RX231 グループ
静電容量式タッチセンサユニットと 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

R01AN2561JJ0102
Rev.1.02
2016.02.29

要旨

本書は、RX231 内蔵の静電容量式タッチセンサユニット（CTSU）と 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 RX231 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て使用します。

動作確認デバイス

RX231 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)
- e² 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 e2 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 21
4.1 プロジェクトのビルド 21
4.2 デバッグの準備 22
4.2.1 機器の構成 22
4.2.2 RSK の設定 23
4.2.3 USB メモリの準備 24
4.3 プロジェクトのデバッグ 25
4.4 サンプルプログラムの動作 29

5. アプリケーションの概要 32
5.1 メモリ構成 33

6. メインプログラム仕様 34
6.1 ファイル構成 34
6.2 モジュール一覧 35
6.3 フローチャート 36

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、静電容量式タッチセンサユニット（CTSU）、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 RX231（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	RX231 グループ
使用ボード	Renesas Starter Kit for RX231 （製品型名: R0K505231S000BE）
統合開発環境	ルネサスエレクトロニクス製 e² 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。

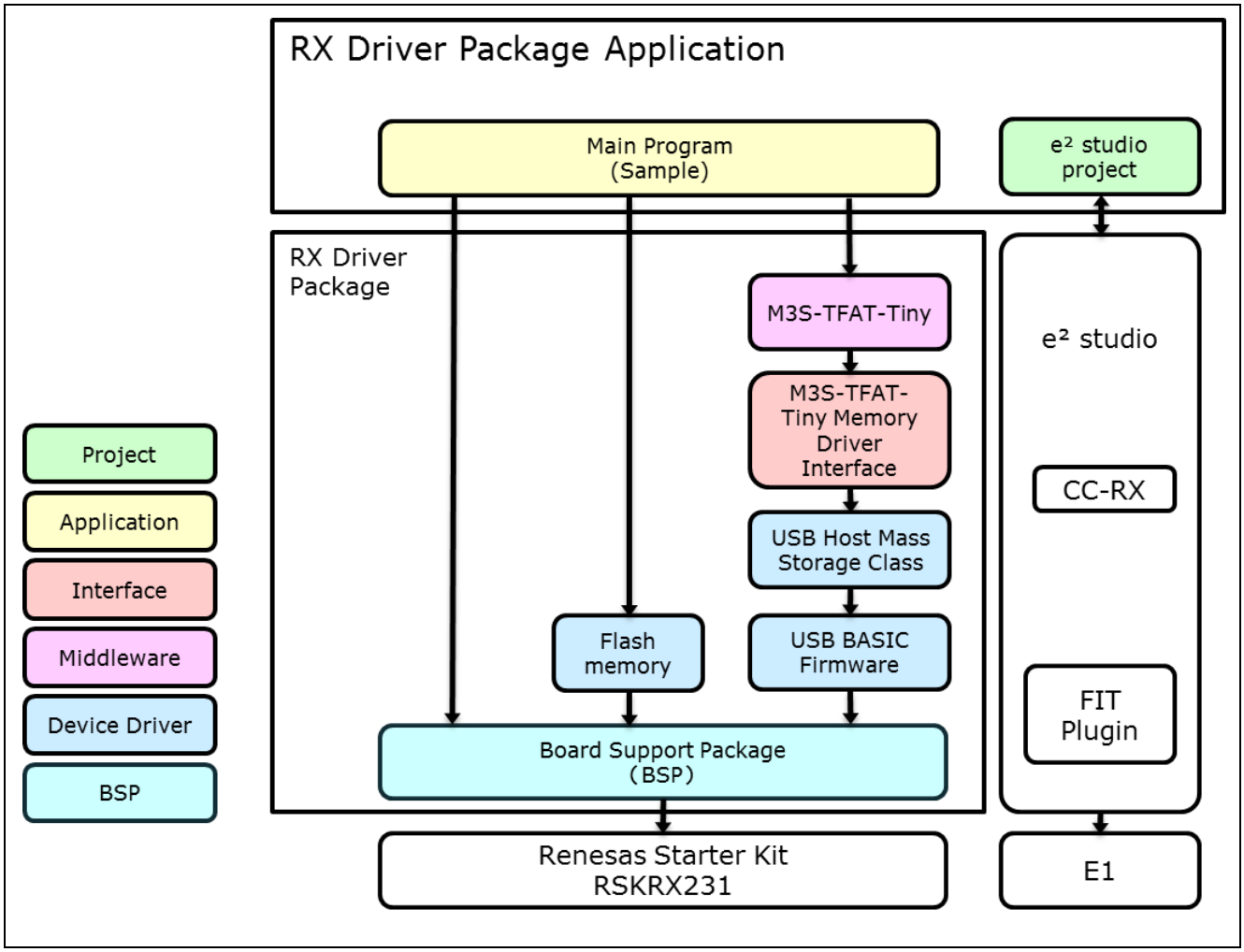


図 1.1 モジュールの構成

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
	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_rx231	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。

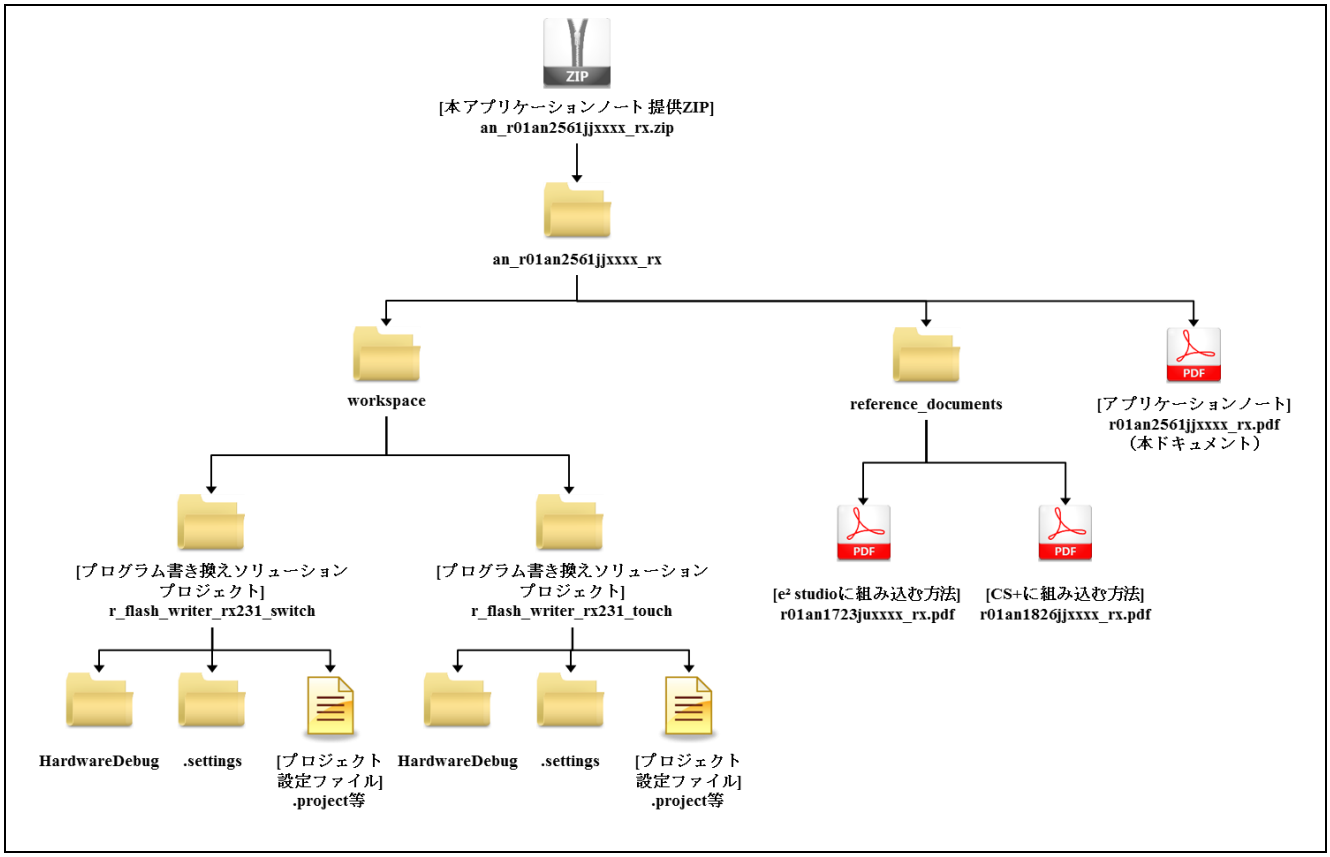


図 1.2　ファイル構成

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。e² 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。「e² 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 e² 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 e² studio の入手とインストール方法

以下の URL にアクセスし、e2 studio をダウンロードしてください。

http://japan.renesas.com/e2studio_download

なお、本ドキュメントは e2 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、e² 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 e2 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

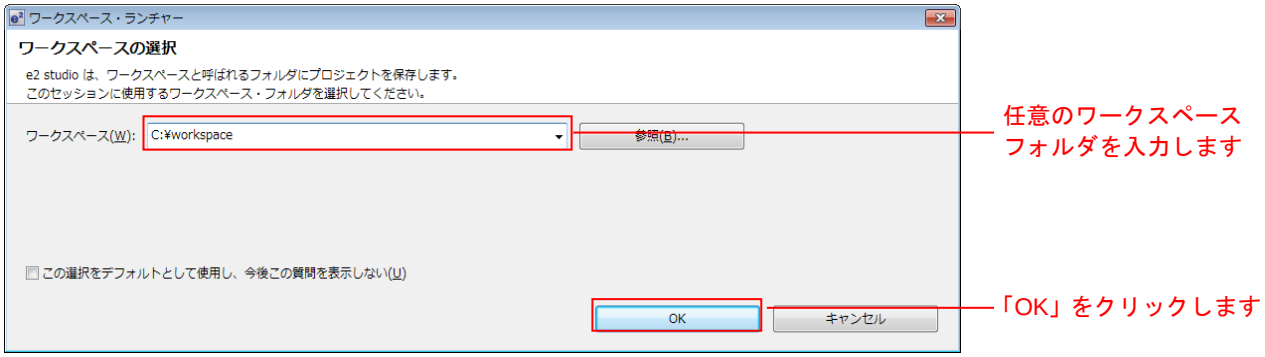
3.　プロジェクトの構築

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・ブラウザー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。

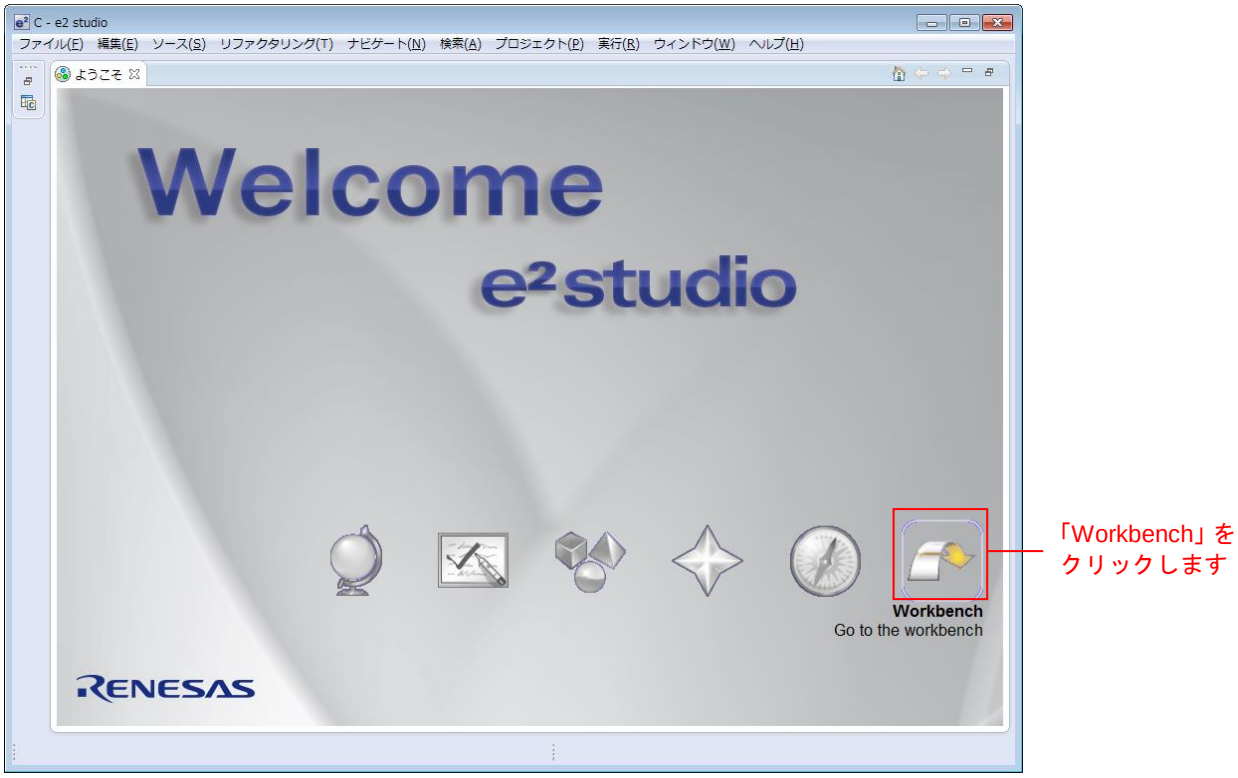
3.1　ワークスペースの作成

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。

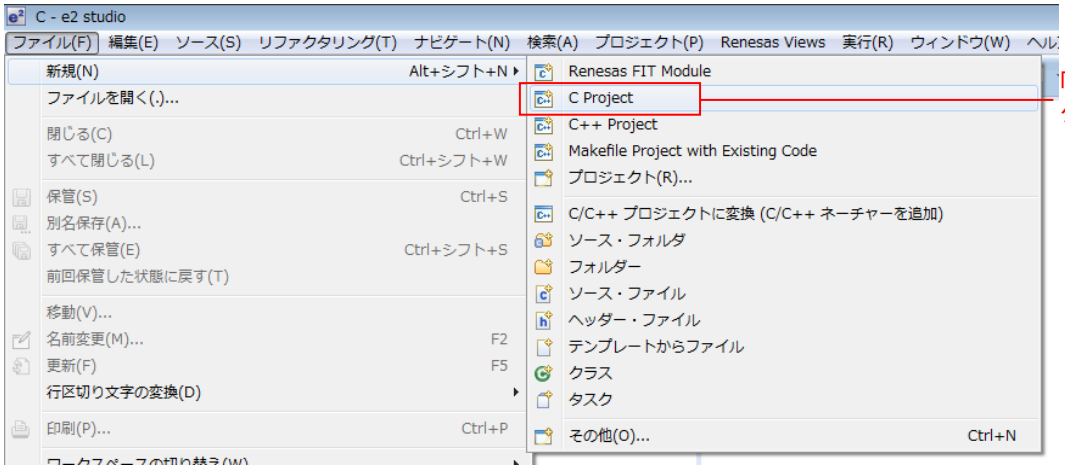


3.2　プロジェクトの作成

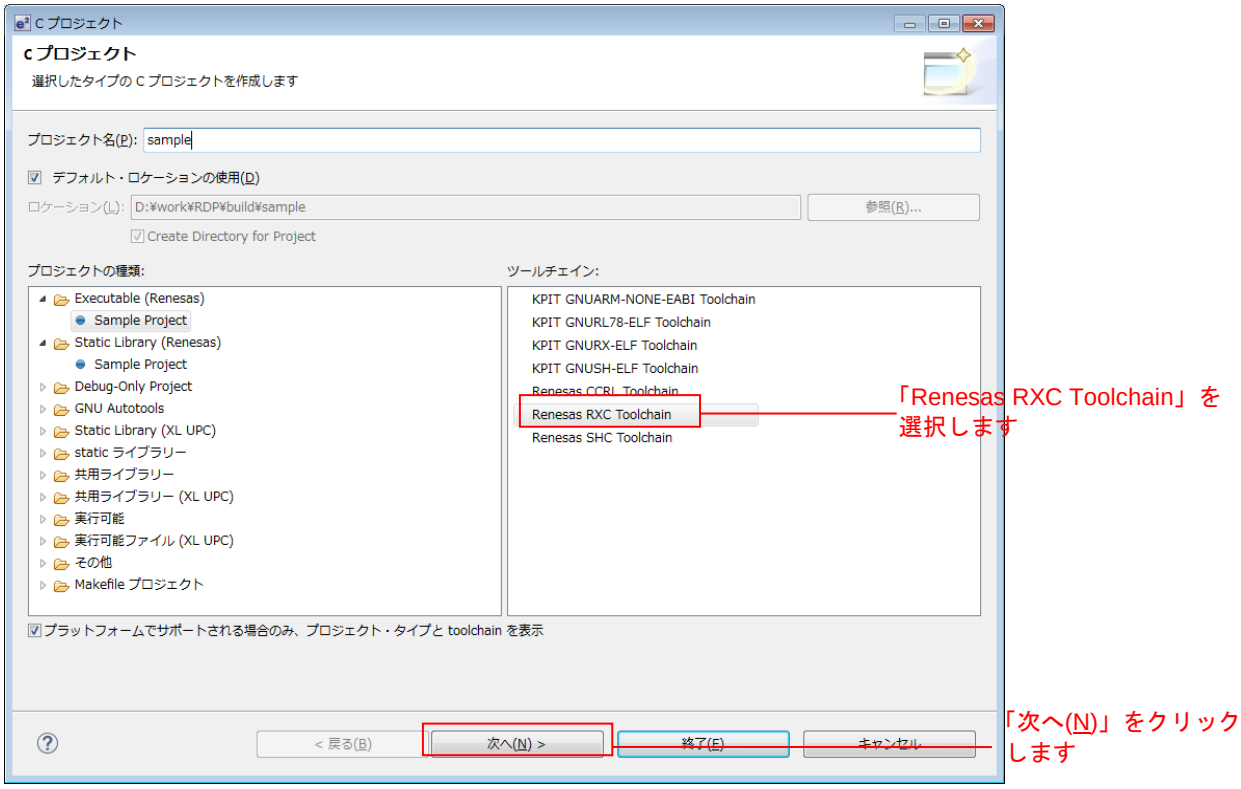
スマート・ブラウザーの機能を使用するときは、対象となるプロジェクトあるいはファイルを選択しておく必要があります。スマート・ブラウザーの機能を使用するために、対象デバイスを指定したプロジェクトを作成します（注１）。

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。

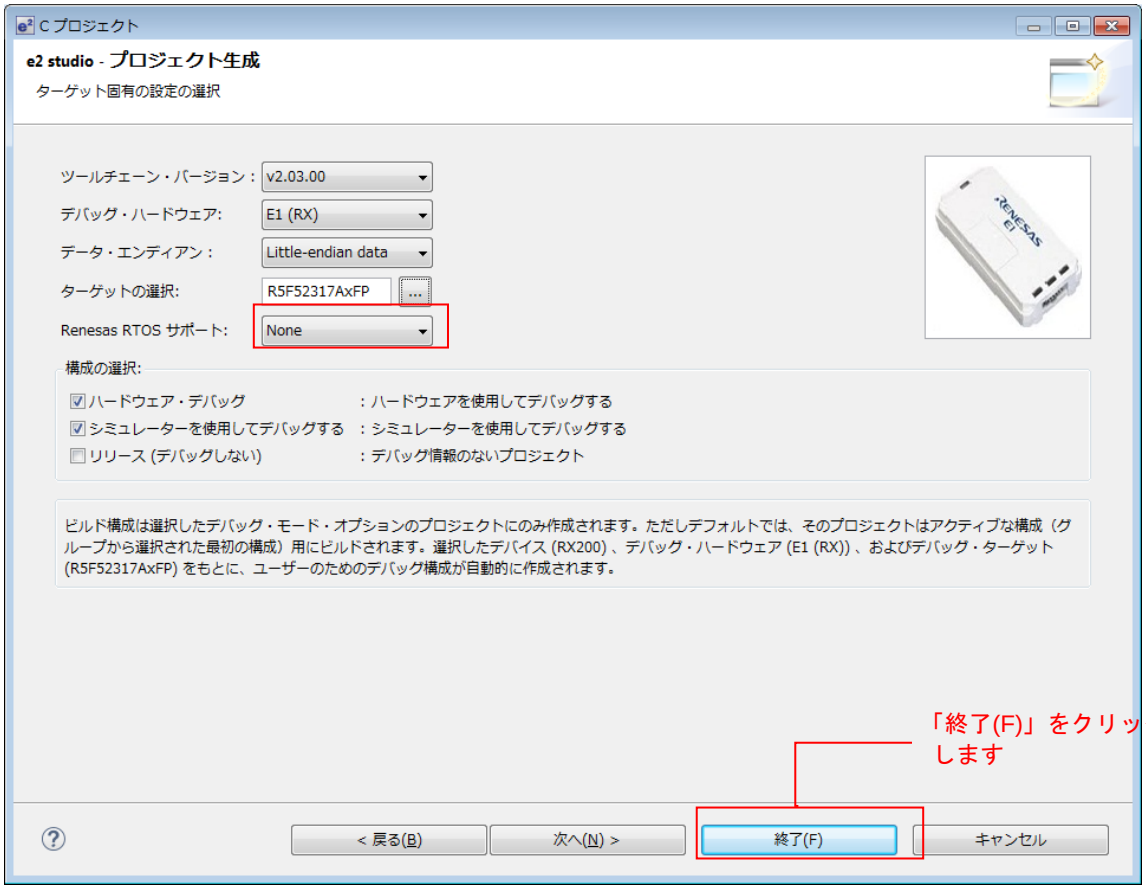
1. [ファイル(F)] → [新規(N)] → [C Project]の順にクリックして新しい C プロジェクトを作成します。新規プロジェクト作成ウィザードを起動します。



