RXファミリ

R01AN2237JJ0131
Rev.1.31
Mar 1, 2021

USB ホストマスストレージクラスドライバ (HMSC)による MSC デバイスとの USB 通信を行うサンプルプログラム Firmware Integration Technology

要旨

本資料は、USB Host Mass Storage Class Driver Firmware Integration Technology（以降、USB HMSC FIT モジュールと記述）を使用したサンプルファームウェアの説明資料です。以降、本サンプルファームウェアを HMSC と記述します。

実際のソフトウェア開発時には、必ず” USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート”(Document No:R01AN2025)および各マイコンのユーザーズマニュアル(ハードウェア編)と併用してご利用ください。また、必要に応じて "USB Host Mass Storage Class Driver (HMSC) Firmware Integration Technology アプリケーションノート(Document No:R01AN2026)”も参照してください。なお、USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート”(Document No:R01AN2025)は、パッケージ内の"reference_documents"フォルダにあります。

対象デバイス

RX65N/RX651 グループ
RX64M グループ
RX71M グループ
RX66T グループ
RX72T グループ
RX72M グループ
RX66N グループ
RX72N グループ
RX671 グループ

本プログラムは Renesas Starter Kits (RSK)を使って動作確認を行っています。

目次

1. はじめに 2
2. ソフトウェア構成 4
3. セットアップ 5
4. サンプルアプリケーション 9
5. クラスドライバ概要 13
6. RI600V4プロジェクトをCS+で使用する場合 14
7. e² studio用プロジェクトをCS+で使用する場合 20

1. はじめに

1.1 機能概要

HMSC は、USB マスストレージクラスの Bulk-Only Transport（BOT）プロトコルで構築されています。ファイルシステム、ストレージデバイスドライバと組み合わせることで BOT 対応の USB マスストレージデバイスと通信が可能です。

HMSC が提供する機能を以下に示します。

- USB マスストレージクラスの BOT に対応。
- USB マスストレージサブクラスの SFF-8070i に対応。
- 1 本のパイプを IN/OUT で共用。

1.2 HMSC 構成要素

HMSC は以下の FIT モジュールとサンプルアプリケーションで構成されています。

Table 1-1　HMSC 構成要素

FIT モジュール名	フォルダ名
RX Family Board Support Package Module Firmware Integration Technology	r_bsp
RX Family USB Basic Host and Peripheral Driver Firmware Integration Technology	r_usb_basic
RX Family USB Host Mass Storage Class Driver (HMSC) Firmware Integration Technology	r_usb_hmsc
RX Family DTC Module Firmware Integration Technology	r_dtc_rx
RX Family DMA Controller DMACA Control Module Firmware Integration Technology	r_dmaca_rx
RX Family Open Source FAT File System [M3S-TFAT-Tiny] Module Firmware Integration Technology	r_tfat_rx
RX Family M3S-TFAT-Tiny Memory Driver Interface Module Firmware Integration Technology	r_tfat_driver_rx

各 FIT モジュールの詳細は、関連ドキュメントを参照してください。また、本サンプルファームウェアで使用している FIT モジュールの最新バージョンは下記のホームページよりダウンロードが可能です。

ルネサスエレクトロニクスホームページ http://japan.renesas.com/

1.3 注意事項

本ドライバは、USB 通信動作を保証するものではありません。システムに適用される場合は、お客様における動作検証はもとより、多種多様なデバイスに対する接続確認を実施してください。

1.4　動作確認済環境

動作確認環境を以下に示します。

Table 1-2　動作確認環境

項目	内容
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V.3.03.00
	コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
リアルタイム OS	FreeRTOS V.10.0.0 RI600V4
エンディアン	リトルエンディアン / ビッグエンディアン
モジュールのリビジョン	Rev.1.31
使用ボード	Renesas Starter Kits for RX64M Renesas Starter Kits for RX71M Renesas Starter Kits for RX65N, Renesas Starter Kits for RX65N-2MB Renesas Starter Kits for RX72T Renesas Starter Kits for RX72M Renesas Starter Kits for RX72N Renesas Starter Kits for RX671

