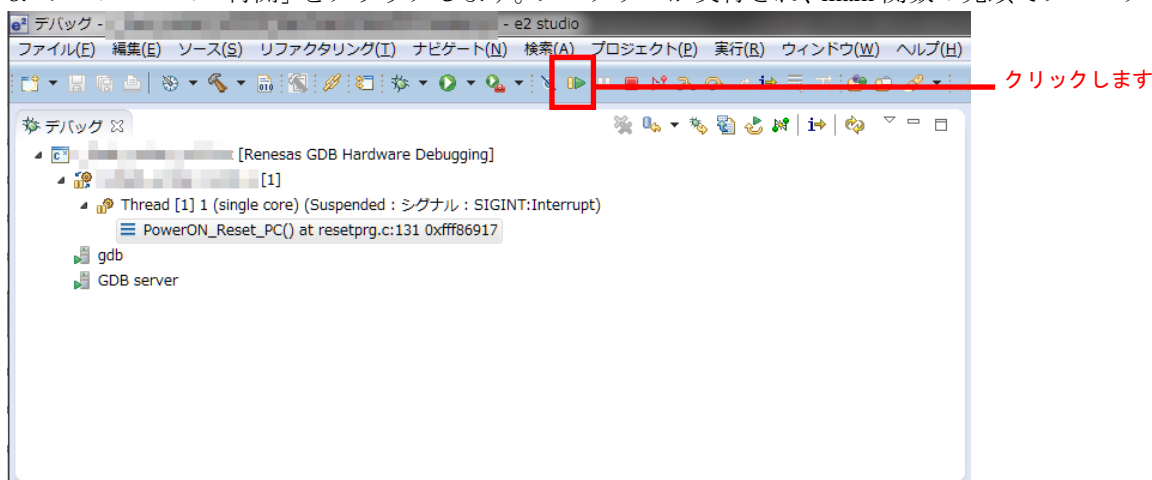


6. 以下のメッセージが表示されたら、「はい(Y)」をクリックします。



7. ロードモジュールのダウンロードが完了すると、「デバッグ」パースペクティブが開きます。

8. ツールバーの「再開」をクリックします。プログラムが実行され、main 関数の先頭でブレークします。

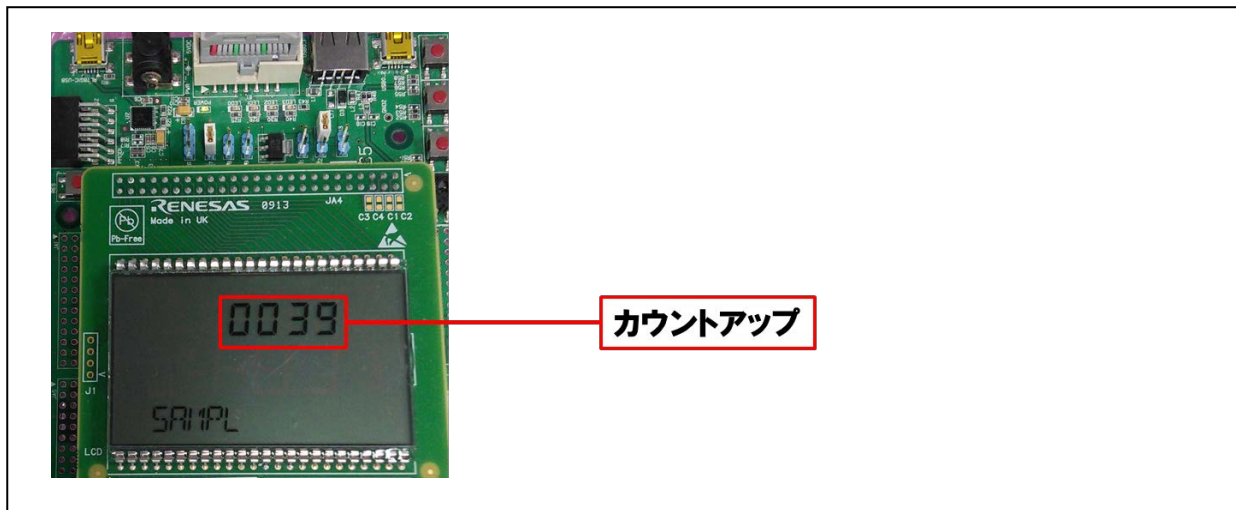


9. main 関数の先頭でブレークした後に、もう一度ツールバーの「再開」をクリックします。

USB メモリを挿し込みます。

自動で、USB メモリからデータが読み出されマイコン内蔵フラッシュメモリに書き込まれます。

再起動され下記画像のように LCD がカウントアップされていれば正常終了です。



## 5. アプリケーションの概要

本サンプルプログラムは、RSK LCD の 7 SEG 表示部を 1 秒毎にカウントアップするプログラムです。特別なメモリを必要とせずマイコン内蔵メモリのみで動作します。

本サンプルプログラムでは、RSKRX113 の ROM/RAM 容量に合わせ適切なメモリ設定を行っています。

- 1 : LCDドライバ初期化
- 2 : LCD表示部に「SAMPL」文字の表示
- 3 : 1 秒Wait処理
- 4 : LCD表示部 (7SEG) にカウントアップ表示「0000~9999」
- 5 : 3と4を繰り返す

注 :