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。RX231 100 ピンデバイスの「R5F52317AxFP」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。

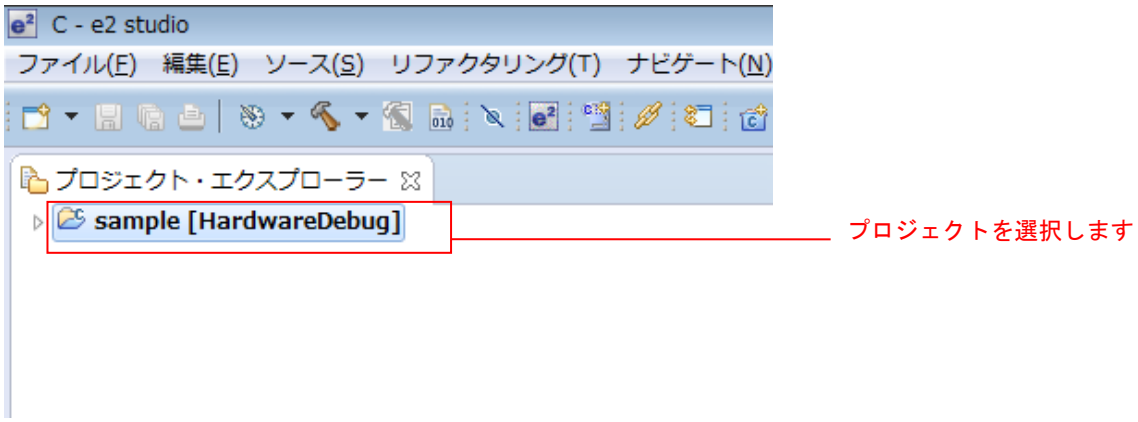
3.3　プロジェクトのインポート

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。

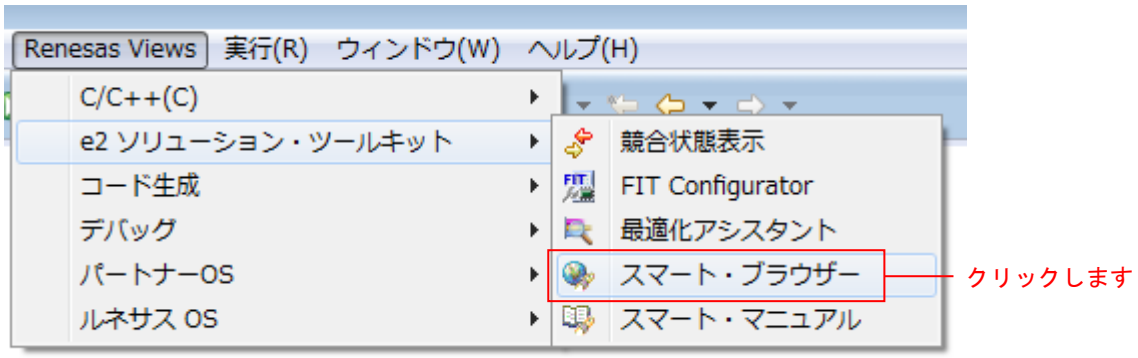
本アプリケーションノートには、以下に示すプロジェクトが含まれます。

- ファイル選択を静電容量式タッチセンサユニット（CTSU）で行うプロジェクト
- ファイル選択をスイッチで行うプロジェクト

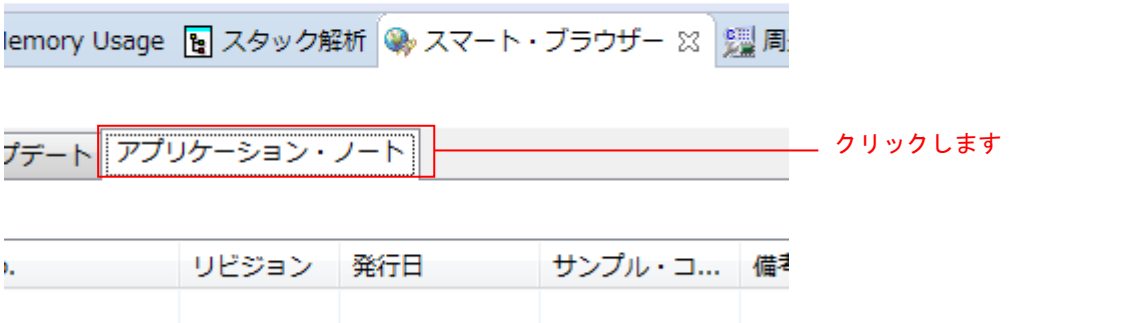
1. プロジェクト・エクスプローラーから「3.2 プロジェクトの作成」で作成したプロジェクトを選択します。



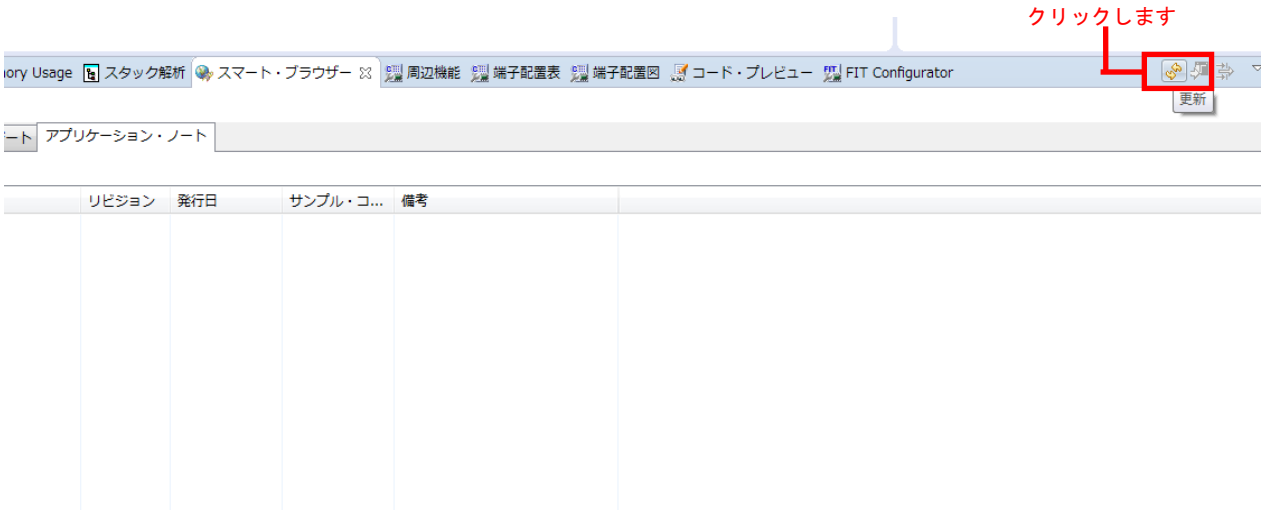
2. [Renesas Views]→[e2 ソリューション・ツールキット]→[スマート・ブラウザー]の順にクリックし、スマート・ブラウザーを起動します。



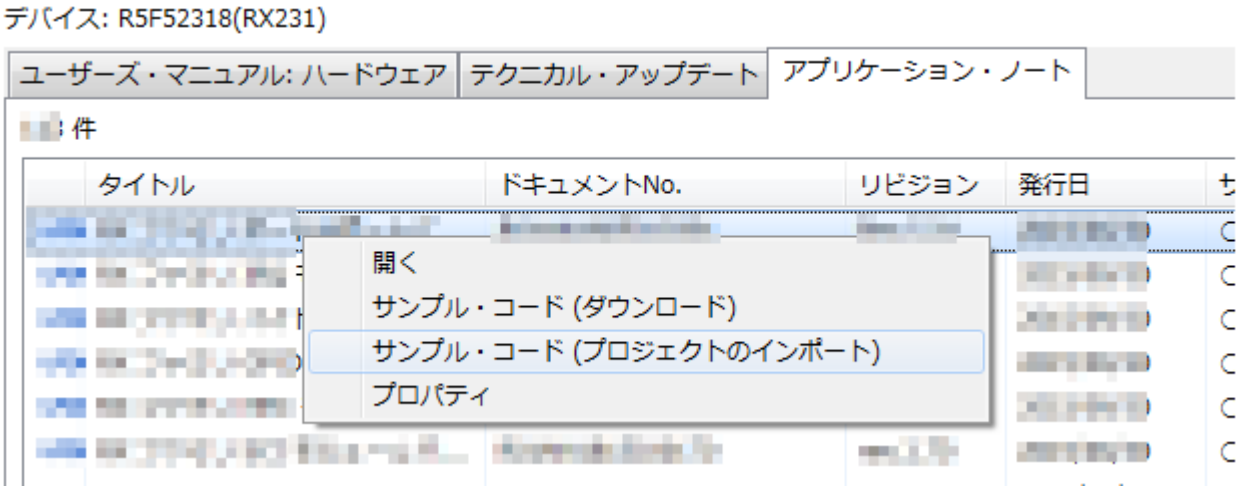
3. [スマート・ブラウザー]タブの[アプリケーション・ノート]タブをクリックします。



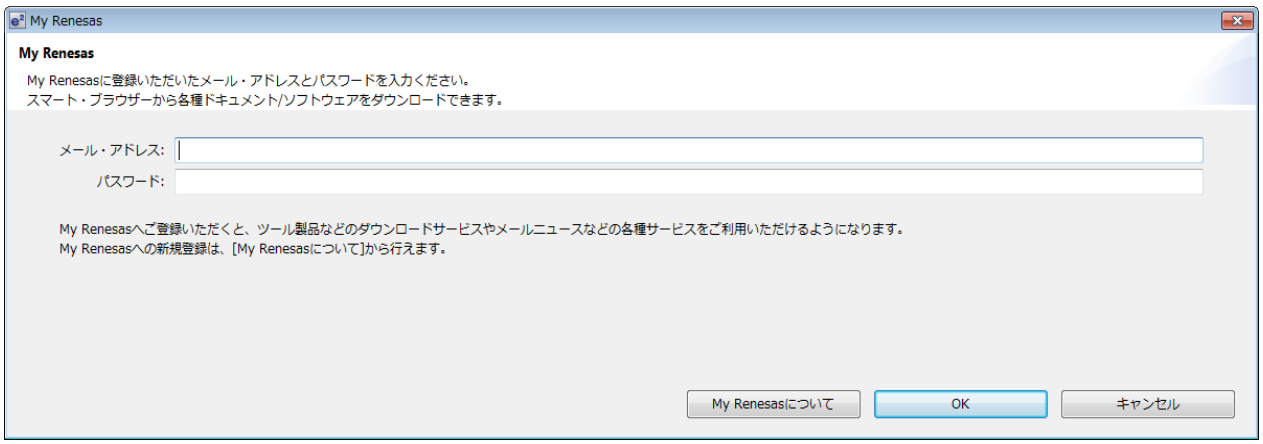
4. [更新] をクリックします。



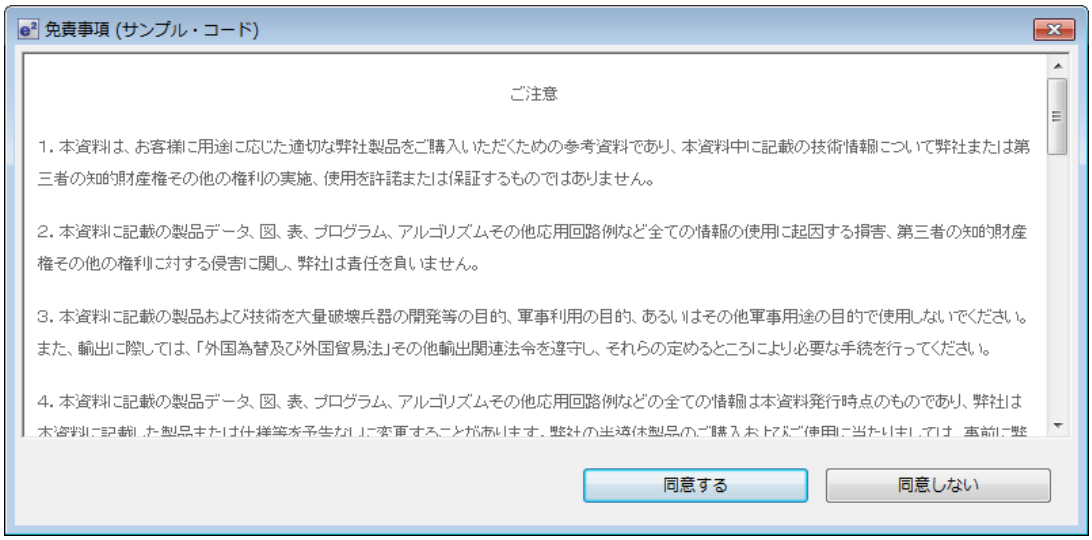
5. 本アプリケーションノートを選択し、右クリックします。コンテキストメニューの[サンプル・コード(プロジェクトのインポート)]をクリックします。



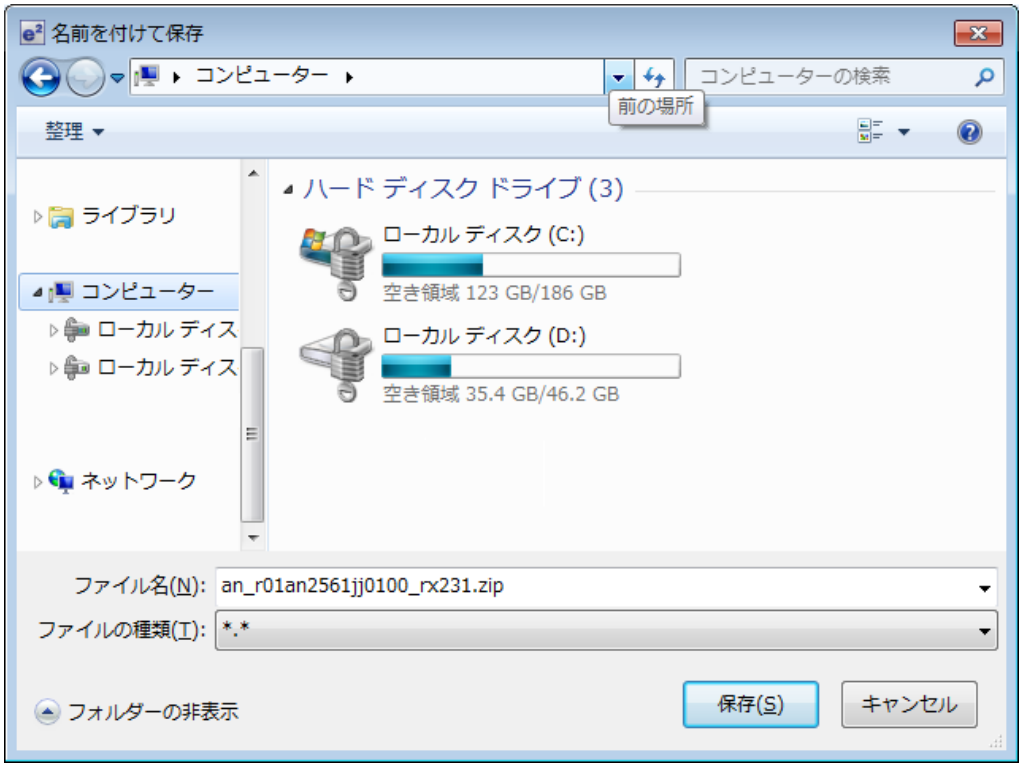
※ 一度も My Renesas による認証をしていない場合、ファイルをダウンロードする際に「My Renesas」ダイアログがオープンします。ルネサス Web サイトで登録しているメール・アドレスとパスワードを入力してください。



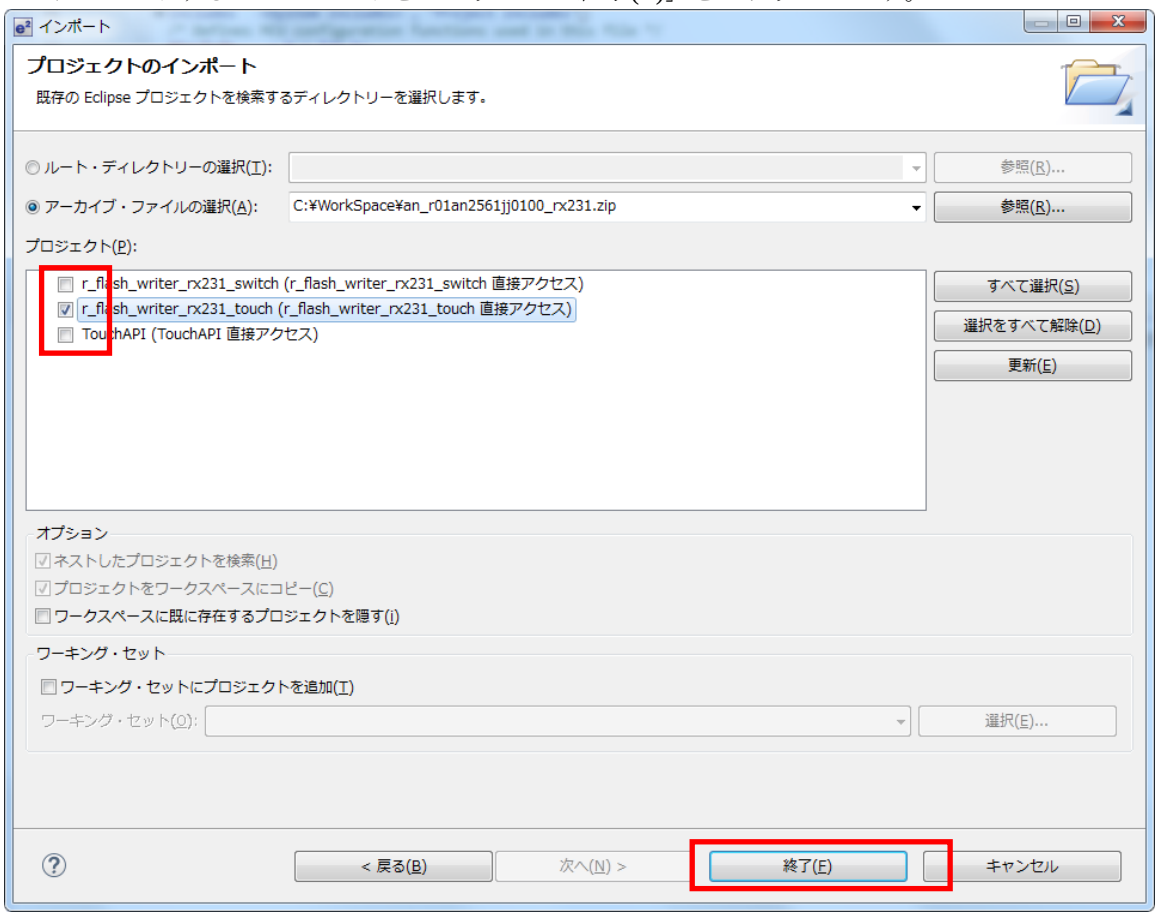
6. [同意する]をクリックします。



7. 本アプリケーションノートを保存します。



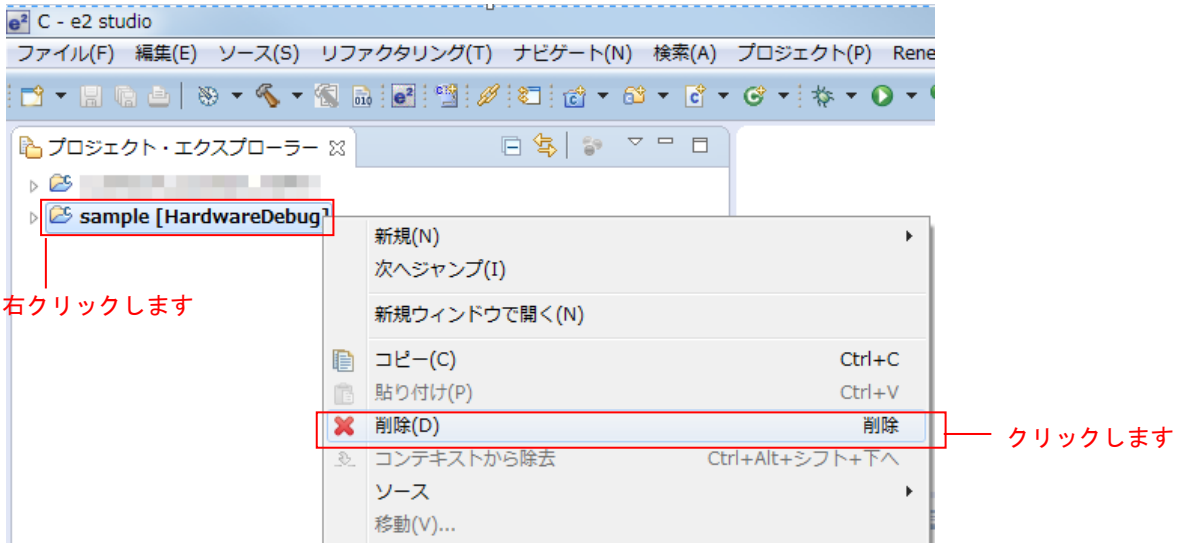
8. インポートするプロジェクトをチェックし「終了(E)」をクリックします。



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_rx231_switch	スイッチでファイル選択を行うプロジェクト
r_flash_writer_rx231_touch	タッチでファイル選択を行うプロジェクト
TouchAPI	Workbench6 を使用してタッチセンサユニットを調整するためのプロジェクトです。今回のサンプルサンプルプロジェクトでは使用しません。