1.5　用語一覧

APL	:	Application program
BOT	:	Mass Storage class Bulk Only Transport
FSL	:	FAT File System Library
HCD	:	Host Control Driver for USB-BASIC-FW
HDCD	:	Host Device Class Driver (Device driver and USB class driver)
MGR	:	Peripheral Device State Manager for HCD
MSC	:	Mass Storage Class
Non-OS	:	USB Driver for OS-less
RSK	:	Renesas Starter Kits
RTOS	:	USB Driver for the real-time OS
TFAT	:	Tiny FAT file System Software for microcontrollers (M3S-TFAT-Tiny-RX)
USB-BASIC-FW	:	USB Basic Host and Peripheral Driver

2.　ソフトウェア構成

2.1　モジュール構成

HDCD(ホストデバイスクラスドライバ)は HMSDD（ホストマスストレージデバイスドライバ）と HMSCD（USB ホストマスストレージクラスドライバ）の総称です。

Figure 2-1に HMSC のモジュール構成図、**Table 2-1**にモジュール機能概要を示します。

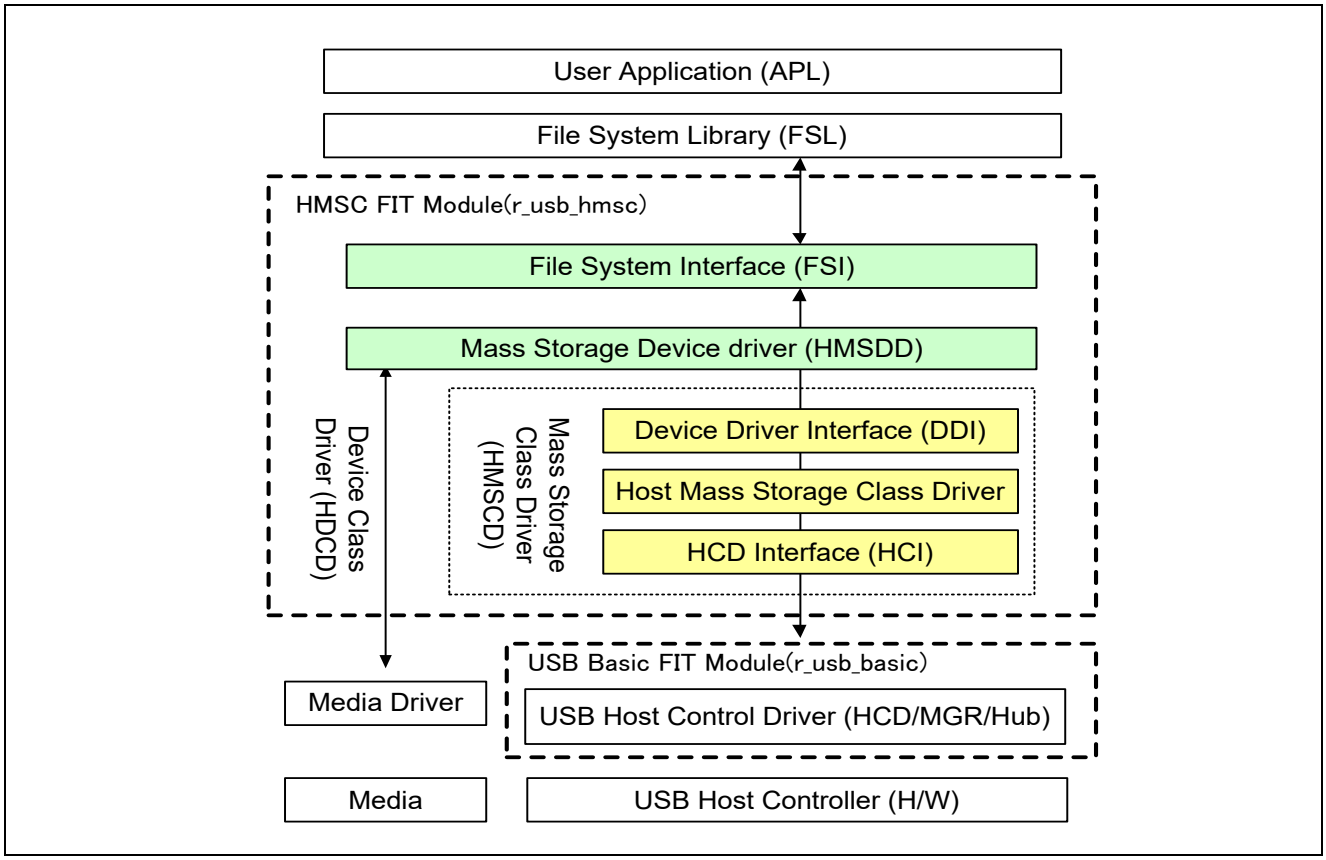


Figure 2-1 モジュール構成図

Table 2-1　モジュール機能概要

モジュール名	機能概要
APL	サンプルアプリケーションプログラム ・　FSL 関数を呼び出してストレージ機能を実現する
FSL	ユーザ仕様の FAT ファイルシステム
FSI	FSL-HMSDD 間のインターフェース関数
HMSDD	マスストレージデバイスドライバ
DDI	HMSDD-HMSCD 間のインターフェース関数
HMSCD	マスストレージクラスドライバ ・　ストレージコマンドに BOT プロトコルを付加して HCD に要求を行う ・　BOT のシーケンス管理を行う
HCI	HMSCD-HCD 間のインターフェース関数
HCD/MGR/Hub	USB Basic Driver
Media Driver	USB でないストレージ機器ドライバ