本サンプルプログラムは、メインプログラムの動作検証用に作成した為、メインプログラムのメモリ未使用領域にて動作させます。

## 5.1 メモリ構成

本アプリケーションで使用する RSKRX113 に搭載している RX113 マイコンのメモリマップを示します。

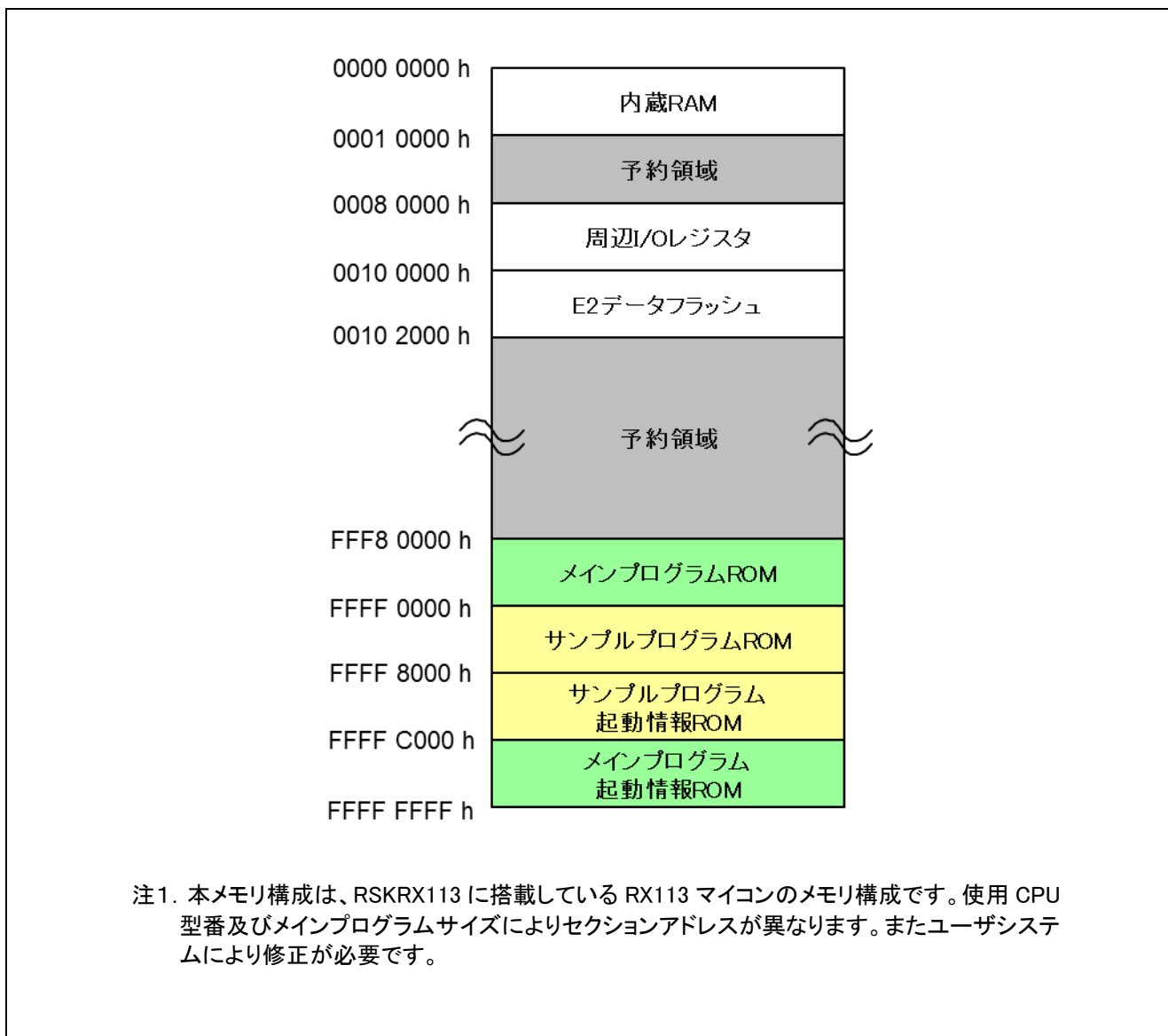


図 5.1 メモリマップ

## 6. メインプログラム仕様

### 6.1 ファイル構成

メインプログラムは、プロジェクトに同梱しています。メインプログラムのソースファイルは src フォルダに入っています。

メインプログラム FIT モジュール名は「r\_flash\_writer\_rx113」です。メインプログラムのソースファイルは src フォルダに入っています。この FIT モジュールのフォルダ構成と、メインプログラムのファイル一覧を以下に示します。

表 6.1にファイル一覧、表 6.2に使用 FIT モジュール一覧を示します。

表6.1 ファイル一覧

フォルダ	ファイル名	内容
src	main.c	メイン処理
	r_rsk_extention_lcd.c	LCD ドライバ応用処理ソースファイル
	r_rsk_flashdriver.c	FLASH 処理ソースファイル
	r_rsk_keydriver.c	Key 入力処理ソースファイル
	r_rsk_keydriver.h	Key 入力処理ヘッダファイル
	r_rsk_lcddriver.c	LCD ドライバ呼び出し処理ソースファイル
	r_rsk_lcddriver.h	LCD ドライバ呼び出し処理ヘッダファイル
	r_rsk_leddriver.c	LED 初期化処理ソースファイル
	r_rsk_leddriver.h	LED 初期化処理ヘッダファイル
	r_usb_hmsc_apl.c	全体管理処理ソースファイル
	r_usb_hmsc_apl.h	USB ドライバ呼び出し処理ヘッダファイル