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。

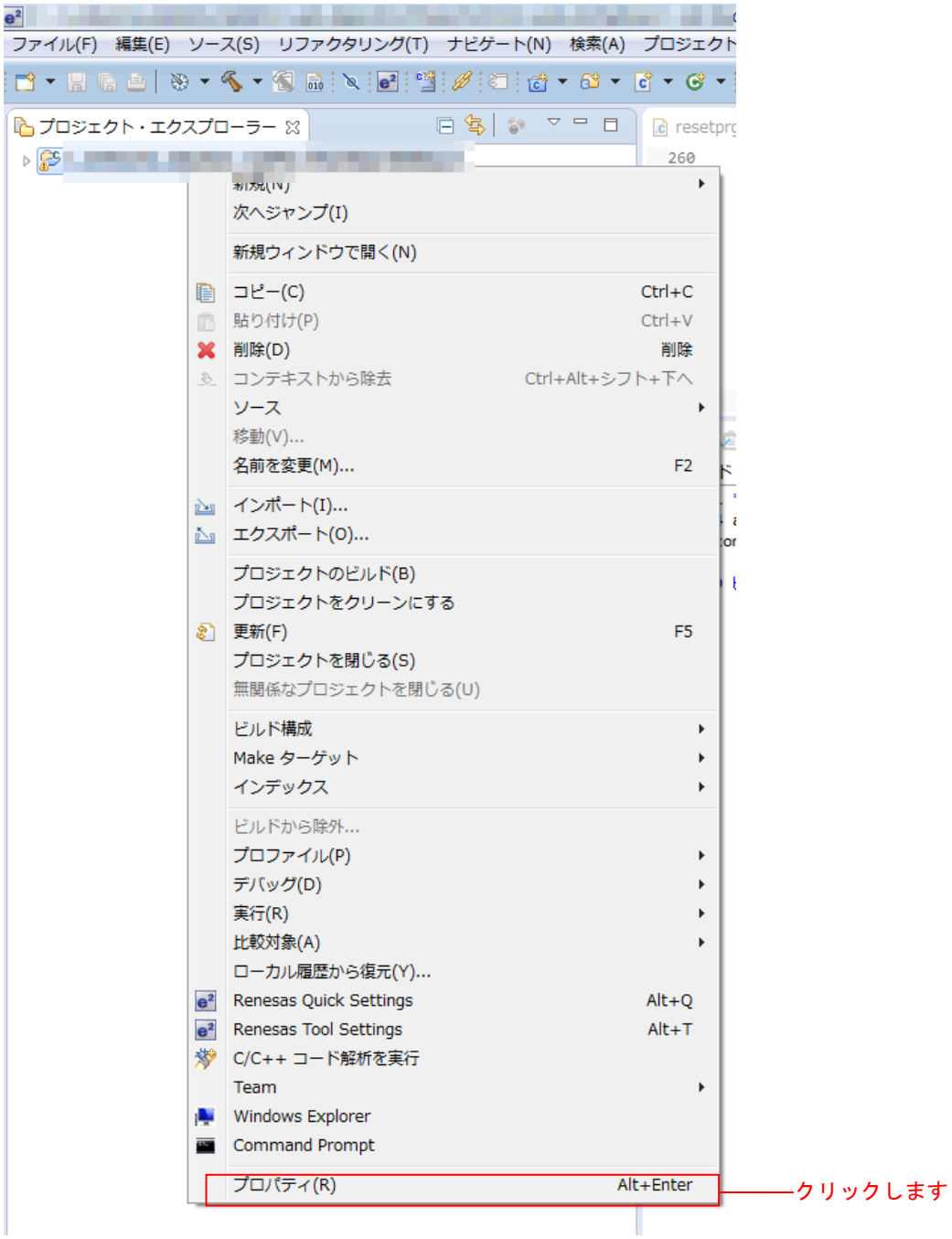
【r_config/r_usb_hmsc_mini_config.h】

```
    #define USB_TFAT_USE_PP
```

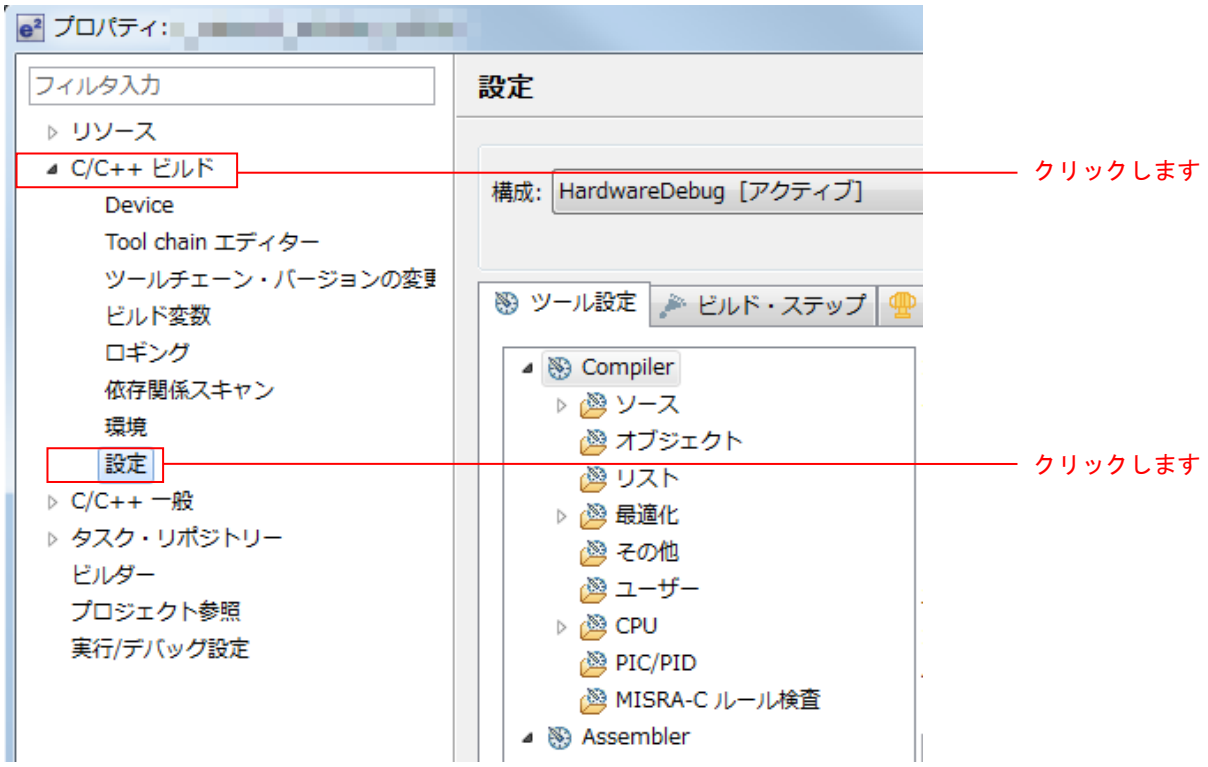
3.4.2　プロジェクト設定の変更

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. e² 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	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）
	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

sample1、sample2 のビルド時のデフォルト設定からの変更点を表 3.3に、sample3 の変更点を表 3.4　変更したビルド設定(sample3)表 3.4に示します。

表3.3　変更したビルド設定(sample1、sample2)

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。
	EXCEPTVECT セクションのアドレスを“0xFFFF BF80”に変更する。	スタートアッププログラム保護機能を使用した後、“0xFFFF FF80”になるアドレスを設定します。
	RESETVECT セクションのアドレスを“0xFFFF BFFC”に変更する。	スタートアッププログラム保護機能を使用した後、“0xFFFF FFFC”になるアドレスを設定します。
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。

表3.4　変更したビルド設定(sample3)

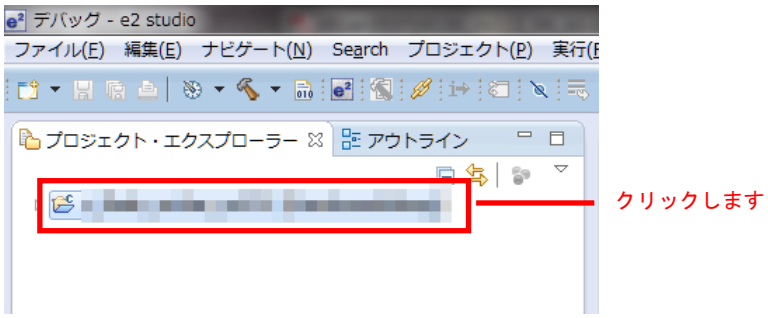
項目	変更内容	説明
Linker -セクション	PResetPRG、PIntPRG を セクション定義から除去する（注）	BSP を使用する際には設定が必要です。 （FIT を使用する際には必須）
	P セクションを P*セクションに変更する（注）	BSP を使用する際には設定が必要です。 （FIT を使用する際には必須）
	C_1 セクションのアドレスを “0xFFFF 0000”に変更	書き込む先頭のアドレスを指定します。
	_MDEREG セクション（アドレスを“0xFFFF BF80”）を追加	スタートアッププログラム保護機能を使用した後、“0xFFFF FF80”になるアドレスを設定します。併せてソースコードを以下のように変更します。 Source¥HwResource¥r_cg_vecttbl.c 変更前 #pragma address id_code=0xffffffa0 変更後 #pragma section C ID_CODE
	OFS1_LOCATION セクション（アドレスを“0xFFFF BF88”）を追加	スタートアッププログラム保護機能を使用した後、“0xFFFF FF88”になるアドレスを設定します。併せてソースコードを以下のように変更します。 Source¥HwResource¥r_cg_vecttbl.c 変更前 #pragma address id_code=0xffffffa0 変更後 #pragma section C ID_CODE
	OFS0_LOCATION セクション（アドレスを“0xFFFF BF8C”）を追加	スタートアッププログラム保護機能を使用した後、“0xFFFF FF8C”になるアドレスを設定します。併せてソースコードを以下のように変更します。 Source¥HwResource¥r_cg_vecttbl.c 変更前 #pragma address id_code=0xffffffa0 変更後 #pragma section C ID_CODE
	ID_CODE セクション（アドレスを“0xFFFF BFA0”）を追加	スタートアッププログラム保護機能を使用した後、“0xFFFF BFA0”になるアドレスを設定します。併せてソースコードを以下のように変更します。 Source¥HwResource¥r_cg_vecttbl.c 変更前 #pragma address id_code=0xffffffa0 変更後 #pragma section C ID_CODE
	FIXEDVECT セクション（アドレスを“0xFFFF BFD0”）を追加	スタートアッププログラム保護機能を使用した後、“0xFFFF BFD0”になるアドレスを設定します。
Linker -出力	出力ファイル・タイプを“Binary via absolute”に変更	USB メモリに書くファイルの形式を設定します

4. 動作確認

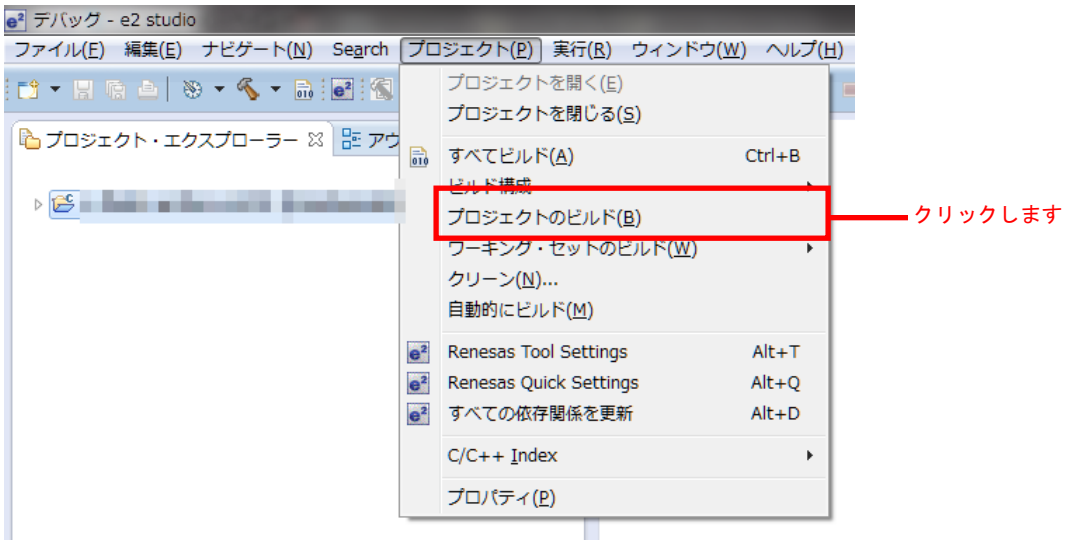
4.1 プロジェクトのビルド

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 RX231 に同梱
USB メモリ	FAT、または FAT32 でフォーマットしたもの。

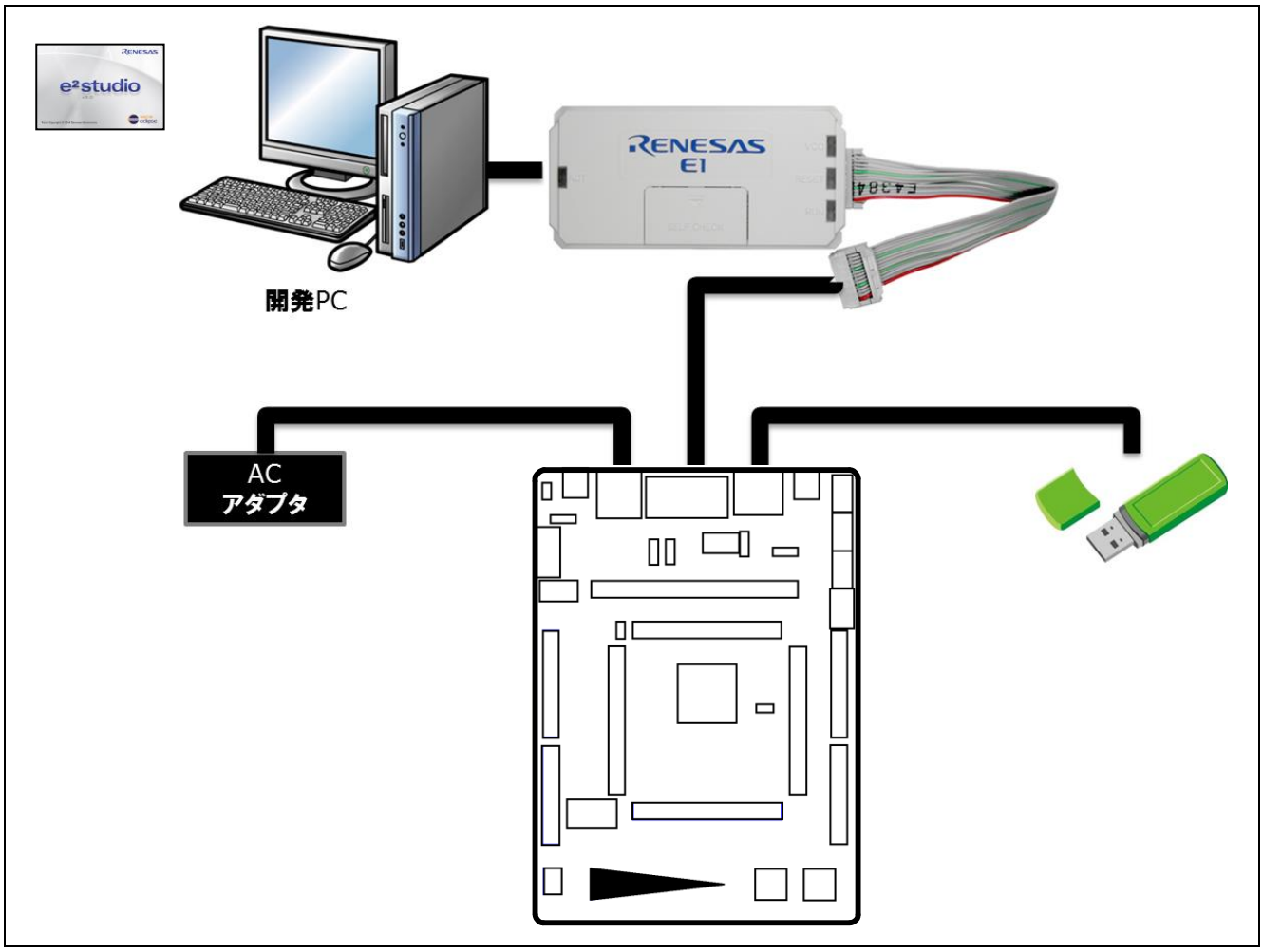


図 4.1　動作環境例

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

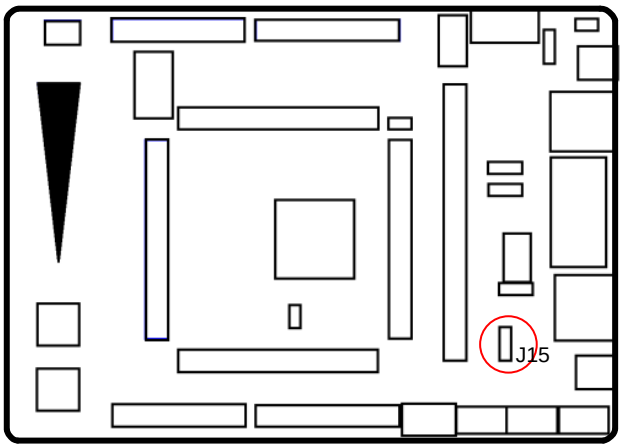
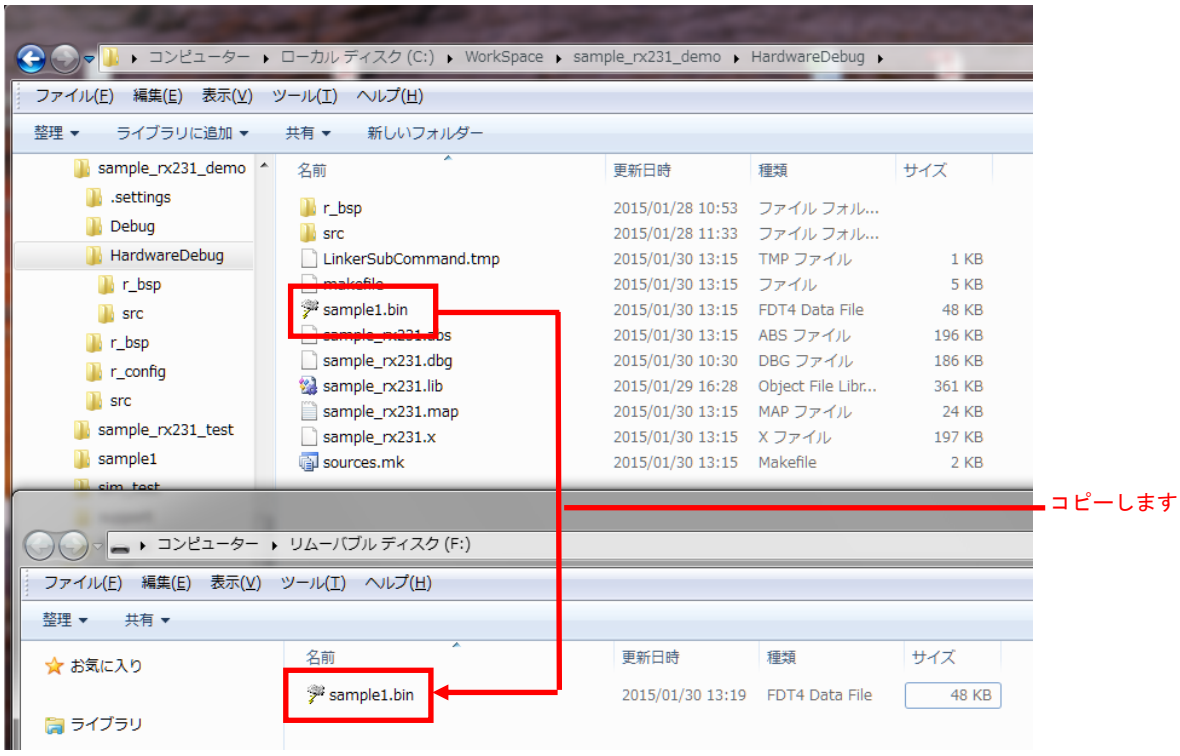