3. セットアップ

3.1 ハードウェア

3.1.1 動作環境例

HMSC の動作環境例をFigure 3-1に示します。評価ボードのセットアップ、エミュレータなどの使用方法については各取扱説明書を参照ください。

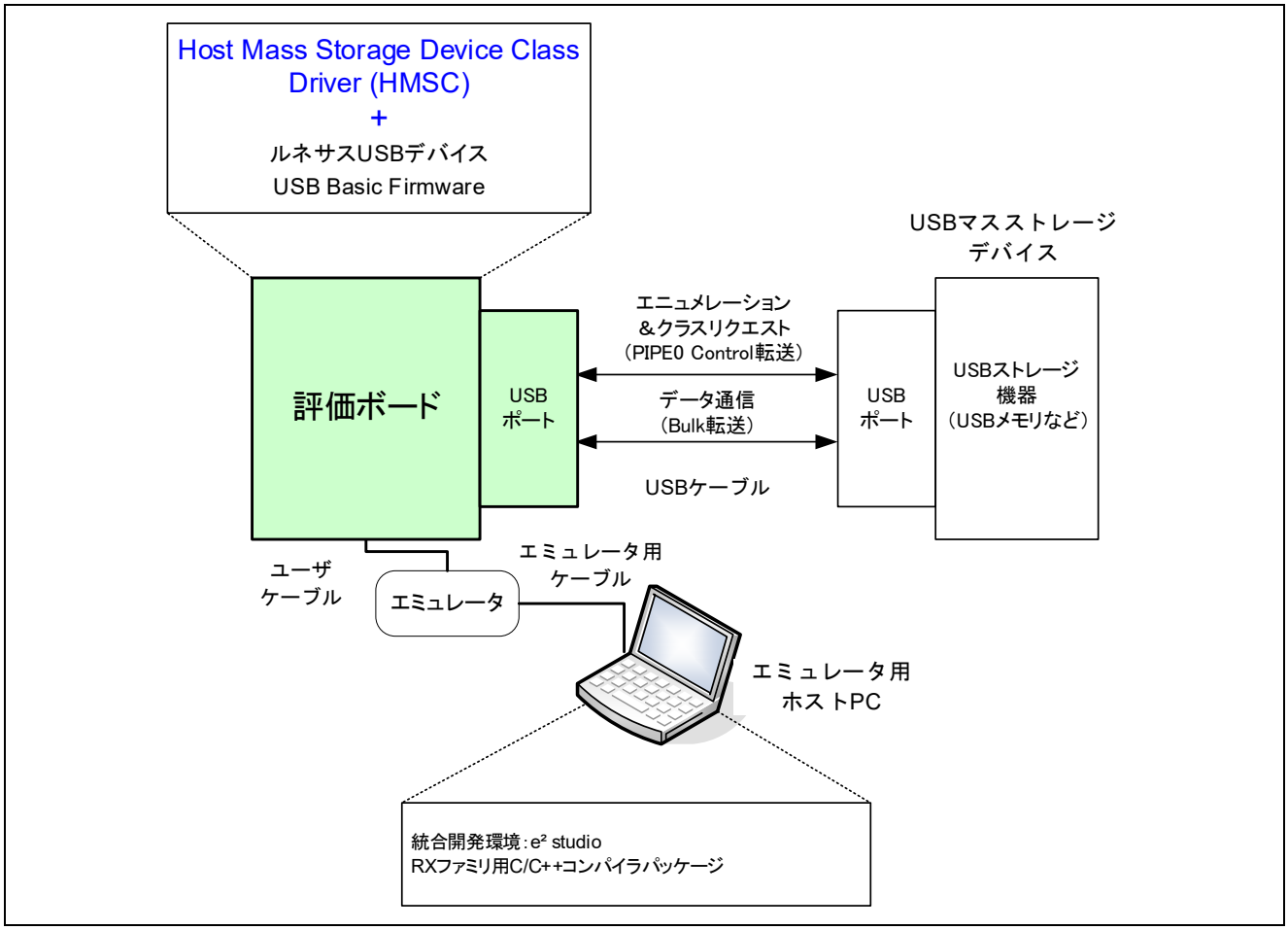


Figure 3-1 動作環境例

動作確認済みの評価ボードをTable 3-1に示します。

Table 3-1 HMSC 動作確認済みの評価ボード

マイコン	評価ボード
RX65N	RSK+RX65N, RSK+RX65N-2MB
RX64M	RSK+RX64M
RX71M	RSK+RX71M
RX72T	RSKRX72T
RX72M	RSK+RX72M
RX72N	RSK+RX72N
RX671	RSK+RX671

3.1.2 RSK 設定

RSK を USB Host モードに設定する必要があります。設定内容は以下を参照してください。

Table 3-2 **RSK 設定**

RSK	ジャンパ設定
RSK+RX65N	J8: Shorted Pin1-2
RSK+RX65N_2MB	J7: Shorted Pin1-2 J16: Shorted Pin2-3
RSK+RX64M (USB0)	J2: Shorted Pin1-2 J6: Shorted Pin2-3
RSK+RX64M (USBH)	J7: Shorted Pin1-2 J9: Shorted Pin2-3
RSK+RX71M (USB0)	J1: Shorted Pin1-2 J3: Shorted Pin2-3
RSK+RX71M (USBA)	J4: Shorted Pin1-2 J7: Shorted Pin2-3
RSKRX72T	J13: Shorted Pin1-2
RSK+RX72M	J8: Shorted Pin2-3 J10: Shorted Pin2-3
RSK+RX72N	J7: Shorted Pin2-3 J8: Shorted Pin2-3
RSK+RX72N	J7: Shorted Pin2-3 J8: Shorted Pin2-3
RSK+RX671	J8: Shorted Pin2-3 J13: Shorted Pin2-3

[Note]