表6.2 使用 FIT モジュール一覧

フォルダ	内容
r_bsp	ボードサポートパッケージ(BSP) ファイル群
r_config	各 FIT モジュールの config ファイル
r_flash_rx	フラッシュメモリ(Flash API) ファイル群
r_lcdc_rx	LCD コントローラ/ドライバ(LCDC) ファイル群
r_tfat_driver_rx	M3S-TFAT-Tiny メモリドライバインタフェース ファイル群
r_tfat_rx	M3S-TFAT-Tiny FAT ファイルシステム(TFAT) ファイル群
r_usb_basic_mini	USB Basic Firmware ファイル群
r_usb_hmsc_mini	USB ホストマスタストレージクラス(USB HMSC) ファイル群

## 6.2 モジュール一覧

表 6.3にモジュール一覧を示します。

表6.3 モジュール一覧

ファイル名	モジュール名	内容
main.c	main	メインプログラムのメイン処理。 FLASH ドライバの open 処理と usb_main 関数の呼び出し。
r_usb_hmsc_api.c	usb_main	各種イニシャル処理と全体制御本体。
	usb_hmsc_driver	HMSC ドライバタスク処理。
	msc_detach_device	USB 抜去処理。
	msc_connect_wait	USB デバイス挿入待ち処理。
	msc_drive	USB デバイス接続&TFAT ファイルシステムマウント処理。
	msc_data_ready	リードステート前のダミーステート処理。
	msc_data_read	USB メモリからのデータリード & FLASH へのデータライト処理。
	usb_hsmpl_device_state	USB ドライバコールバック処理。
	msc_configured	USB デバイス挿入通知処理。
	msc_drive_complete	USB 接続完了通知処理。
	msc_detach	USB デバイス抜去通知処置。
	usb_mcu_init	USB ポート初期化処理。
	usb_board_init	(LED&LCD) 初期化処理 & USB 割り込みイネーブル処理。
	usb_driver_init	USB ドライバ初期化処理。
	apl_init	管理用テーブル初期化処理。
	msc_registration	USB コールバック関数登録処理。
	msc_event_set	イベントセット処理。
	msc_event_get	スイッチ入力検知 & イベントゲット処理。
msc_led_control	LED コントロール処理。	
r_rsk_leddriver.c	usb_cpu_LedInitial	LED 初期化処理。
	usb_cpu_led_set_bit	LED 個別制御処理。
	usb_cpu_led_set_data	LED 一斉制御処理。
r_rsk_extention_lcd.c	Init_Extension_LCD	LCD ポート初期化処理。
	usb_lcd_print_14seg_string	14SEG 点灯処理。
	usb_lcd_print_u7seg_digital	7SEG 表示処理 (LCD 上部部分)。
	usb_lcd_print_c7seg_digital	7SEG 表示処理 (LCD 中央部分)。
	dfw_lcd_ascii_calc	Ascii 文字変換処理。
r_rsk_lcdriver.c	usb_cpu_LcdInitial	LCD モジュール初期化処理。
	usb_cpu_LcdDisp	LCD 表示処理。
	string_length_count	文字列数通知処理。
r_rsk_flashdriver.c	SAMPLE_FLASH_Write	FLASH イレーズ & 書き込み処理。
r_rsk_keydriver.c	usb_cpu_key_read	チャタリング除去処理。
	usb_cpu_sw_data	スイッチ入力検知処理。
	usb_cpu_sw1_data	スイッチ 1 入力。
	usb_cpu_sw2_data	スイッチ 2 入力。
	usb_cpu_sw3_data	スイッチ 3 入力。

## 6.3 フローチャート

### (1) メイン処理

ボードサポートパッケージ (BSP モジュール) のスタートアップルーチンから最初に呼ばれるメイン関数です。

FLASH のオープン処理を行い、usb\_main 関数を呼び出します。

usb\_main 関数内で無限ループ処理を行っていますが、本関数の最後でも無限ループします。

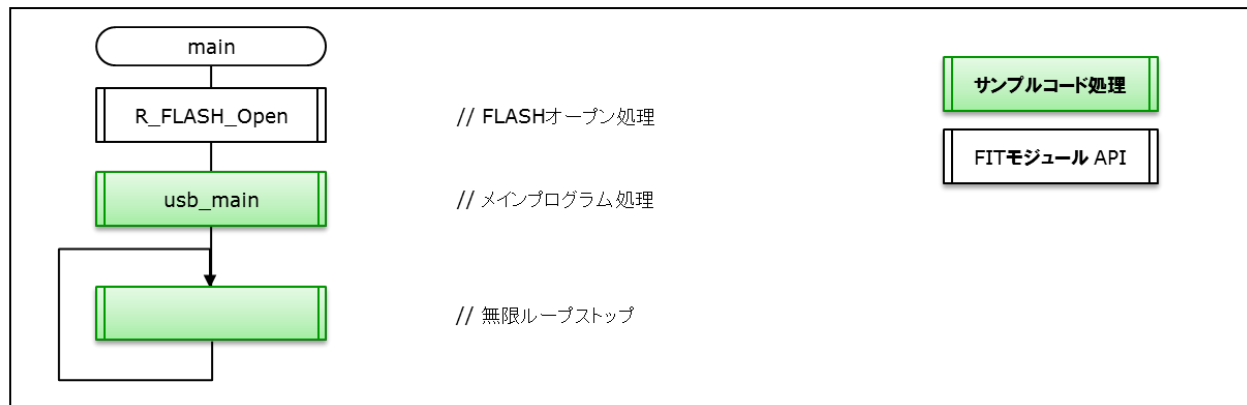


図6.1 メイン処理(1)