図 4.2　RSK でのジャンパの位置

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「sample1.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「sample1¥release」フォルダに入っている「sample1.bin」ファイルを USB メモリにコピーします。



「demo」フォルダに同梱されている「sample2.zip」ファイルは、「sample1.zip」と同様にサンプルプログラムとして使用できます。「sample2¥release」フォルダに入っている「sample2.bin」ファイルを USB メモリにコピーしてご使用ください。「sample3.zip」ファイルは、「sample3¥DefaultBuild」フォルダに入っている「sample3.bin」ファイルを USB メモリにコピーしてご使用ください。

sample1 プログラム及び、sample2 プログラム、sample3 の動作については、4.4サンプルプログラムの動作を参照ください。

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。

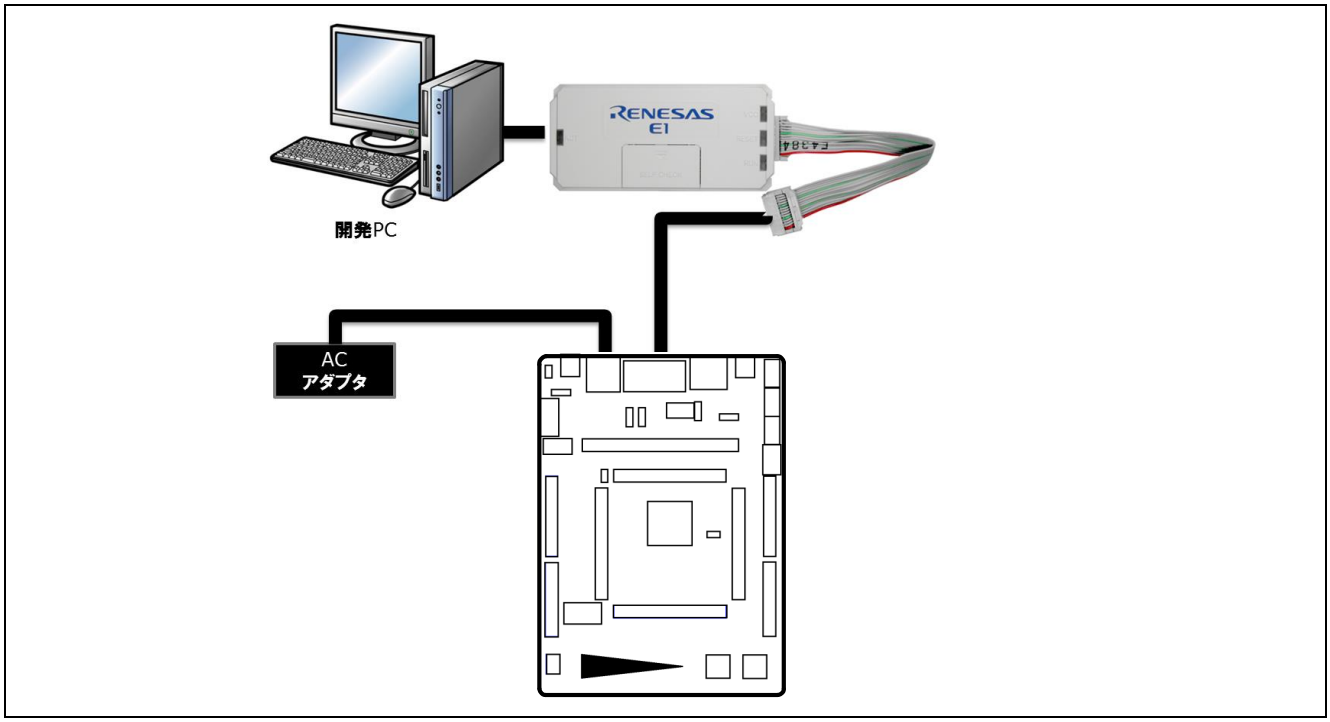
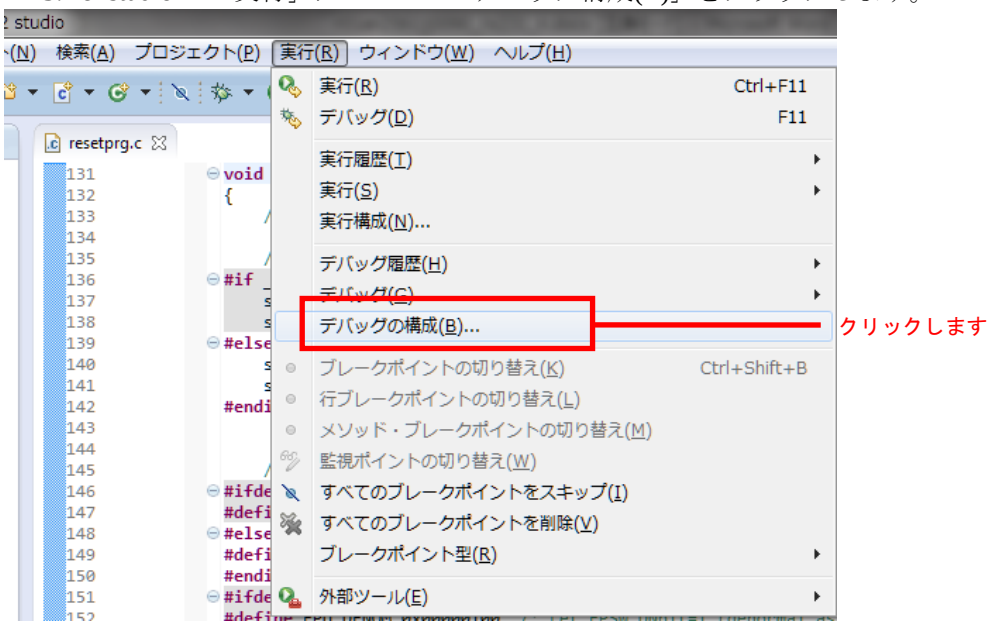
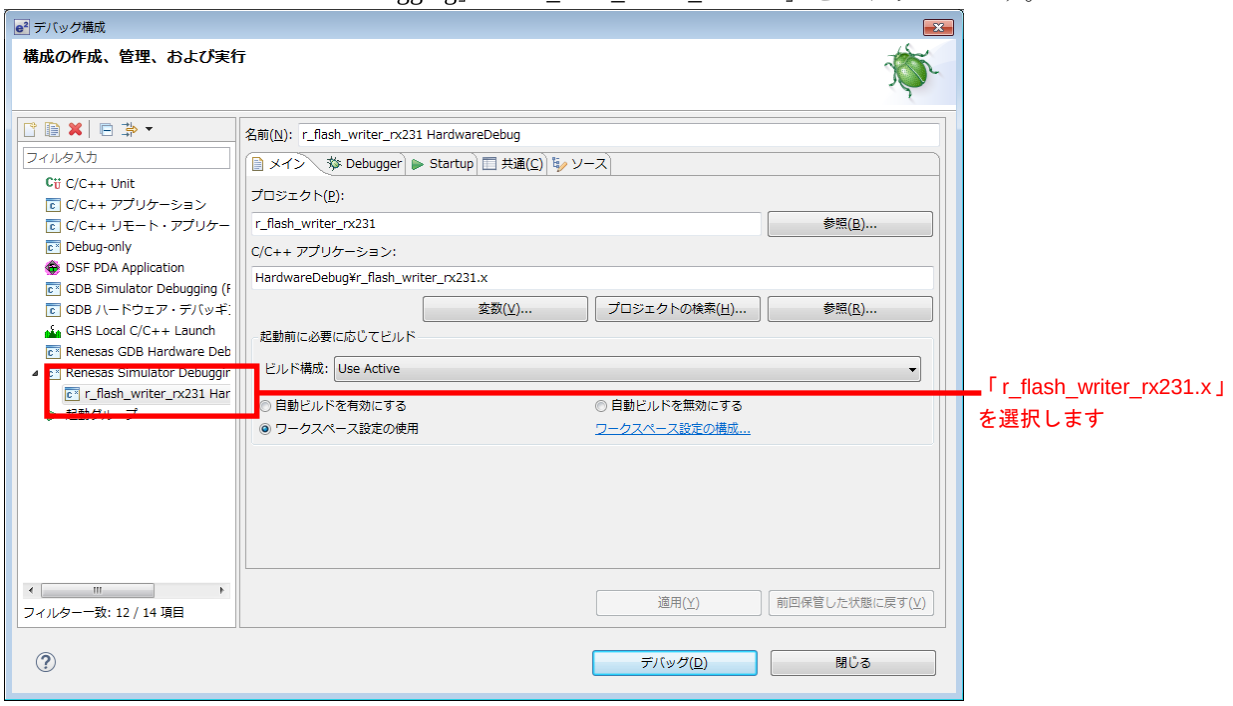


図 4.3　ハードウェア構成

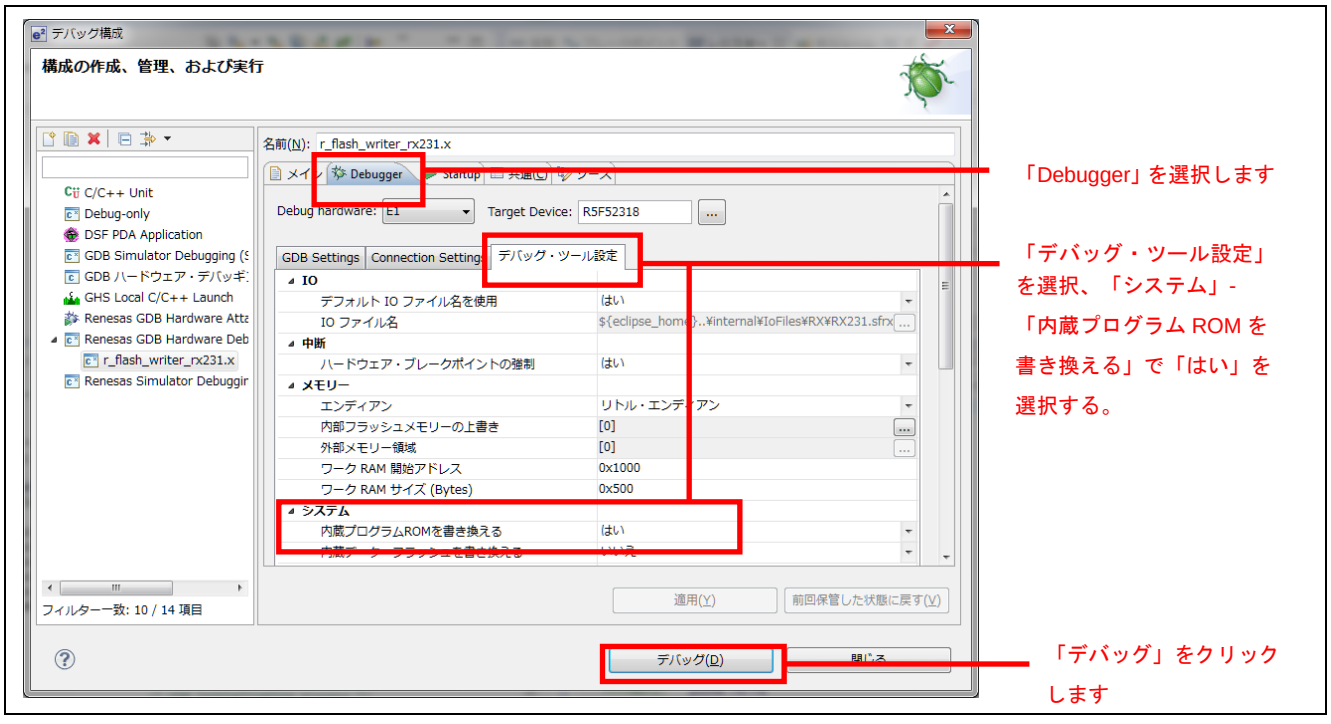
3. e² studio の「実行」メニューの「デバッグ構成(B)」をクリックします。



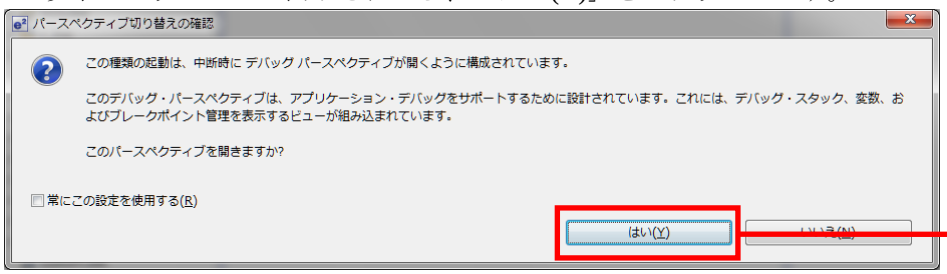
4. 「Renesas GDB Hardware Debugging」の「r_flash_writer_rx231.x」をクリックします。



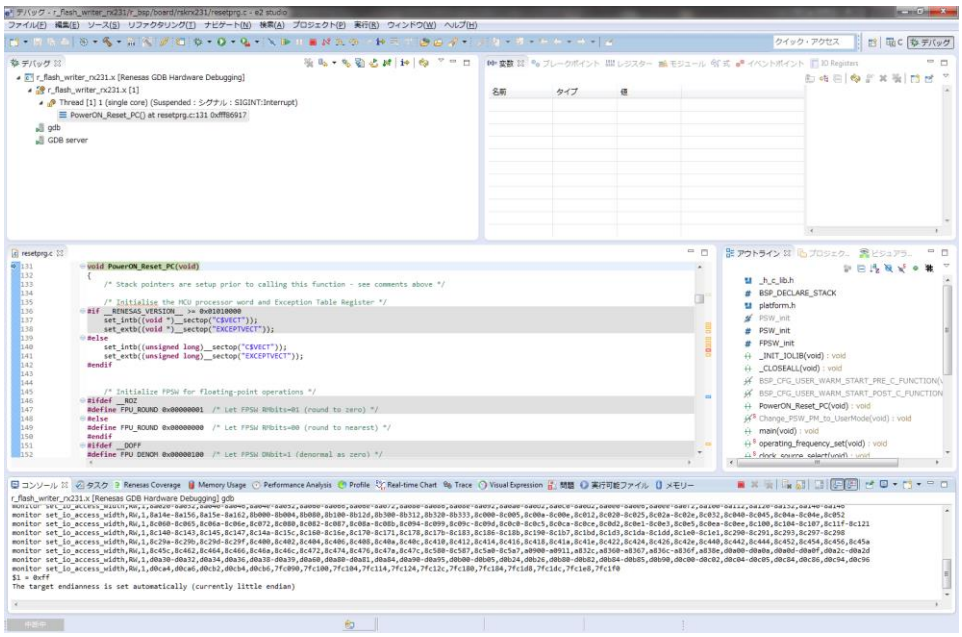
5. 「デバッグ・ツール設定」をクリックし、「システム」―「内蔵プログラム ROM を書き換える」の選択を「はい」に設定して「デバッグ」をクリックします。



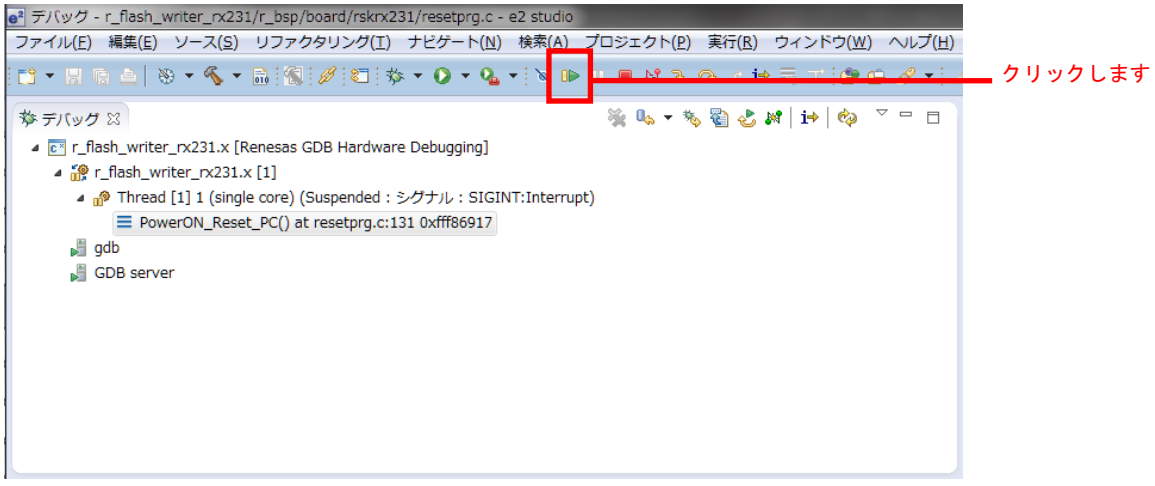
6. 以下のメッセージが表示されたら、「はい(Y)」をクリックします。



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。
   以降は、RSK で操作しながら、メインプログラムを動作させ、RSK の Pmod LCD に表示される情報を確認します。

10. Pmod LCD に”MAIN”と表示されたら、RSK の USB - A コネクタ(USB0-1)に USB メモリを挿し込みます。（Pmod LCD に”ATTACH”が表示されます。）

11. タッチセンサの場合、タッチセンサの+側をタッチするとカーソルが上に、タッチセンサの+側をタッチするとカーソルが下に移動します。書き込むサンプルを選択し、タッチセンサの 1 をタッチします。
    S/W の場合、S/W1 を押すとカーソルが上に、S/W2 を押すとカーソルが下に移動します。書き込むサンプルを選択し、S/W3 を押します。

12. メインプログラムが USB メモリからデータを読み出し、内蔵フラッシュに読み出したデータを書き込みます。（Pmod LCD に”START”、”STOP”、”DETACH”と順に表示されます。）

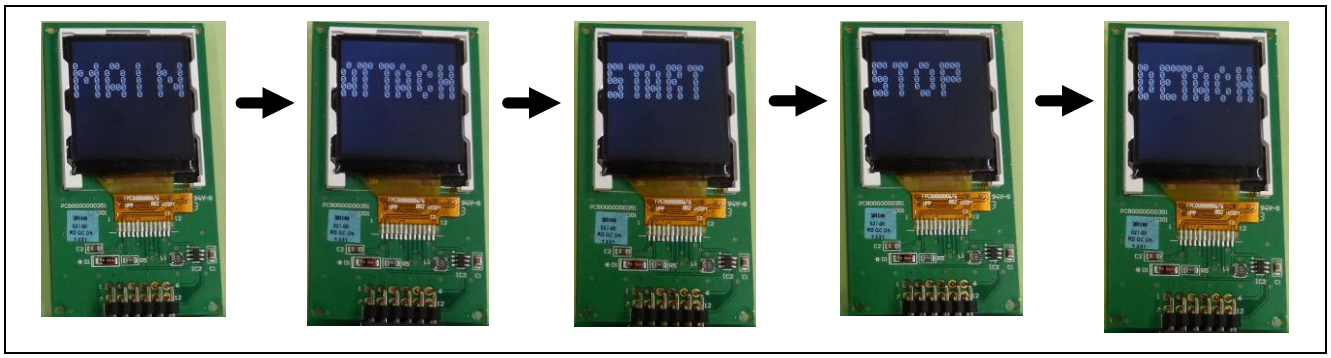


図 4.4　LCD 表示内容

13. RSK の Pmod LCD が初期化され、サンプルプログラムが表示されます。
    （サンプルプログラムを USB メモリから読み出し、内蔵フラッシュメモリに書き込まれます。）
    表示内容は、「4.4 サンプルプログラムの動作」を参照してください。

4.4　サンプルプログラムの動作

(1)　sample1¥sample1.bin、”TEST!”の場合

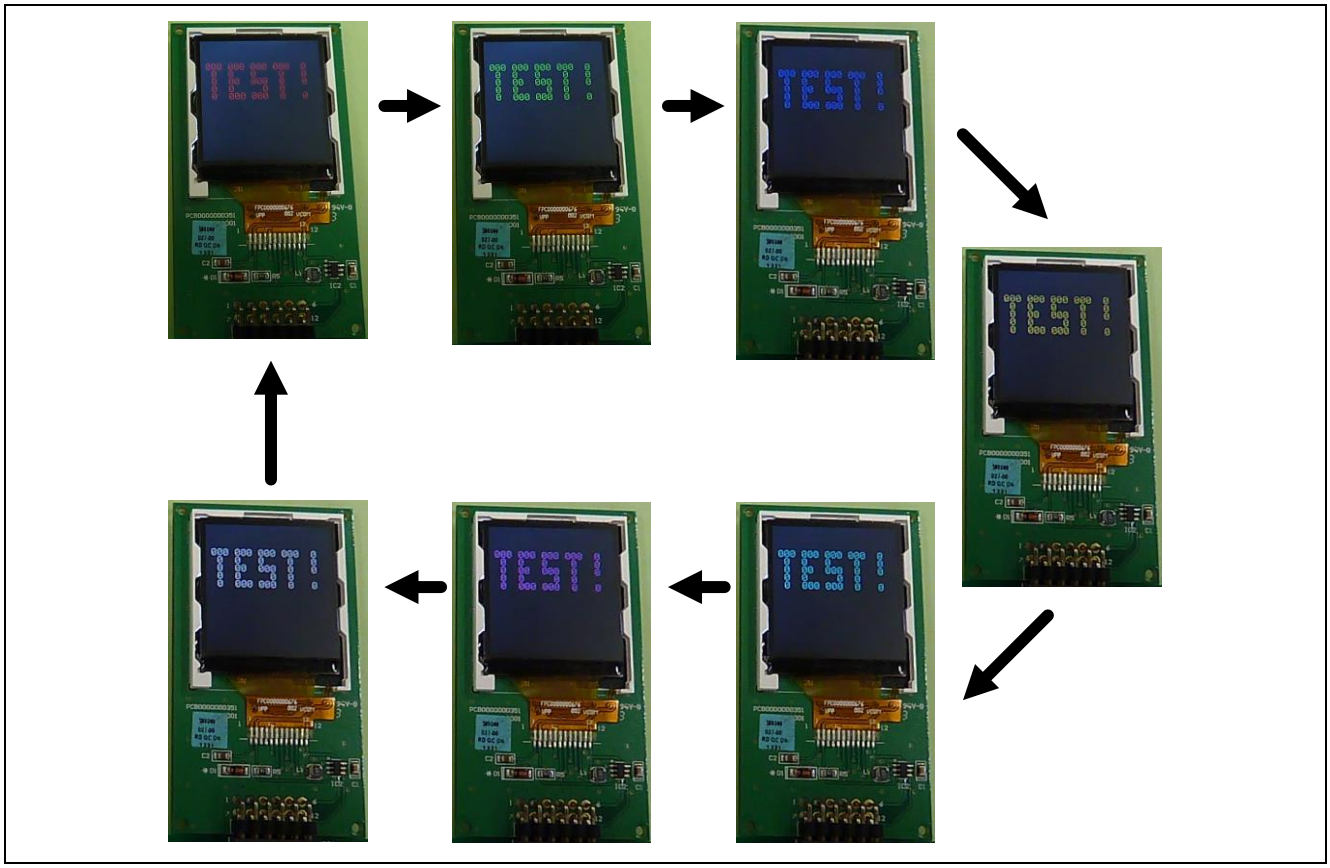


図 4.5　LCD 表示内容（sample1.bin）

(2)　sample2¥sample2.bin、”DEMO”の場合

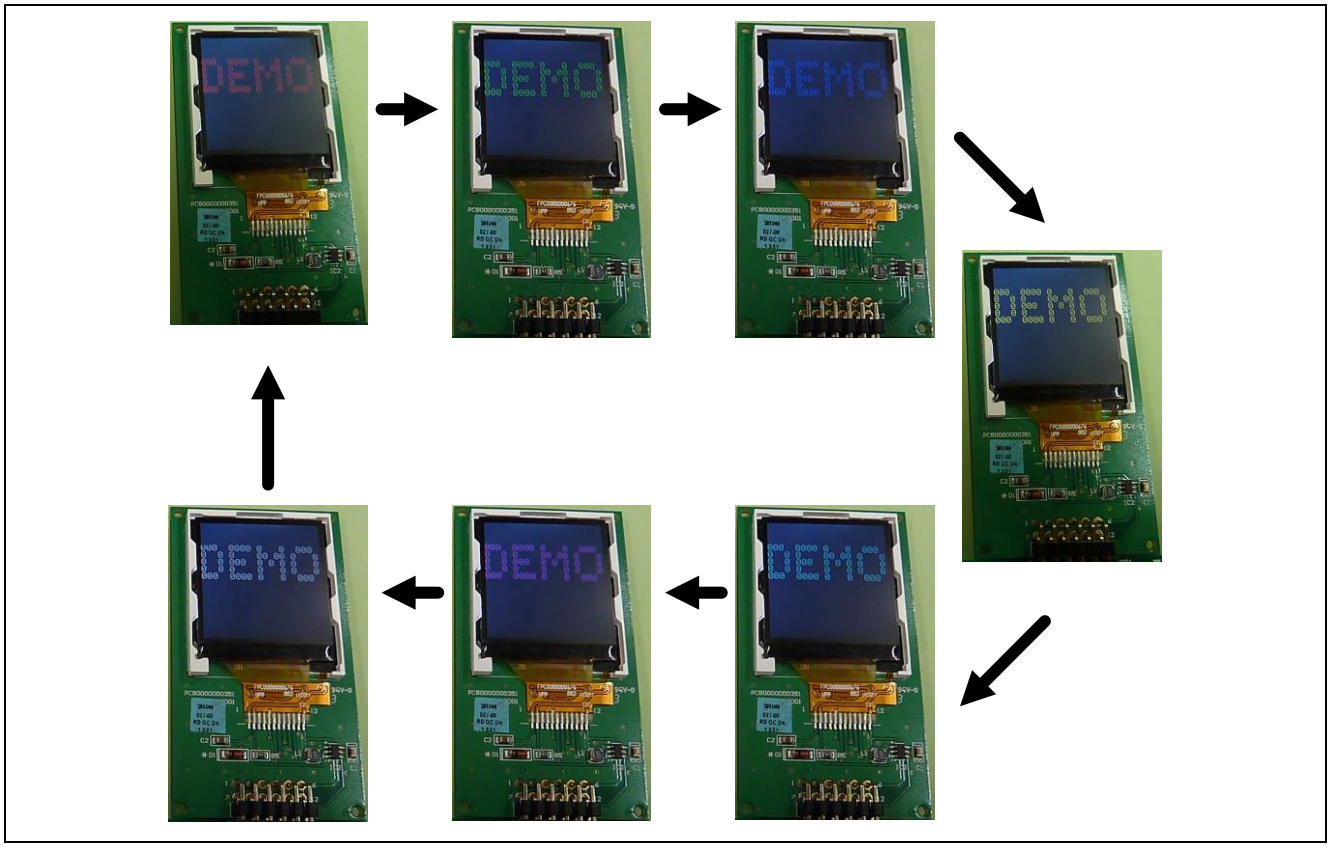


図 4.6　LCD 表示内容（sample2.bin）

(3)　sample3¥sample3.bin、”Touch”の場合

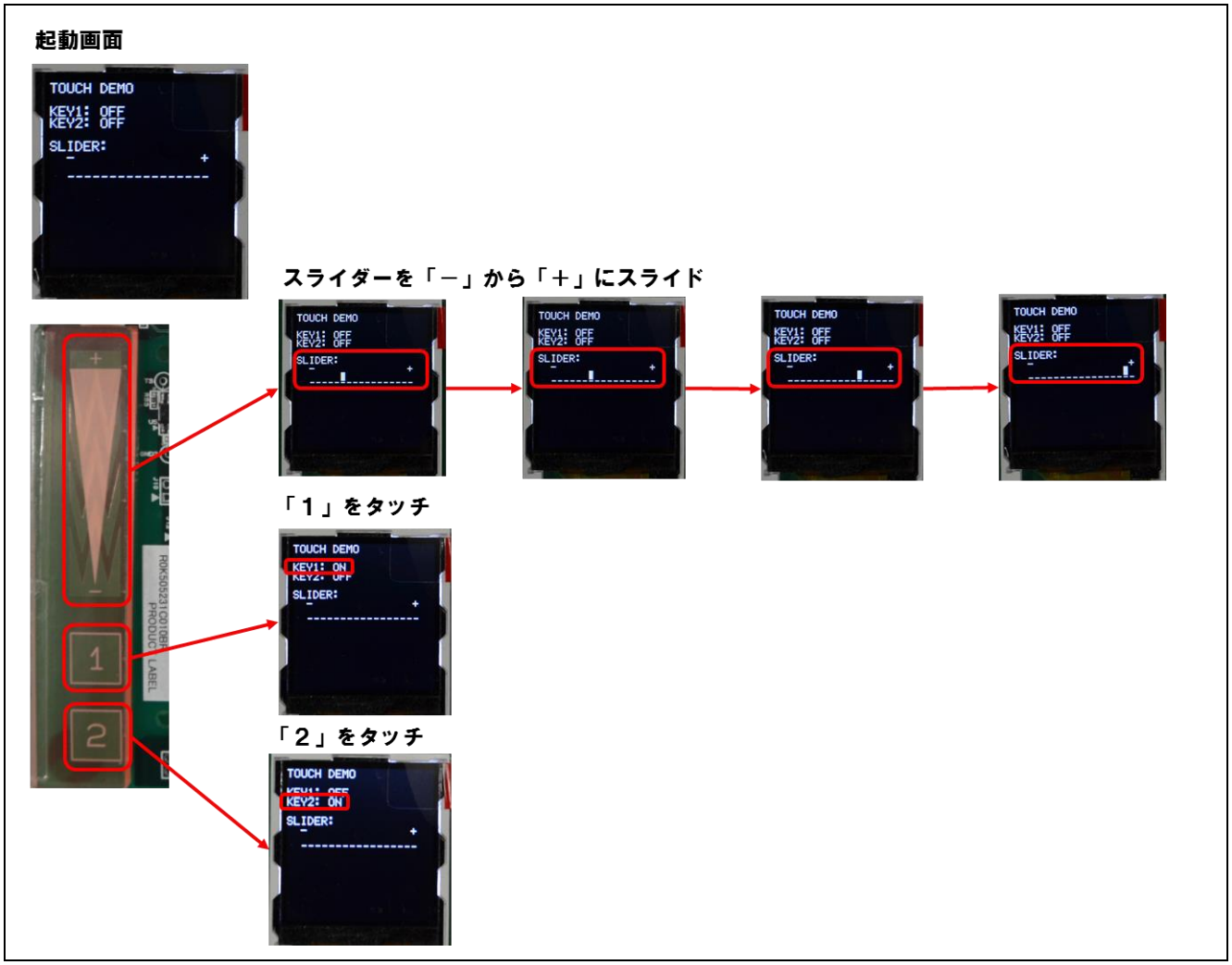


図 4.7　LCD 表示内容（sample3.bin）

5.　アプリケーションの概要

本アプリケーションは、メインプログラムとサンプルプログラムで構成しています。

メインプログラムを使って、USB メモリにあるサンプルプログラムのバイナリファイルを読み込み、内蔵フラッシュメモリに書き込みます。書き込みが完了すると、スタートアッププログラム保護機能を使用して、サンプルプログラムの実行を行います。その後にリセットがかかると元のメインプログラムが起動します。

スタートアッププログラム保護機能は、リセット後にリセットベクタの情報切り替えを保持する/しないを選択でき、本アプリケーションでは「保持しない」を選択しています。そのため、リセットベクタの情報変更は、リセットがかかると元のメインプログラムが起動する状態に戻ります。詳細は、ユーザーズマニュアル ハードウェア編 の「スタートアッププログラム保護機能」を参照してください。

図 5.1に本アプリケーションのステータス遷移を示します。

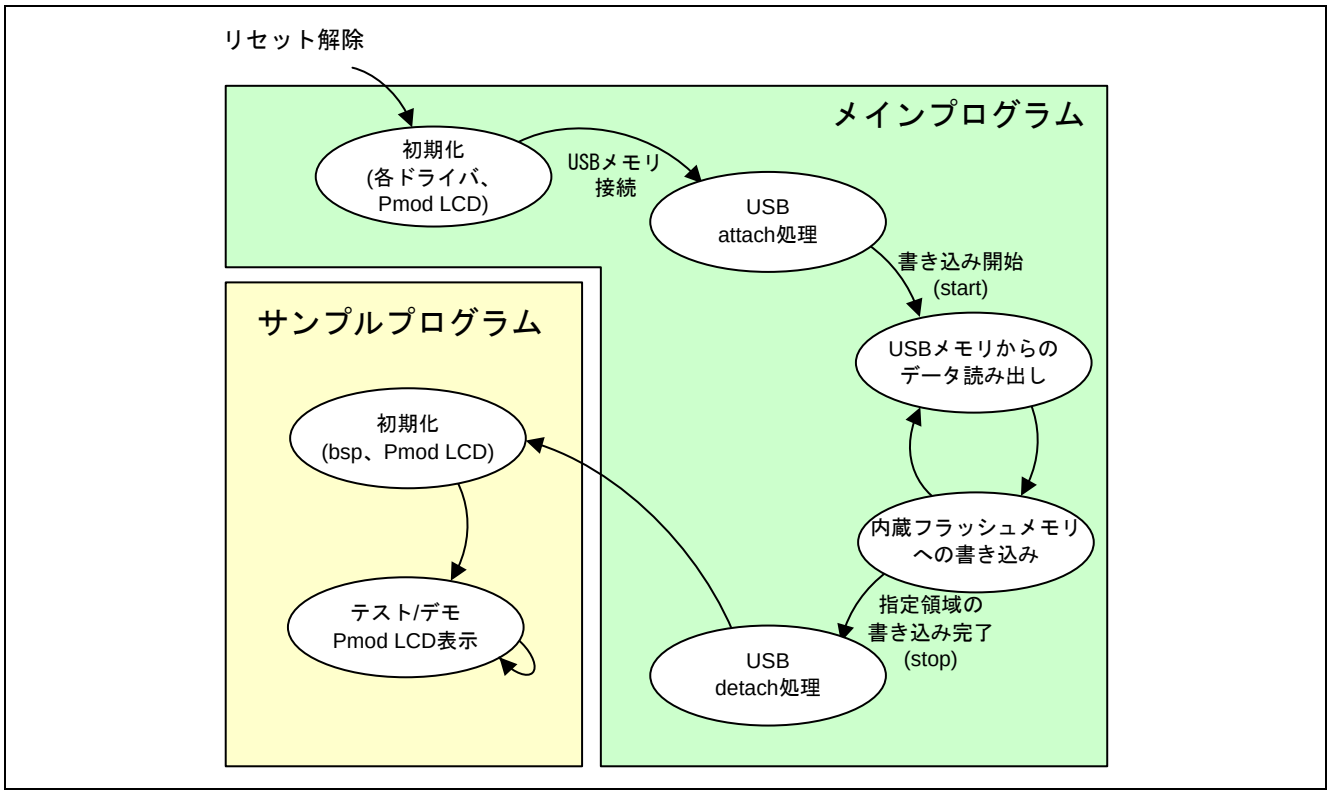


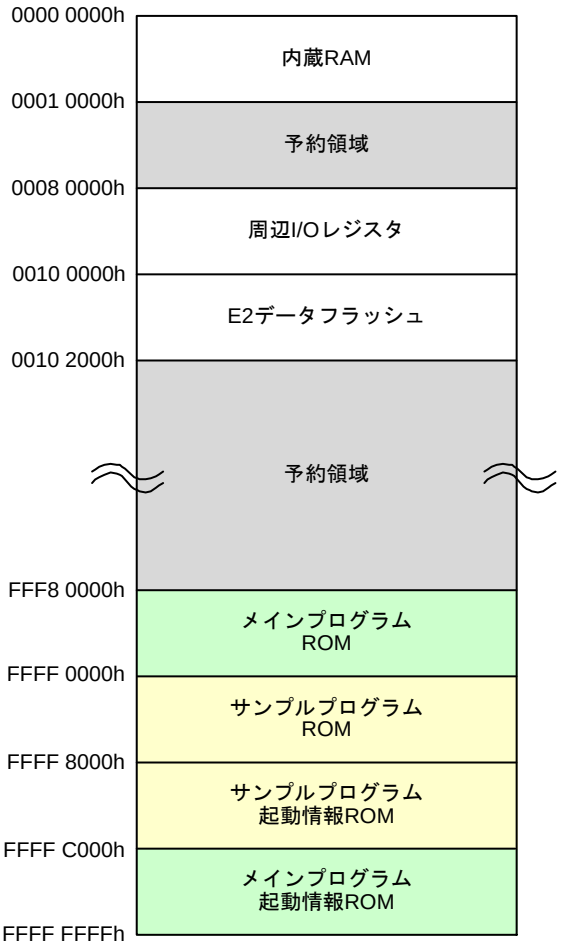
図 5.1 アプリケーションのステータス

(1) 各ドライバ、Pmod LCD の初期化
(2) USB ドライバを使用して、USB 接続
(3) USB ドライバと FAT ファイルシステムを使用し、USB メモリからデータ(注 1)の読み出し
(4) Flash API を使用して、内蔵フラッシュメモリへのデータ(注 1)書き込み
(5) USB の切断後、Flash API を使用して、スタートアッププログラムを切り替え
(6) 書き込んだデータ(注 1)へジャンプ。
　～ ここまでがメインプログラムの実行 ～
(7) BSP、Pmod LCD の初期化
(8) Pmod LCD にデータを表示

注 1.　　サンプルプログラムの実行バイナリファイルを指します。

5.1　メモリ構成

本アプリケーションで使用する RSKRX231 に搭載している RX231 マイコンのメモリマップを示します。



注1.本メモリ構成は、RSKRX231に搭載しているRX231マイコンのメモリ構成です。
使用CPU 型番及びメインプログラムサイズによりセクションアドレスが異なります。
また、ユーザシステムにより修正が必要になります。

図 5.2　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_rx231」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
	main.h	アプリケーション設定ヘッダファイル (読み出しファイル名、書き込み領域設定)
	r_usb_hmsc_apl.c	USB メインプログラム処理
	r_usb_hmsc_apl.h	r_usb_hmsc_apl.c のヘッダファイル
	r_rsk_flashdriver.c	フラッシュメモリ プログラム処理
	r_rsk_flashdriver.h	r_rsk_flashdriver.c のヘッダファイル
	r_rsk_leddriver.c	LED 出力用 プログラム
	r_rsk_leddriver.h	r_rsk_leddriver.c のヘッダファイル
	lcd.c、ascii.c、r_cg_sci.c、r_cg_sci_user.c	Pmod LCD 表示用 プログラム
	lcd.h、ascii.h、r_cg_sci.h、r_cg_macrodriver.h	Pmod LCD 表示用プログラムのヘッダファイル
	r_rsk_keydrive.c	プッシュ S/W 出力用 プログラム
	r_rsk_keydriver.h	r_rsk_keydriver.c のヘッダファイル

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_flash_rx	フラッシュメモリ(Flash API) ファイル群
r_usb_basic_mini	USB Basic Firmware ファイル群
r_usb_hmsc_mini	USB ホストマスストレージクラス(USB HMSC) 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_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_config	各 FIT モジュールの config ファイル

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	メイン処理
	usb_mcu_init	USB ポート初期化処理
	usb_board_init	LCD 初期化処理、USB 割り込み許可設定
	apl_init	アプリの管理用テーブルの初期化
	jmp_user_program	サンプルプログラムへのジャンプ処理
r_usb_hmsc_apl.c	usb_driver_init	USB ドライバ初期化処理
	msc_registration	USB コールバック関数登録処理
	usb_hmsc_driver	USB HMSC ドライバタスク処理
	msc_connect_wait	USB デバイス検出待ち処理
	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設定処理
	msc_data_read	USB メモリからのデータリード､フラッシュメモリへのデータ書き込み処理
	msc_detach_device	USB 切断処理
	usb_hsmpl_device_state	USB ドライバコールバック処理
	msc_configured	USB デバイス検出通知処理
	msc_drive_complete	USB デバイス接続通知処理
	msc_detach	USB デバイス切断通知処理
	msc_event_set	イベント設定処理
	msc_event_get	イベント取得処理
r_rsk_flashdriver.c	SAMPLE_FLASH_Write	フラッシュメモリの書き換え処理
lcd.c(注)	Init_LCD	Pmod LCD 初期化
	Display_LCD	Pmod LCD 表示処理
	DisplaySetFontColour	Pmod LCD フォント色切り替え処理
r_rsk_leddriver.c (注)	usb_cpu_LedInitial	LED 初期化
	usb_cpu_led_set_data	LED 表示処理
r_rsk_keydrive.c	usb_cpu_key_read_touch	タッチセンサ読み込み処理
	usb_cpu_key_read	S/W 読み込み処理
	usb_cpu_sw_data	ファイル選択操作判定処理
	usb_cpu_sw1_data	ファイル選択操作“上”判定処理
	usb_cpu_sw2_data	ファイル選択操作“下”判定処理
	usb_cpu_sw3_data	ファイル選択操作“決定”判定処理
TouchAPI	-	タッチ関連処理 Workbench6 の出力ファイル