### (2) USB メイン処理

各種初期化、処理順序のステート管理を行います。

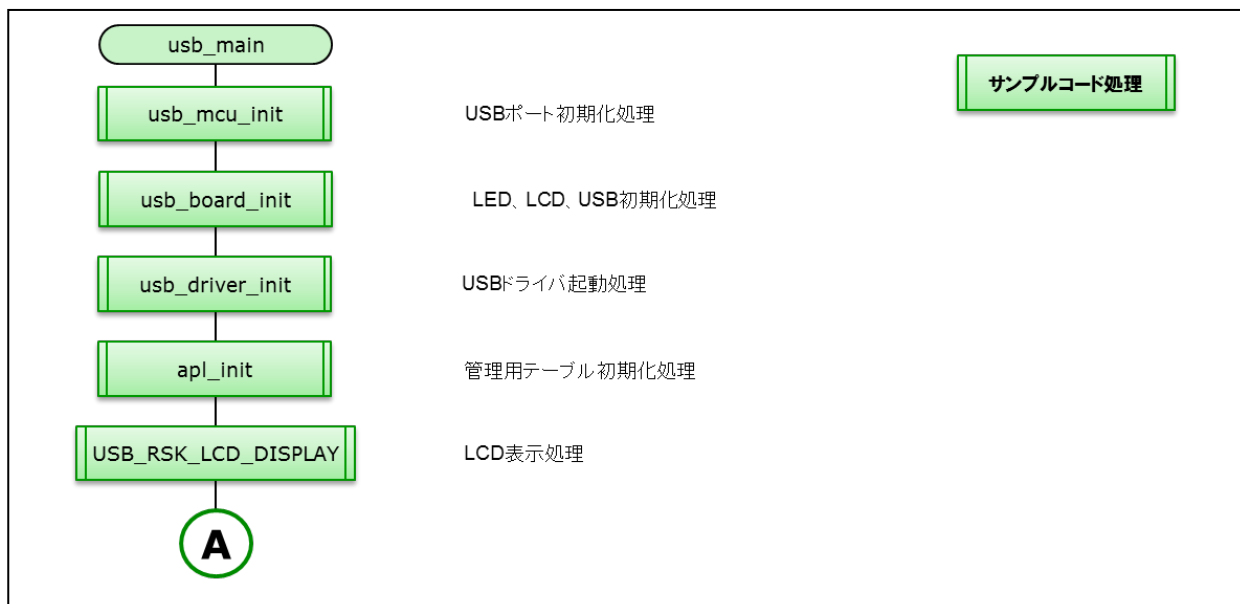


図 6.2 USB メイン処理 1



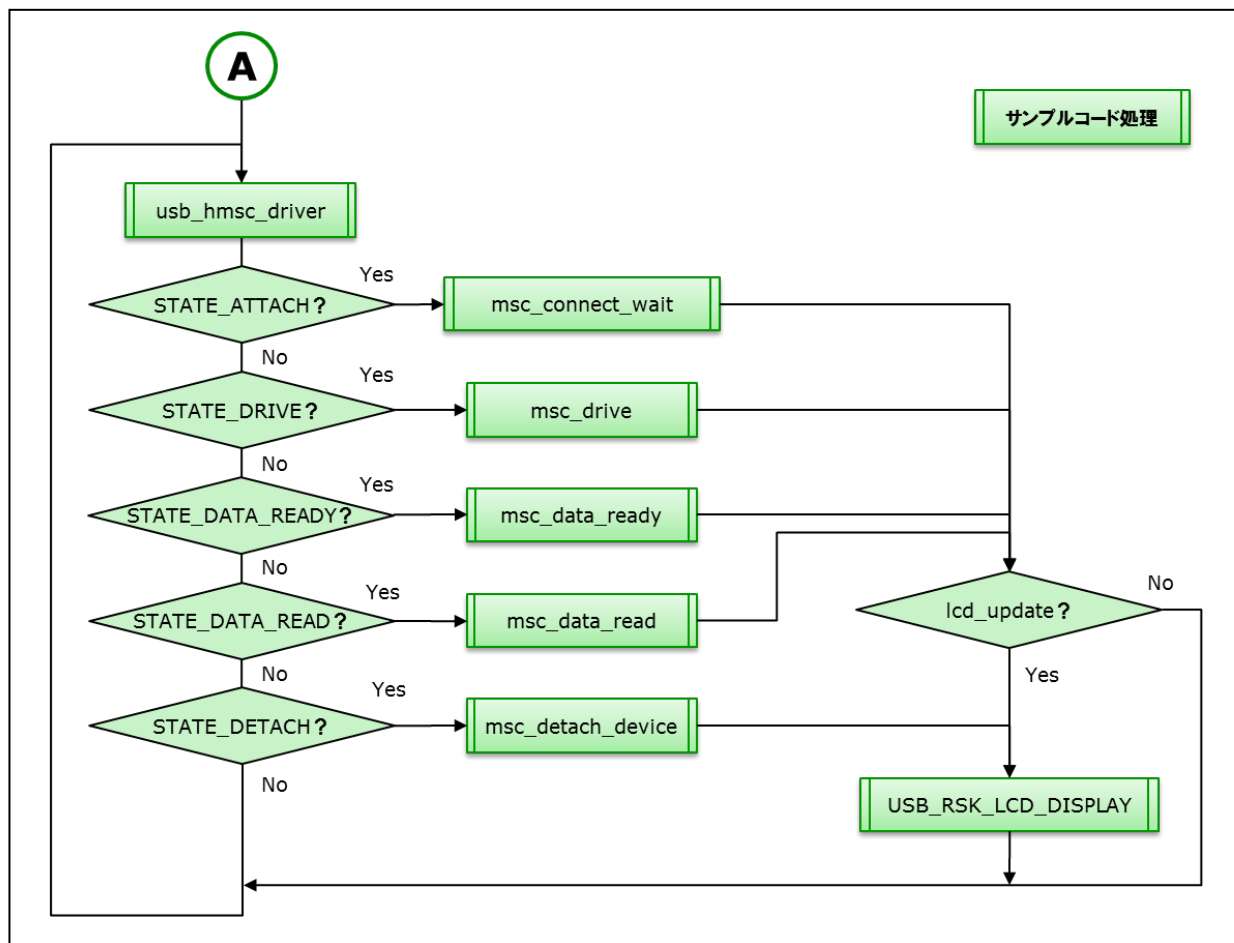


図 6.3 USB メイン処理 2

(3) USB ドライバポート初期化

USB ドライバの初期ポート設定です。



図 6.4 USB ドライバポート初期化処理

(4) LCD、LED、USB ドライバ初期化

LCD、LED、USB ドライバの初期化処理をします。



図 6.5 LCD、LED、USB ドライバ初期化処理

(5) USB ドライバ起動

USB ドライバの起動処理です。

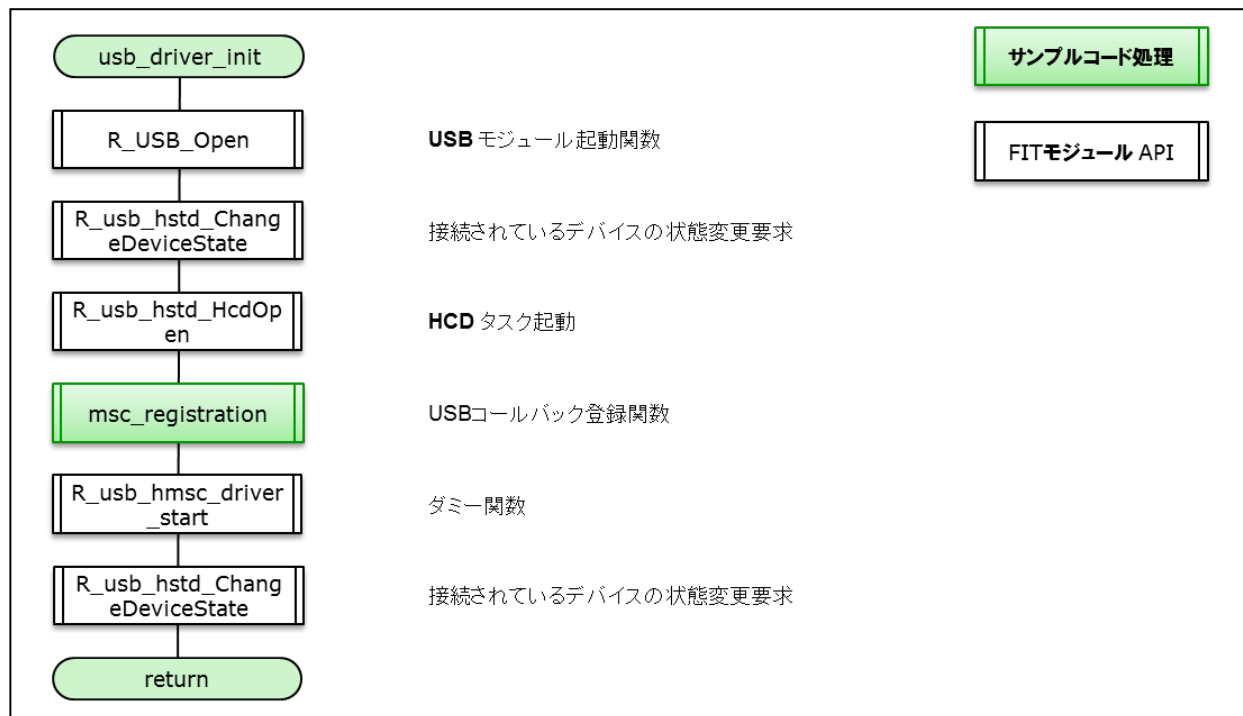


図 6.6 USB ドライバ起動処理

(6) 管理テーブル初期化

管理用テーブル初期化を行います。



図 6.7 USB ドライバ初期化処理

(7) クラスドライバ登録

クラスドライバの登録処理です。

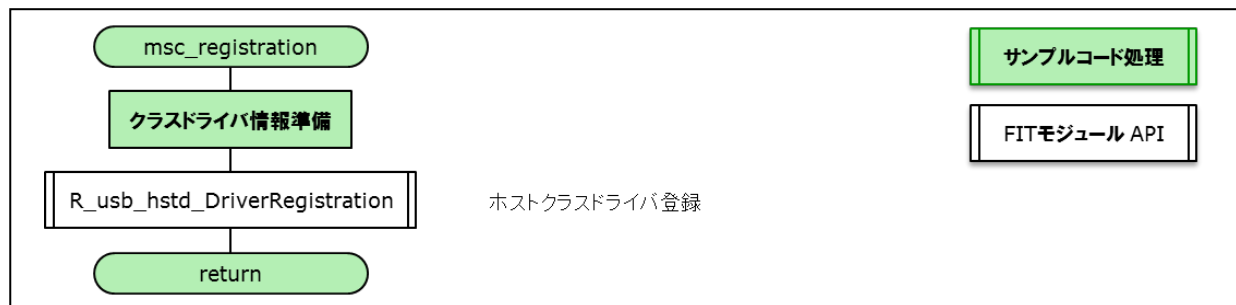


図 6.8 USB コールバック関数登録処理

(8) USB HMSC ドライバタスク