(注)　LED 処理、LCD 処理の詳細は、ルネサススタータキットのサンプルコードのアプリケーションノートを参照ください。

6.3　フローチャート

(1)　メイン処理

図 6.1、図 6.2にメイン処理のフローチャートを示します。

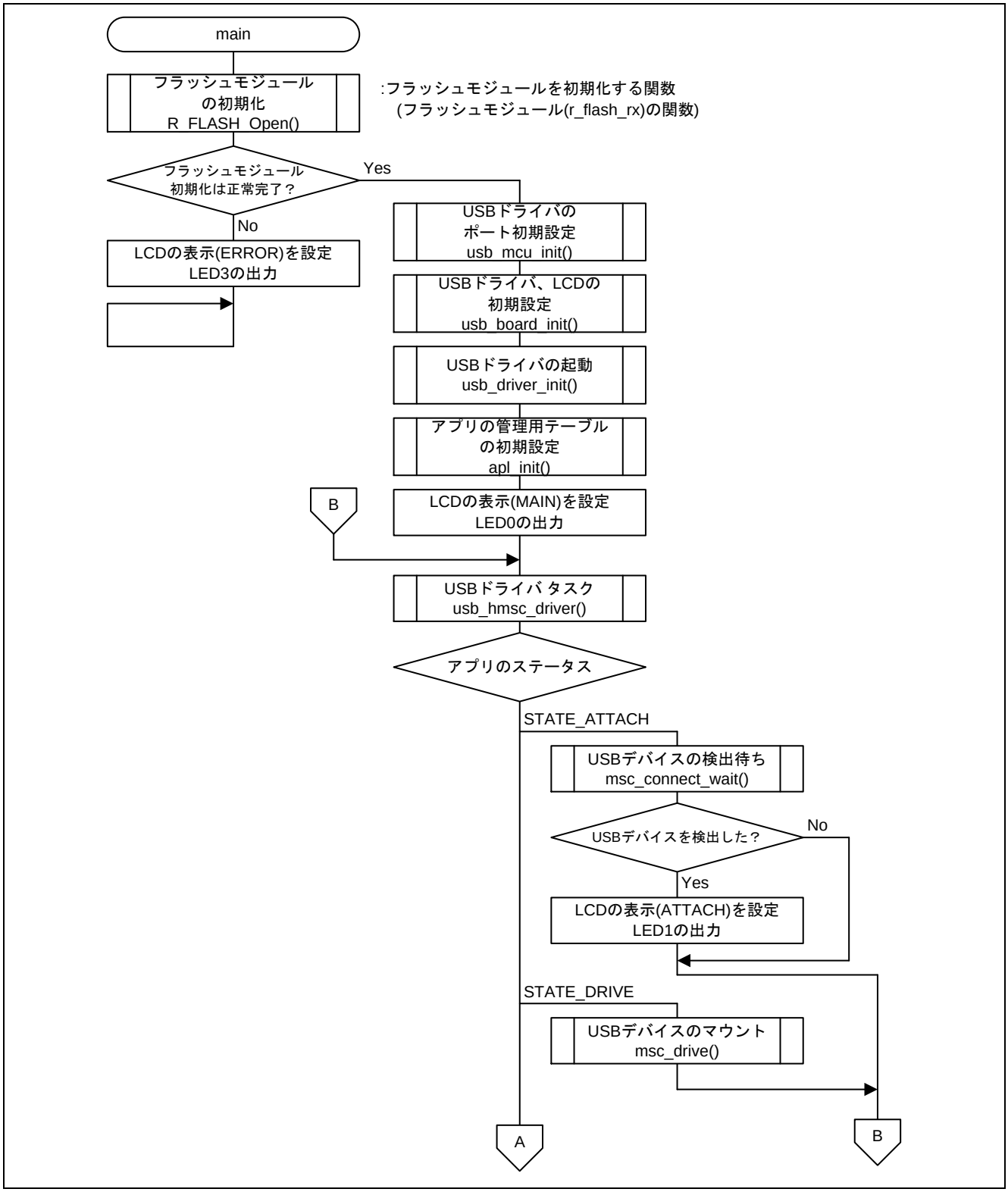


図6.1　メイン処理(1)

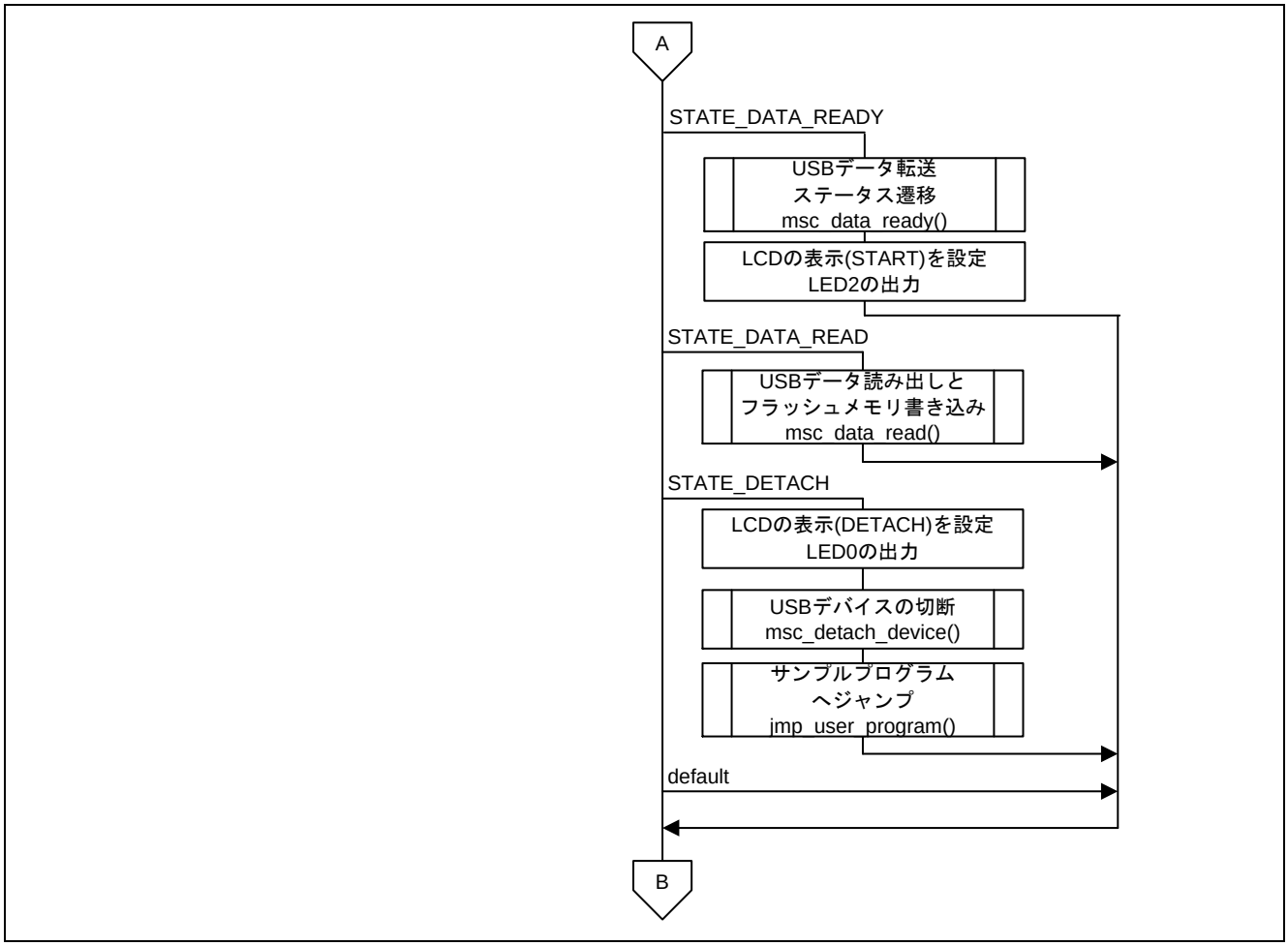


図6.2　メイン処理(2)

(2)　USB ポート初期化

図 6.3にUSB ポート初期化処理のフローチャートを示します。

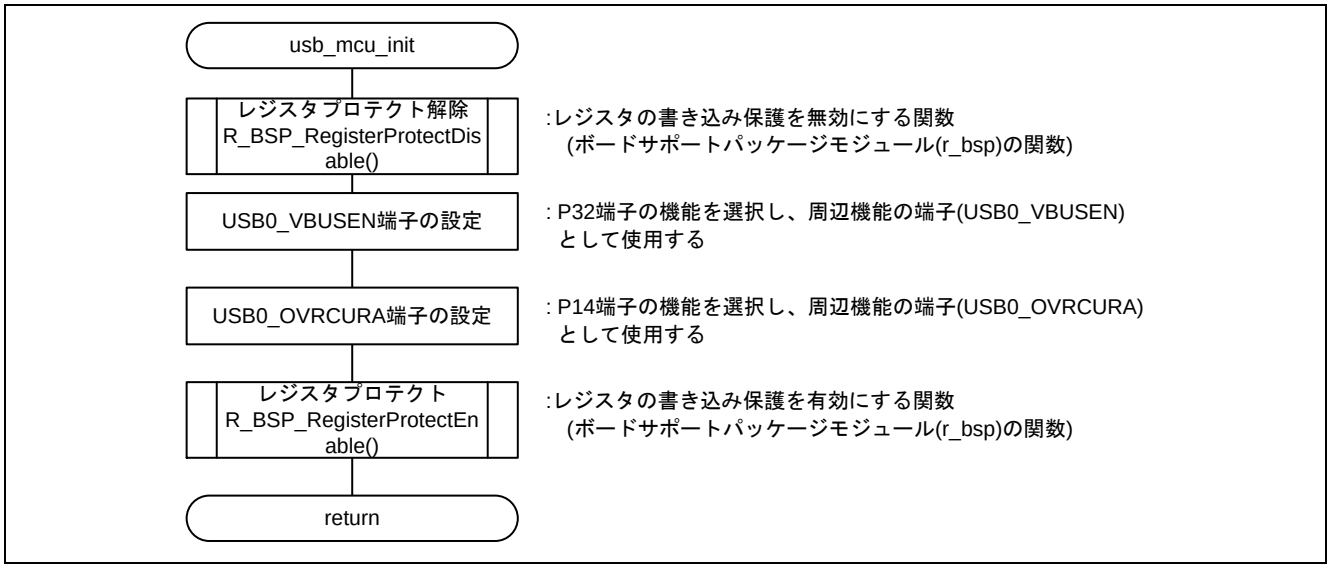


図 6.3　USB ポート初期化処理

(3)　LED、LCD 初期化、USB 割り込み許可設定

図 6.4にLED、LCD 初期化、USB 割り込み許可設定処理のフローチャートを示します。

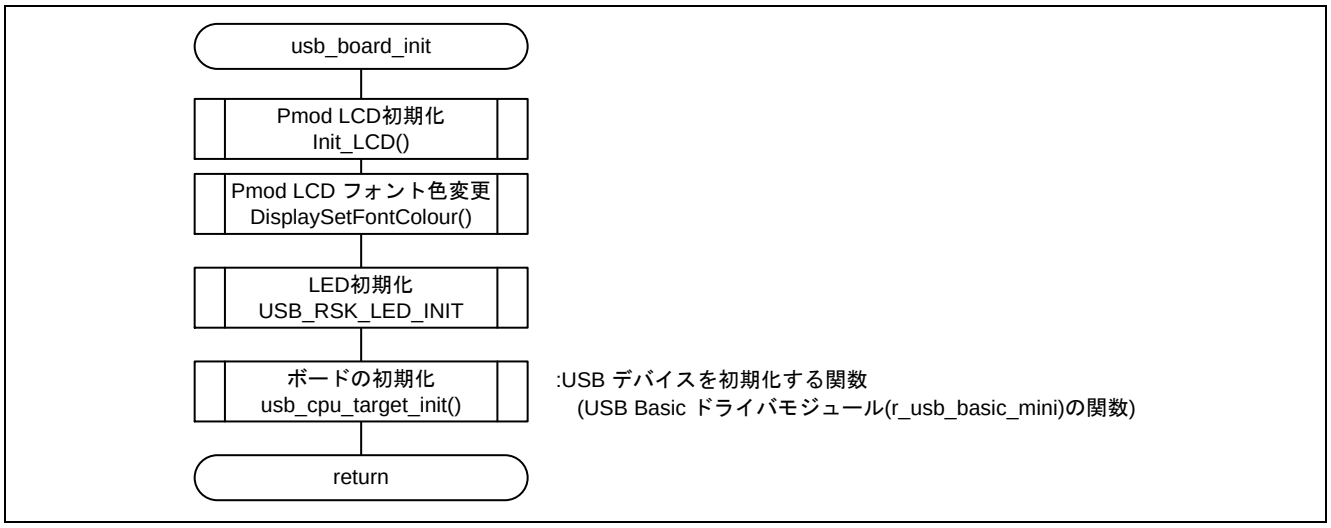


図 6.4 LED、LCD 初期化、USB 割り込み許可設定処理

(4)　管理用テーブルの初期化

図 6.5にアプリの管理用テーブルの初期化処理のフローチャートを示します。

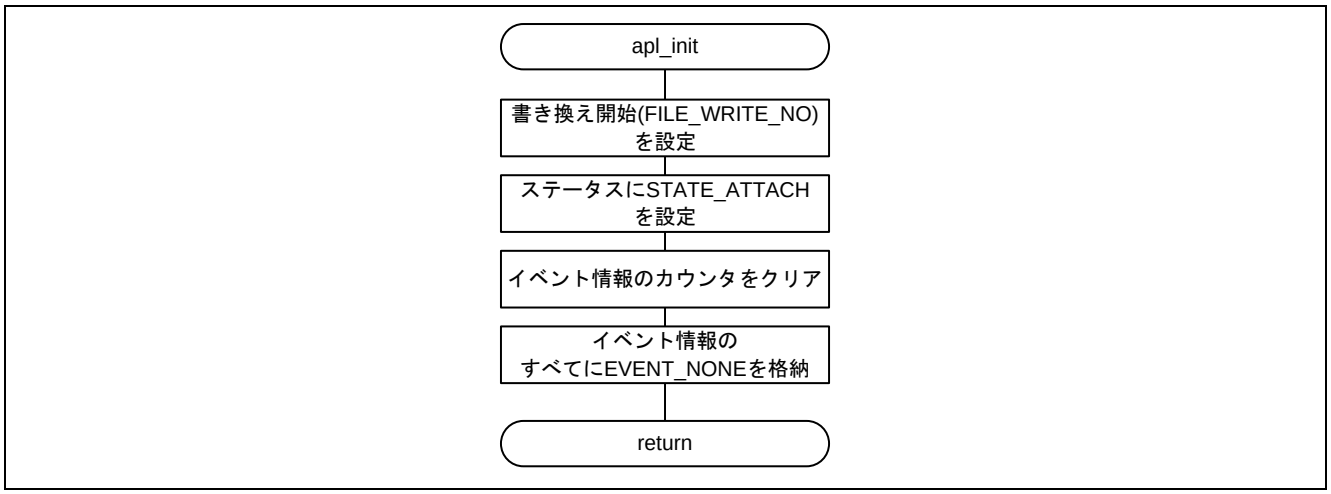


図 6.5 アプリの管理用テーブルの初期化処理

(5) ユーザプログラムへの分岐

図 6.6にユーザプログラムへの分岐処理のフローチャートを示します。

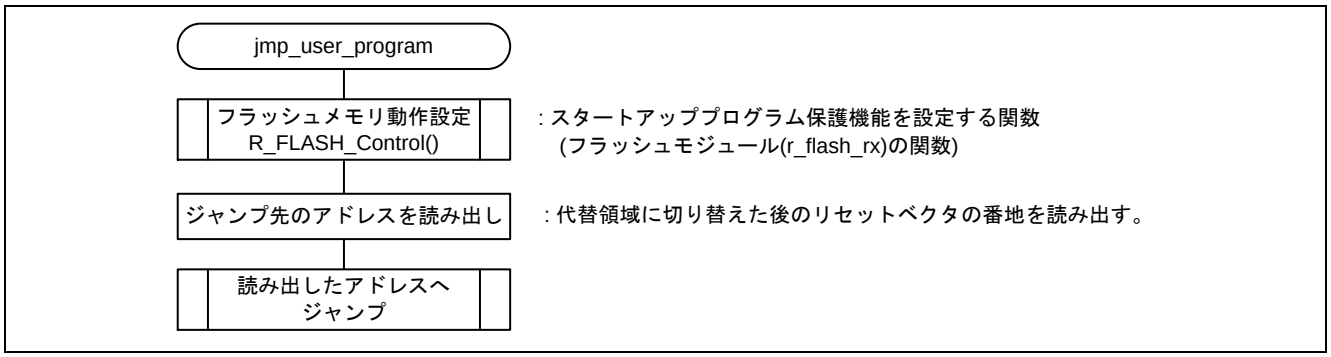


図 6.6 ユーザプログラムへの分岐処理

(6) USB ドライバ初期化

図 6.7にUSB ドライバ初期化処理のフローチャートを示します。

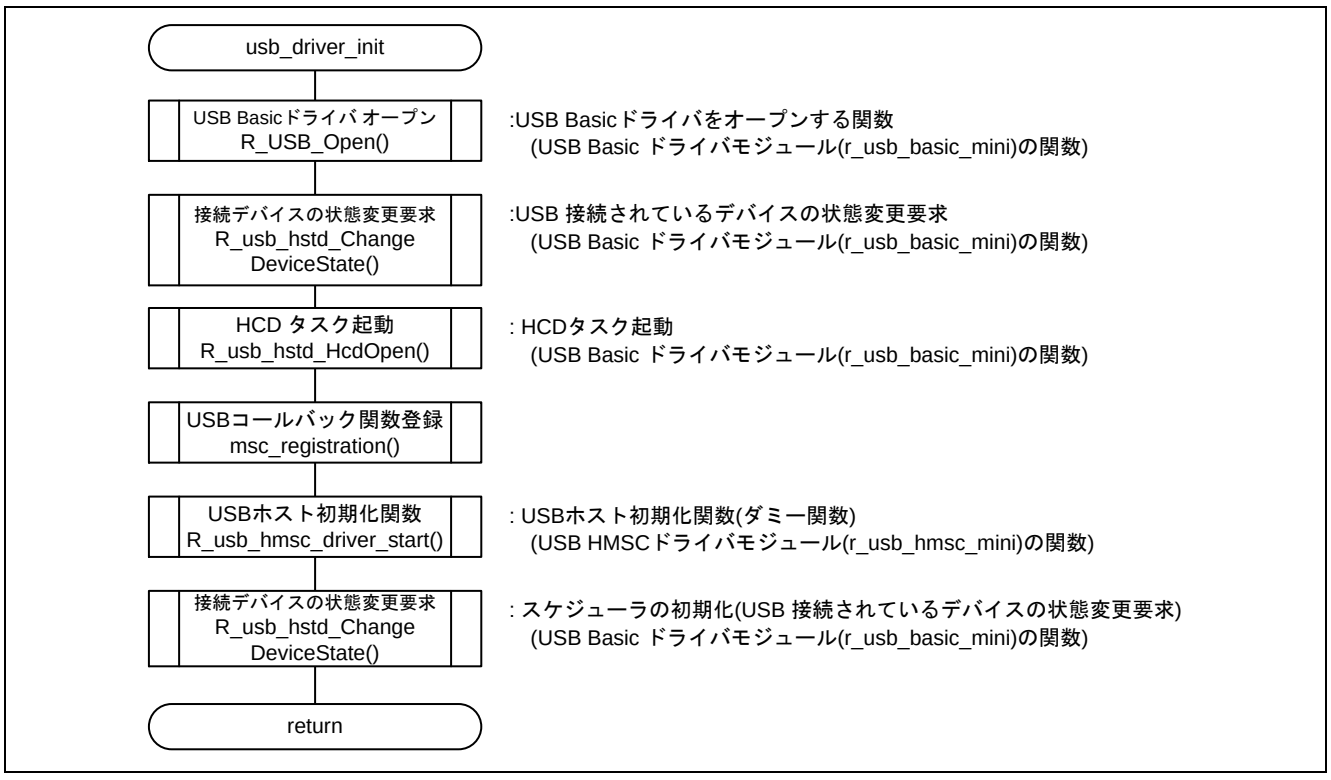


図 6.7 USB ドライバ初期化処理

(7) USB コールバック関数登録

図 6.8にUSB コールバック関数登録処理のフローチャートを示します。

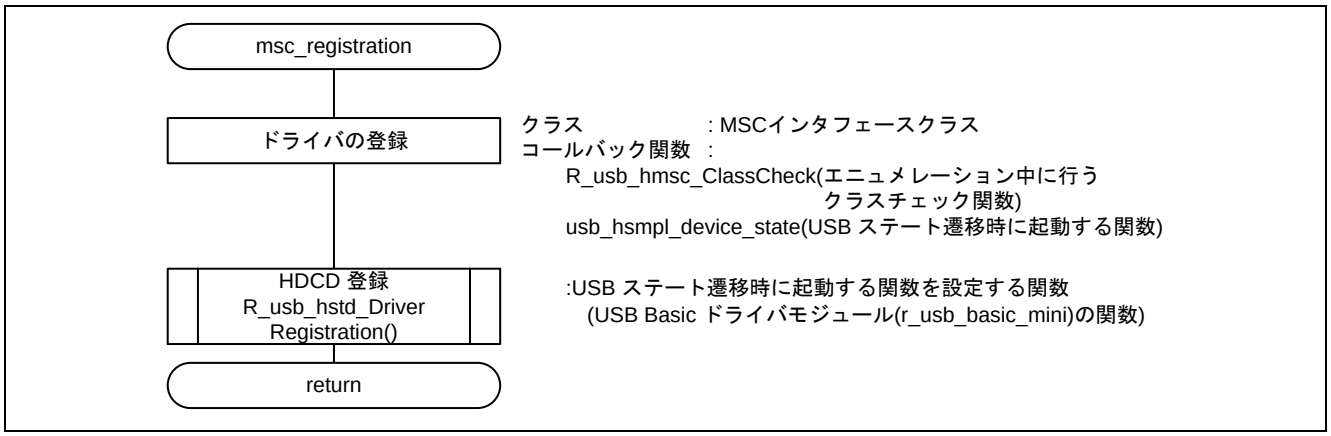


図 6.8 USB コールバック関数登録処理

(8) USB HMSC ドライバタスク

図 6.9にUSB HMSC ドライバタスク処理のフローチャートを示します。

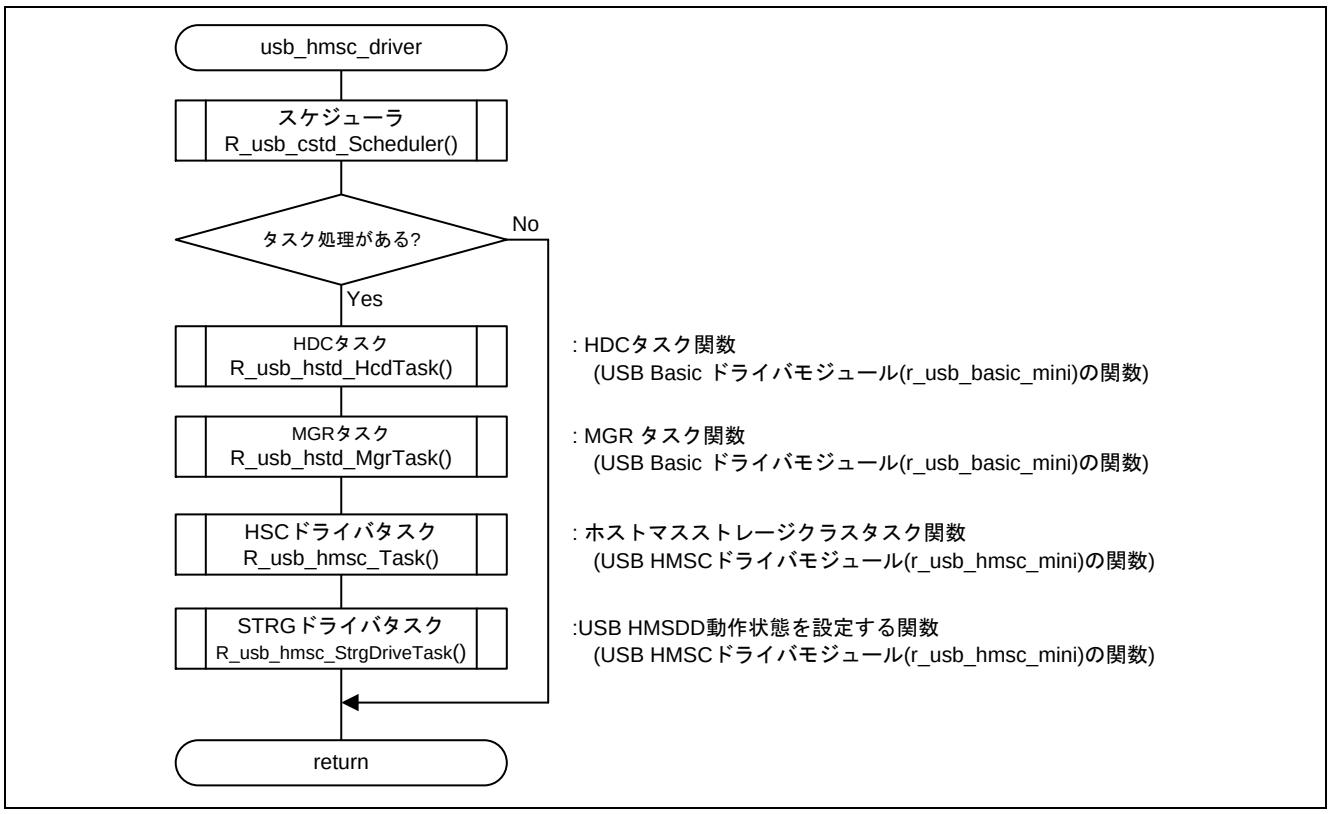


図 6.9 USB HMSC ドライバタスク処理

(9)　USB デバイス検出待ち

図 6.10にUSB デバイス検出待ち処理のフローチャートを示します。

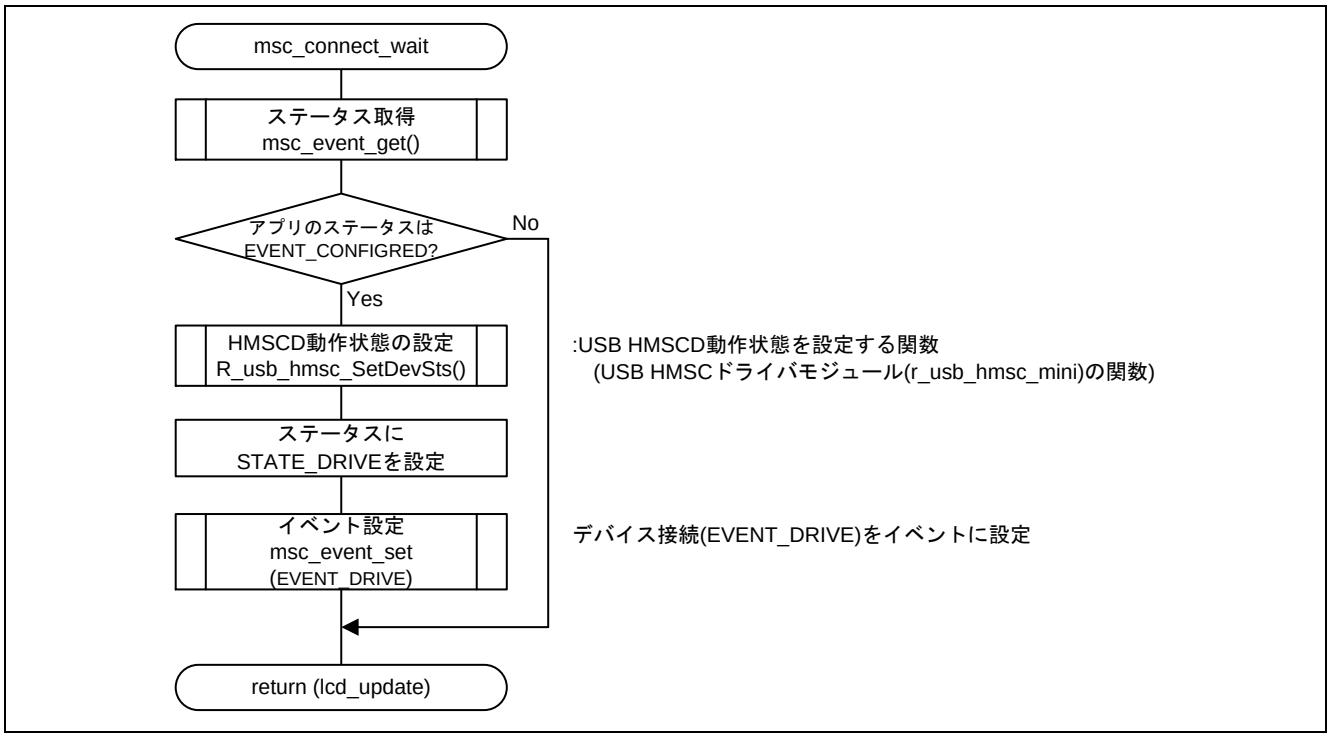


図 6.10 USB デバイス検出待ち処理

(10) USB デバイス接続、TFAT ファイルシステムマウント

図 6.11にUSB デバイス接続、TFAT ファイルシステムマウント処理のフローチャートを示します。

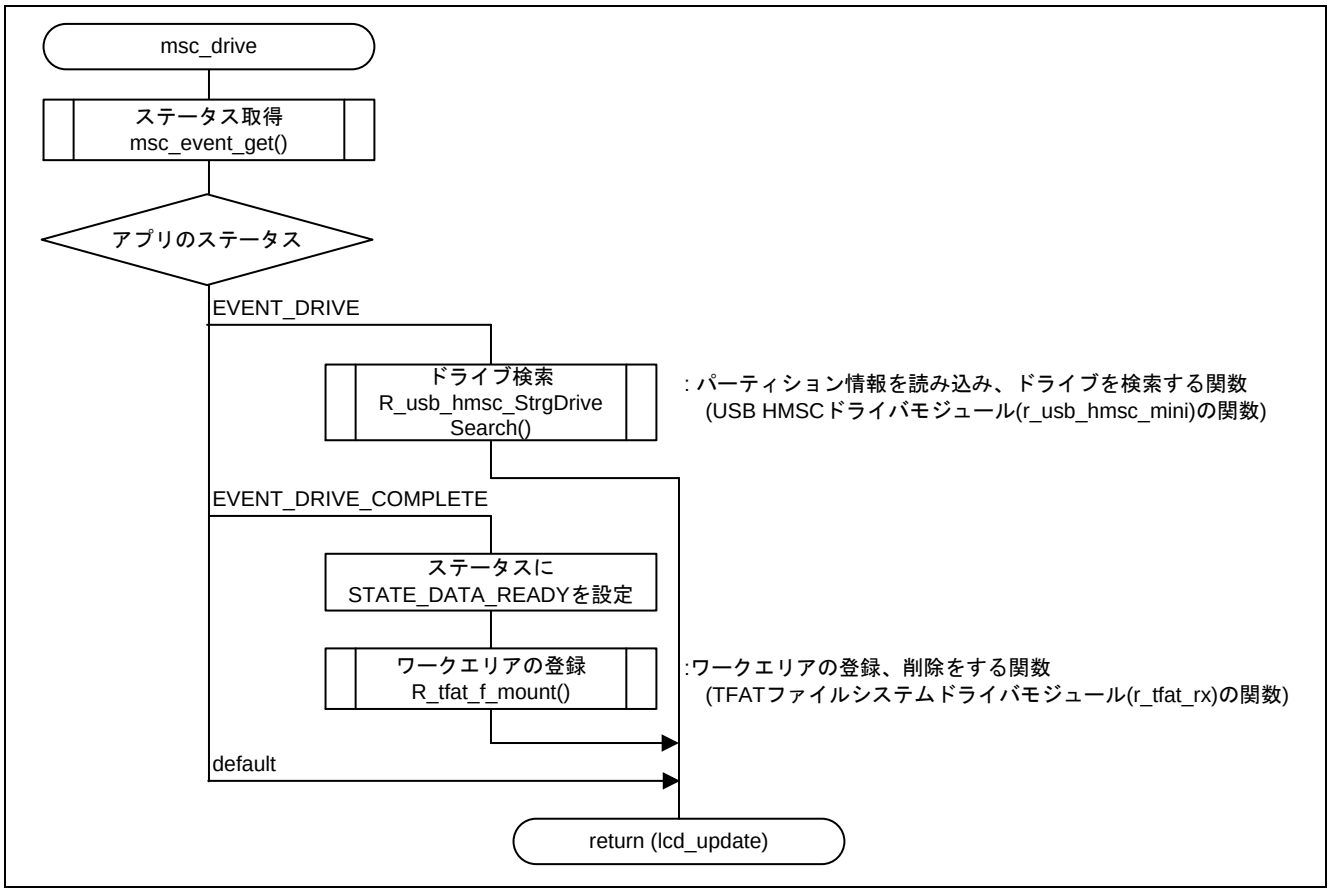


図 6.11 USB デバイス接続、TFAT ファイルシステムマウント処理

(11) リードステータス前の設定

図 6.12にリードステータス前の設定処理のフローチャートを示します。

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

図 6.13、図 6.14に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のフローチャートを示します。USB メモリに格納しているファイルの名称、書き換えの先頭アドレスは、この関数で設定しています。

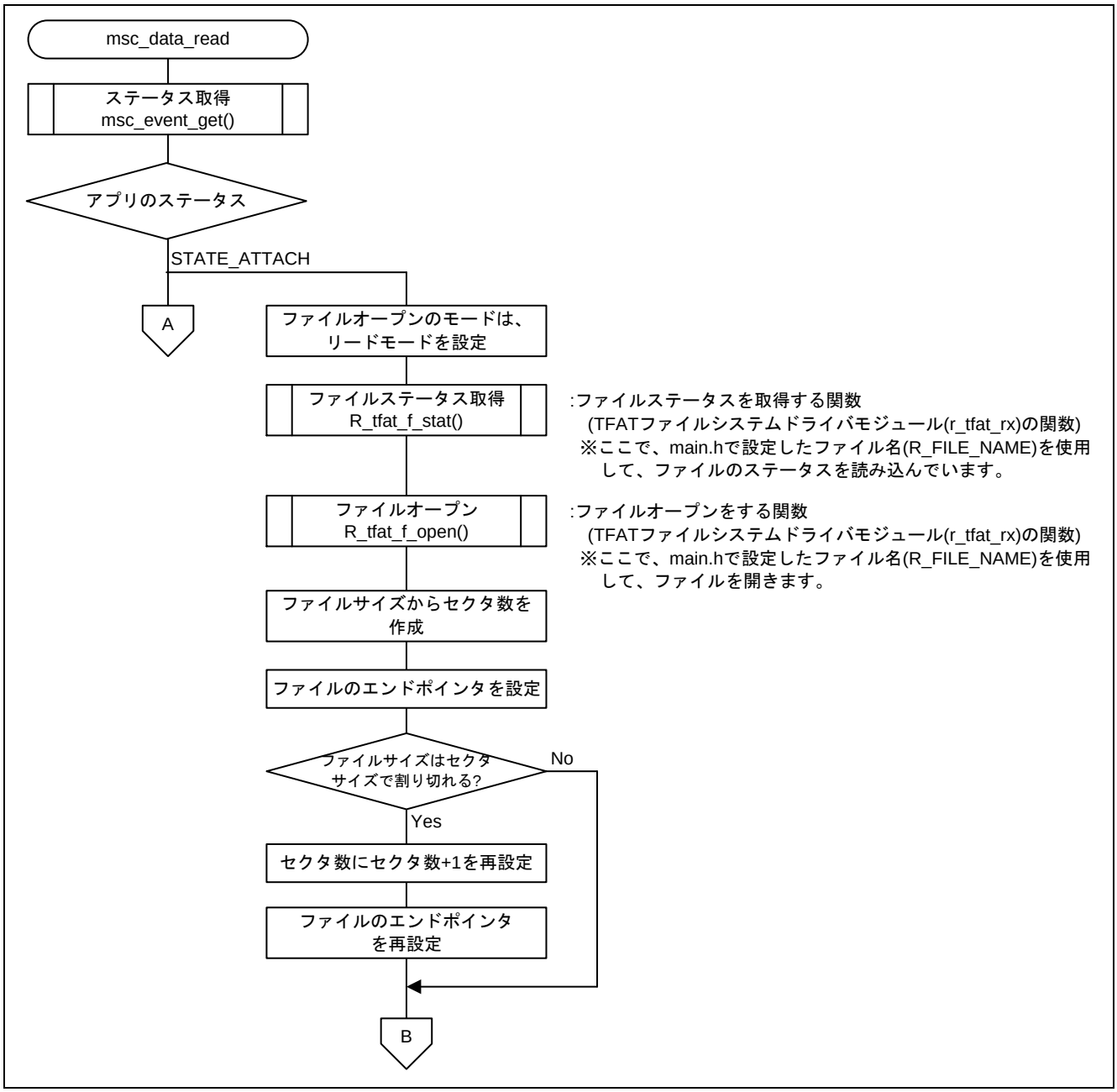


図 6.13 USB メモリからのデータリード、フラッシュメモリの書き換え処理(1)

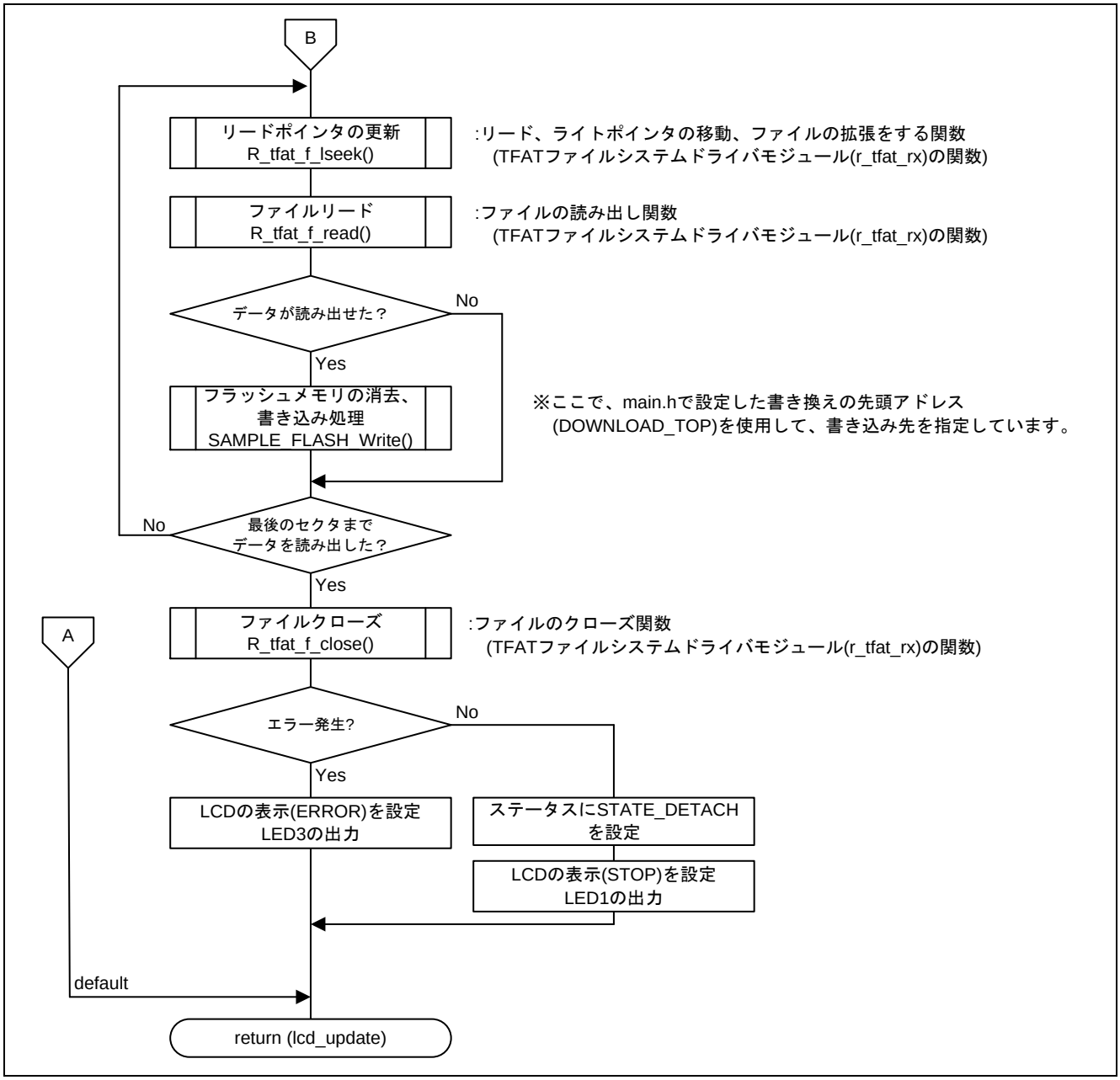


図 6.14 USB メモリからのデータリード、フラッシュメモリの書き換え処理(2)