USB ドライバタスク処理です。

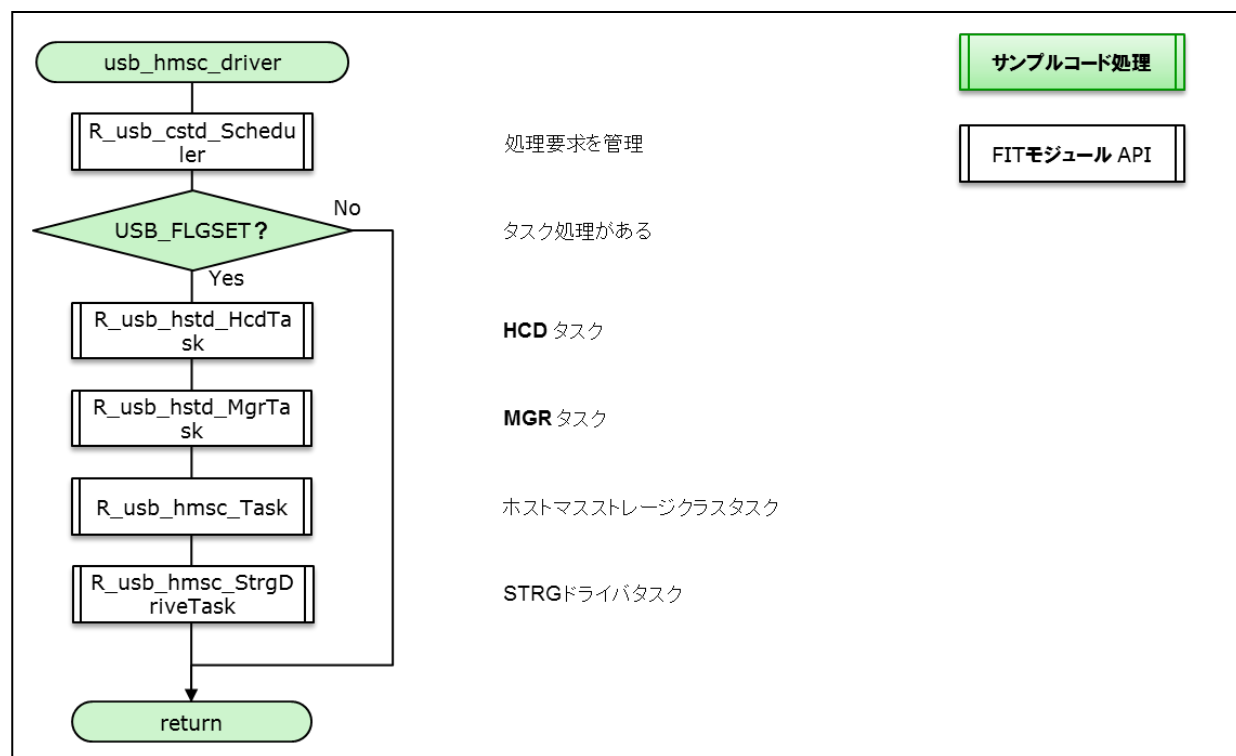


図 6.9 USB HMSC ドライバタスク処理

(9) USB デバイス検出待ち

USB デバイス挿入待ち関数です。

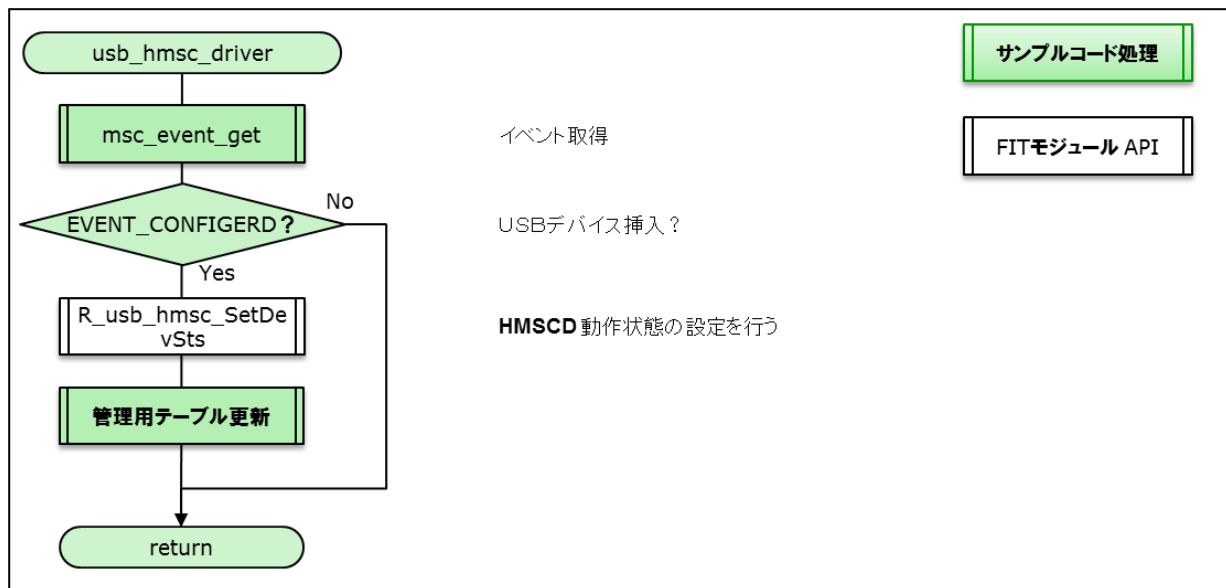


図 6.10 USB デバイス検出待ち処理

(10) USB デバイス接続、TFAT ファイルシステムマウント

USB ドライブ情報を読み込み、TFAT ファイルシステムのマウント処理です。

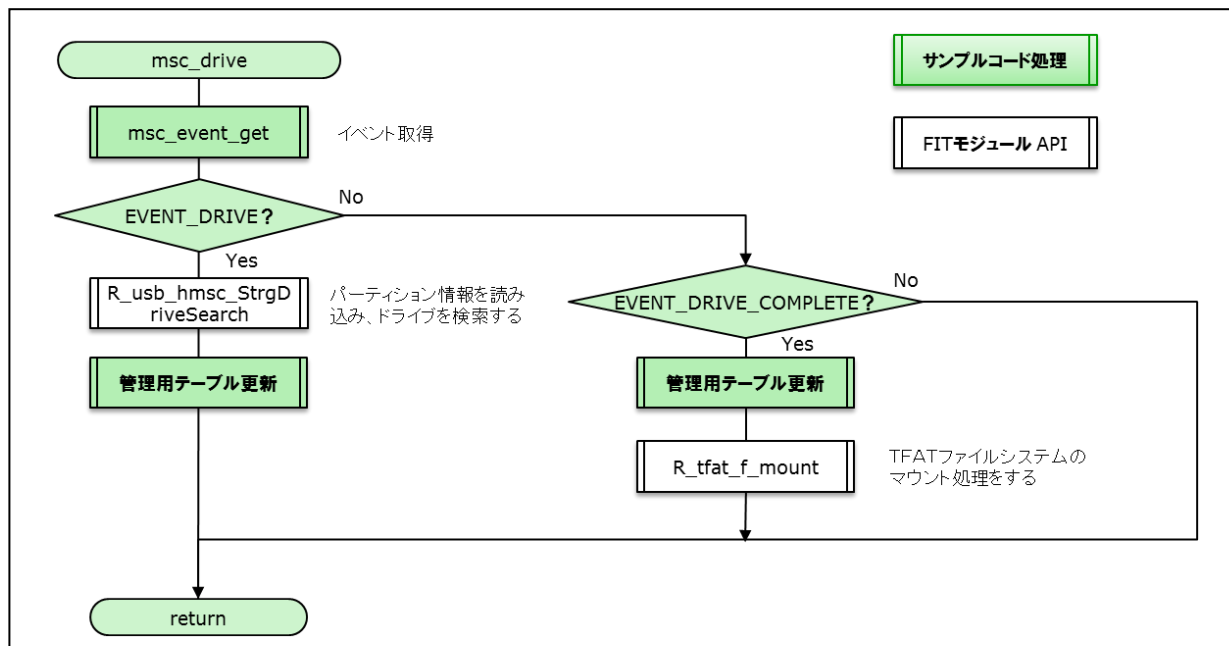


図 6.11 USB デバイス接続、TFAT ファイルシステムマウント処理

(11) リードステータス前の設定

ダミー管理処理です。



図 6.12 リードステータス前の設定処理

(12) USBメモリからのデータリード、フラッシュメモリの書き換え

USBメモリからデータを読み込み、内蔵フラッシュメモリに書き込み処理を行います。

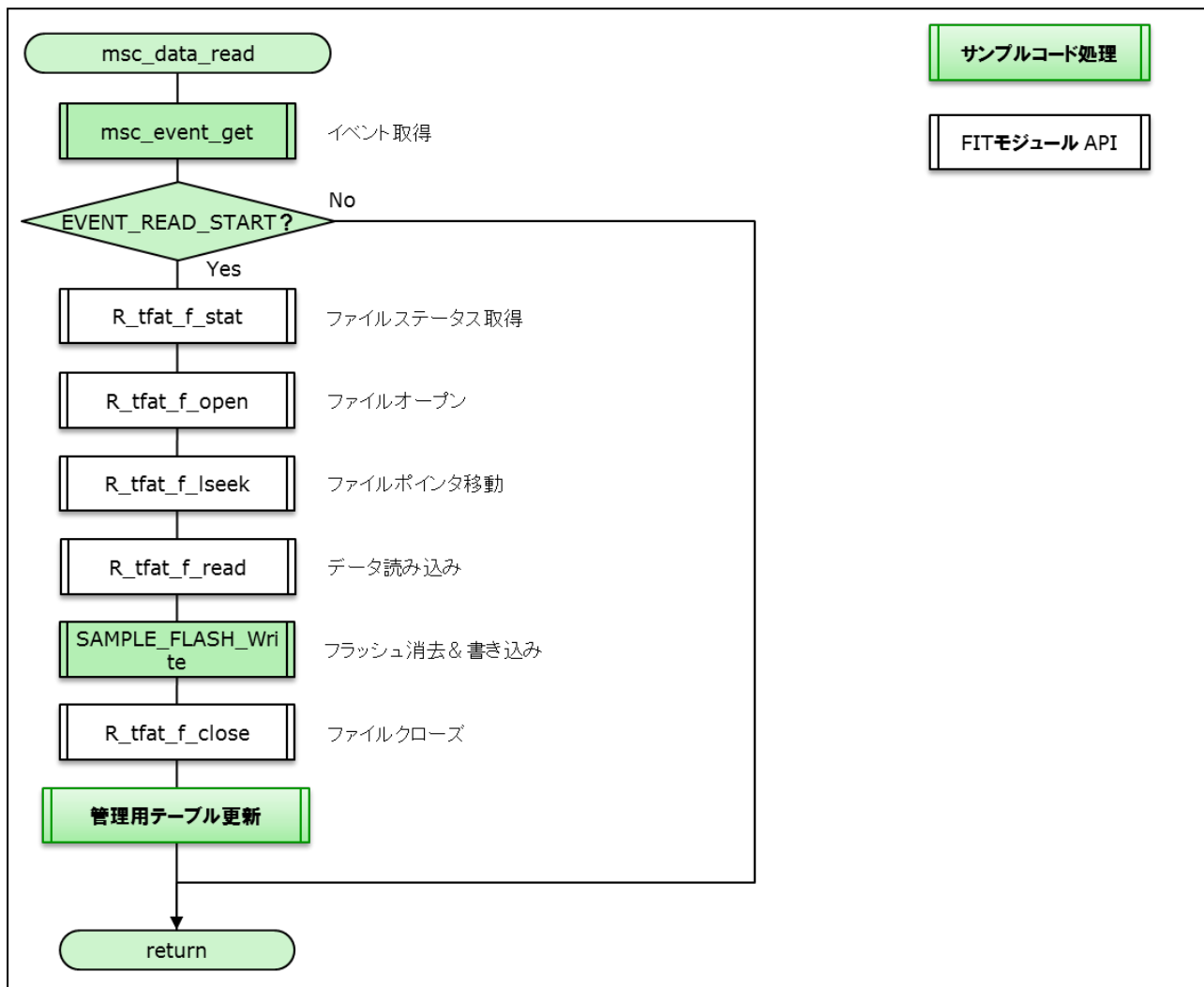


図 6.13 USBメモリからのデータリード、フラッシュメモリの書き換え処理(1)

(13) USB 切断