(13) USB 切断

図 6.15にUSB 切断処理のフローチャートを示します。

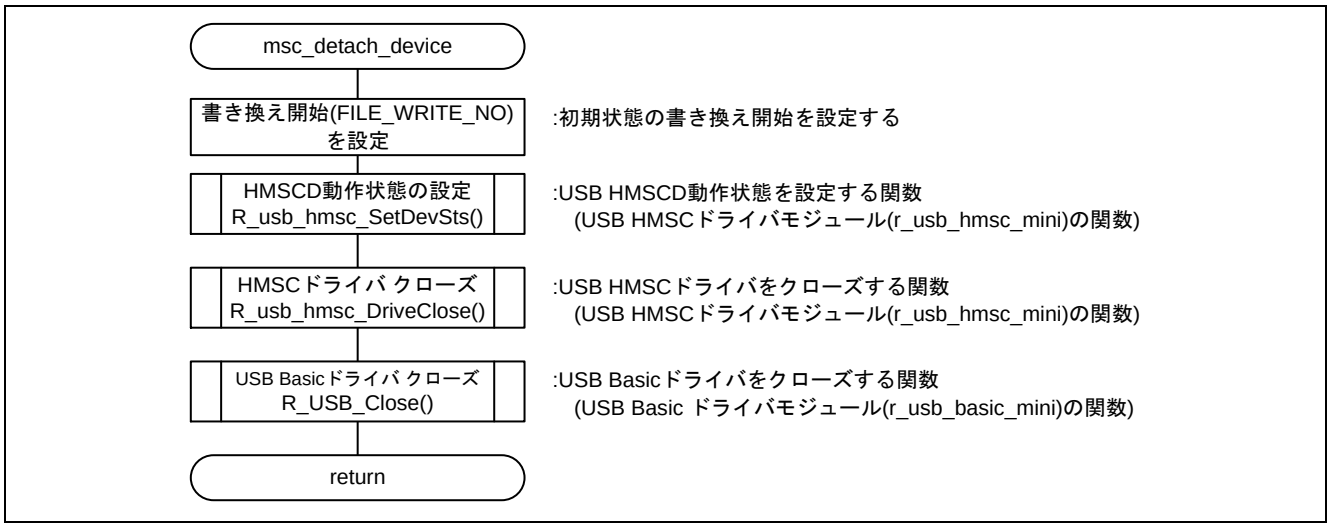


図 6.15 USB 切断処理

(14) USB ドライバコールバック処理

図 6.16にUSB ドライバコールバック処理のフローチャートを示します。

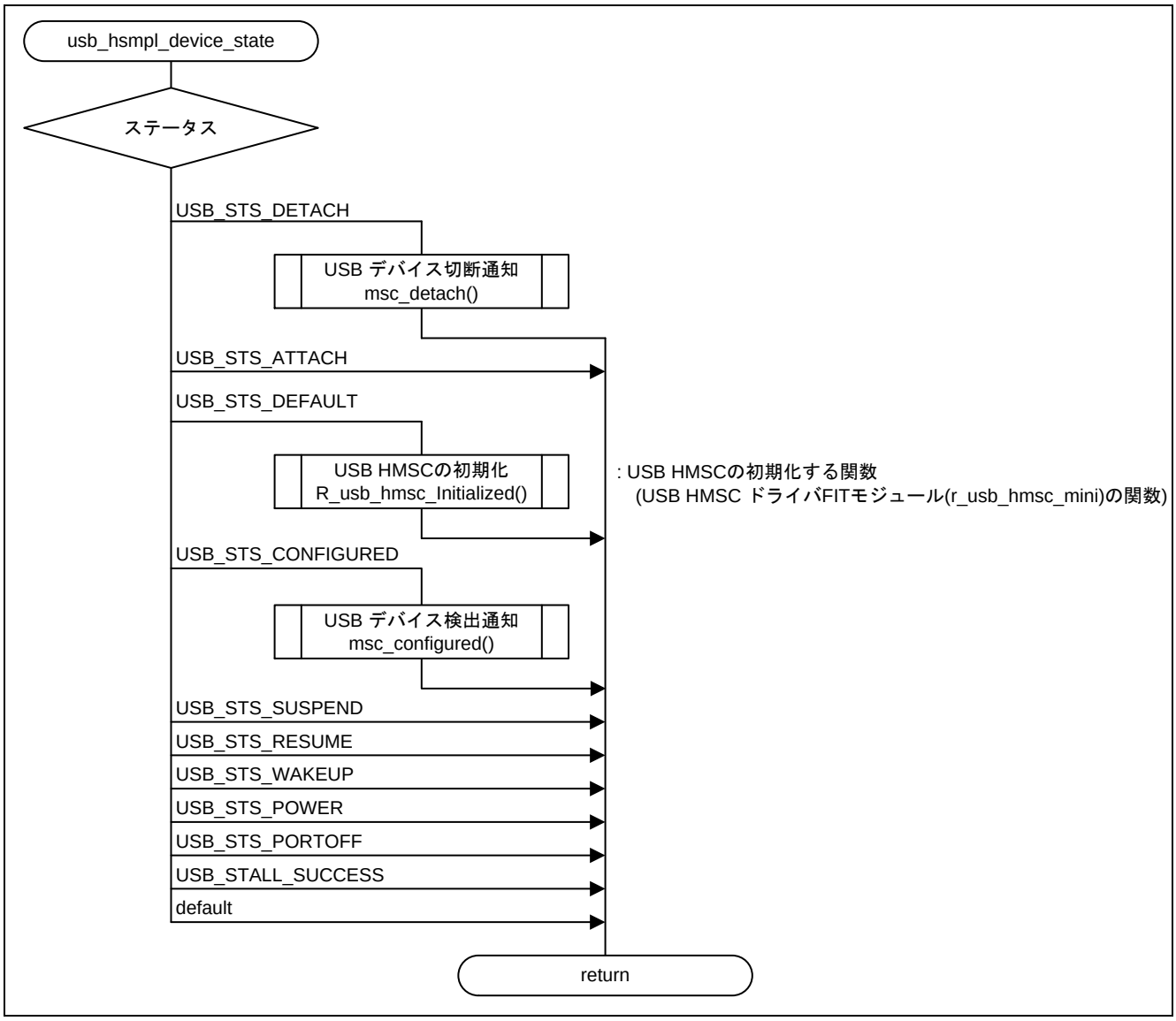


図 6.16 USB ドライバコールバック処理

(15) USB デバイス検出通知

図 6.17にUSB デバイス検出通知処理のフローチャートを示します。

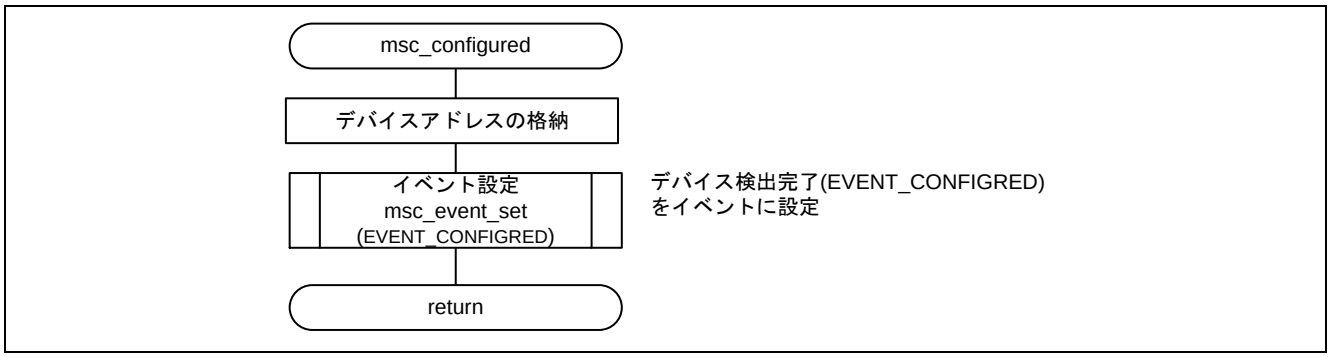


図 6.17 USB デバイス検出通知処理

(16) USB デバイス接続通知

図 6.18にUSB デバイス接続通知処理のフローチャートを示します。

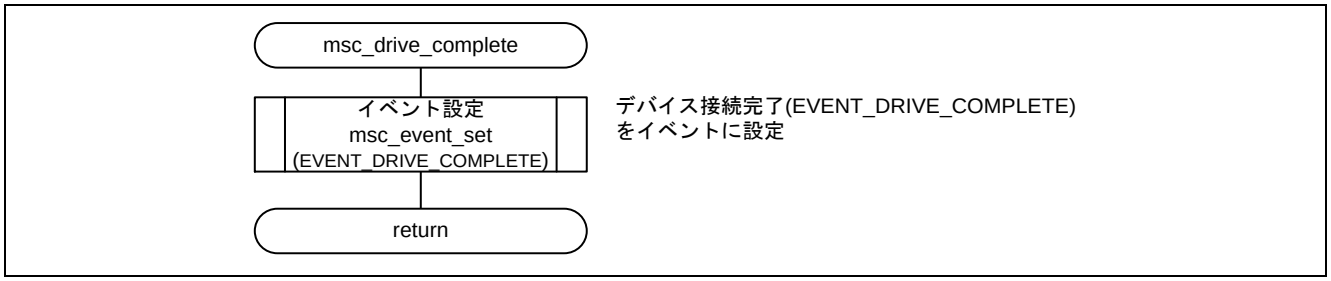


図 6.18 USB デバイス接続通知処理

(17) USB デバイス切断通知

図 6.19にUSB デバイス切断通知処理のフローチャートを示します。

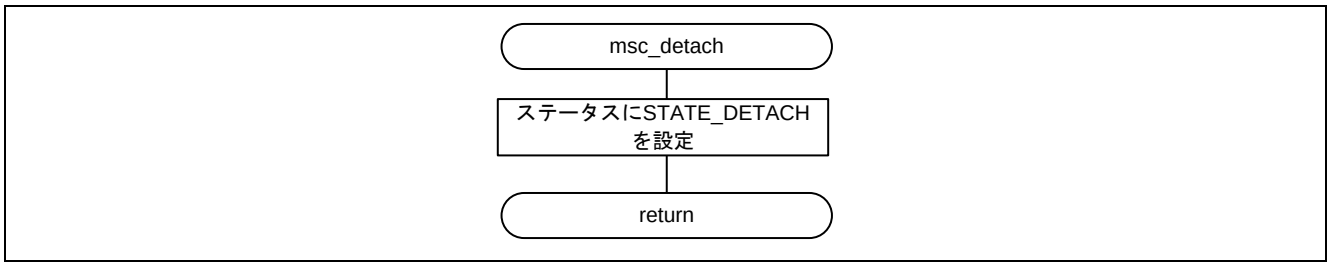


図 6.19 USB デバイス切断通知処理

(18) イベント設定

図 6.20にイベント設定処理のフローチャートを示します。

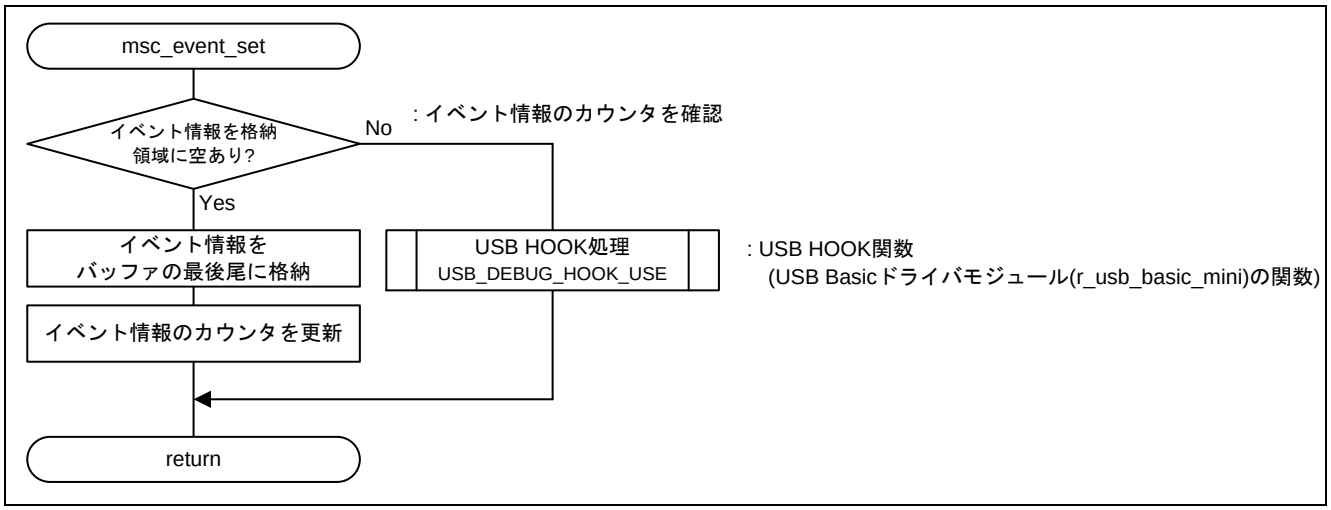


図 6.20 イベント設定処理

(19) イベント通知

図 6.21にイベント通知処理のフローチャートを示します。

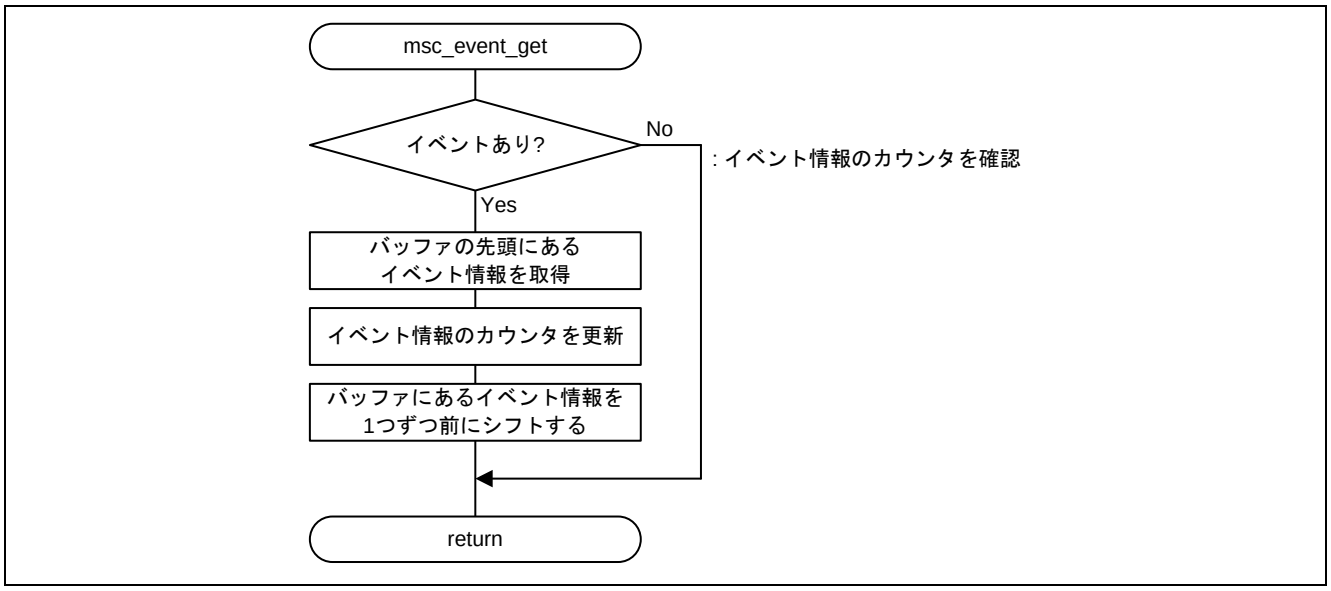


図 6.21 イベント通知処理

(20) フラッシュメモリの書き換え制御処理

図 6.22にフラッシュメモリの書き換え制御処理のフローチャートを示します。書き換えの最終アドレスは、この関数で設定しています。

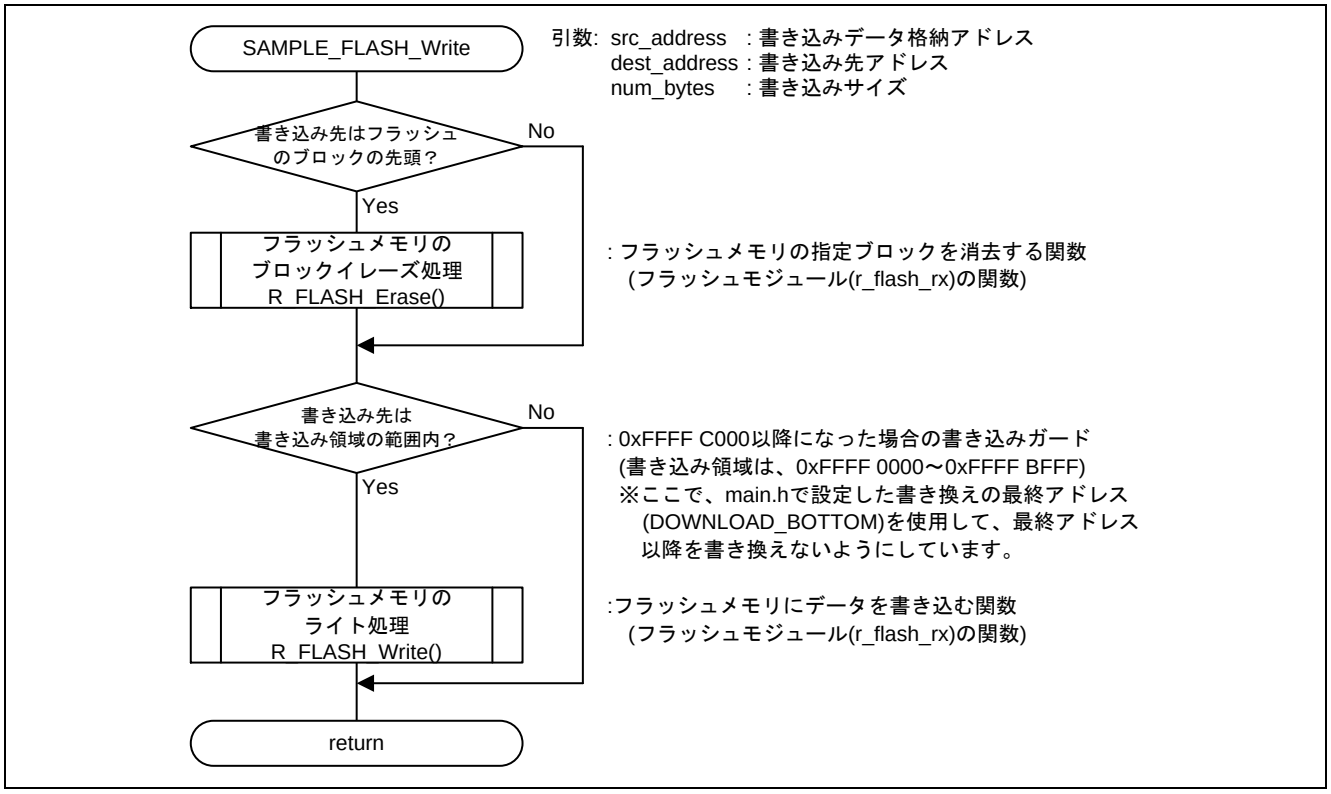


図 6.22　フラッシュメモリの書き換え制御処理

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/

お問合せ先
http://japan.renesas.com/contact/

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	2015.10.31	ー	新規発行
1.02	2016.02.29	4	1.3 モジュール構成　表 1.2 モジュール一覧　にて、 モジュール名 USB Basic Firmware の FIT モジュール名と USB ホストマストストレージクラス(USB HMSC) の FIT モジュール名を修正した。
		8	3.2 プロジェクトの作成　にて、 「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。」を追加した。
		11	3.3. プロジェクトのインポート　4.　において、 アイコンを追加し、図を修正した。
		15-16	3.4.1 コンフィギュレーションの変更　にて、 コード画面を修正した。
		25	4.3 プロジェクトのデバッグ　にて、 図 4.3 のタイトルを修正した。
		26	4.3 プロジェクトのデバッグ　4.　にて、 図を修正した。

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うと、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

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ものですが、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。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て、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事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が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/