USB デバイスを切断し、サンプルプログラムを起動します。

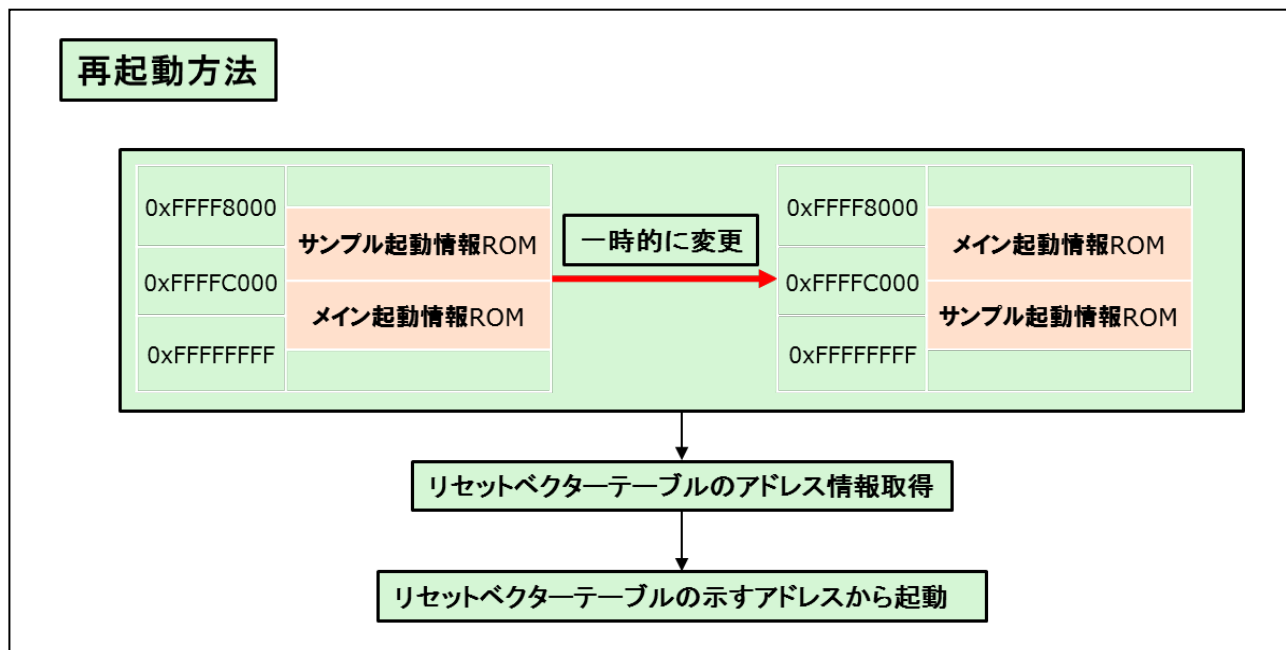


図 6.14 USB 切断処理



## 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/>

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
1.01	2016.02.29	—	新規発行
1.02	2017.03.31	24	4.3 プロジェクトのデバッグ 4, “「Renesas GDB Hardware Debugging」の 「r_flash_writer_rx113.x」をクリックします。” 元は、 “「Renesas GDB Hardware Debugging」の 「r_flash_writer_rx113.x」をクリックし、「デバッグ(D)」をク リックします。”であった。 上記に合わせて、図を更新した。
		—	プロジェクト内の doc フォルダ内のドキュメントを更新した。

## 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

- 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か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。
  - 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。
  - 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。
  - 当社製品を、全部または一部を問わず、改造、改変、複製、その他の不適切に使用しないでください。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。
  - 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、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、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損害が生じても、当社は一切その責任を負いません。
  - 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。
  - 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。
  - 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。
  - 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、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。  
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。
  - お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
  - 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。
  - 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。
- 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。

(Rev.3.0-1 2016.11)



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どうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>