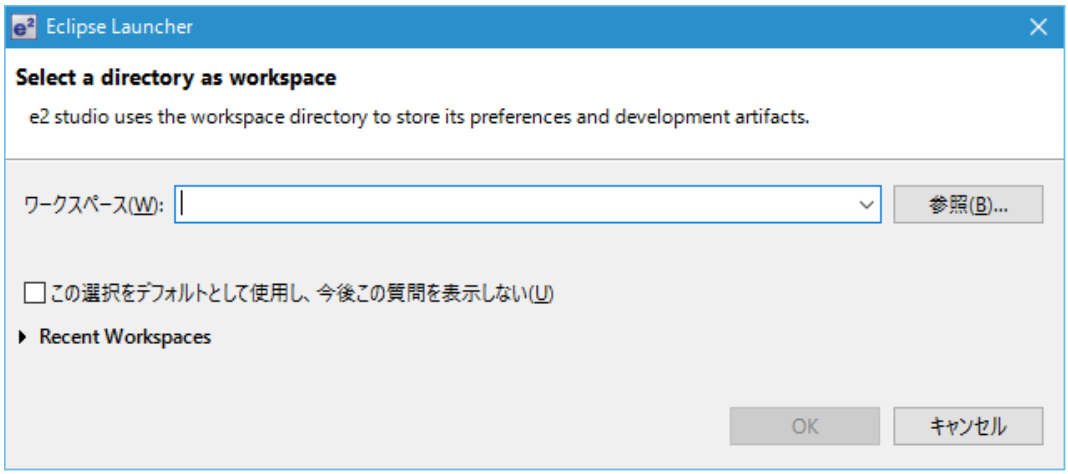
RSK 設定の詳細については、RSK のユーザーズマニュアルを参照してください。

3.2　ソフトウェア

(1). e² studio を起動

a)　e² studio を起動してください。

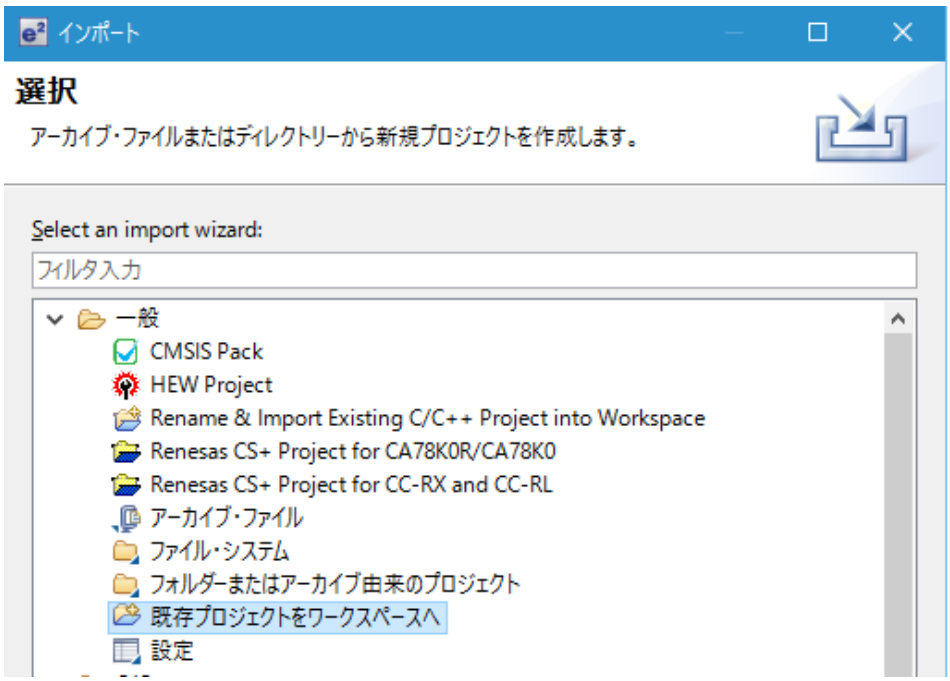
b)　はじめて e² studio を起動する場合、Eclipse Launcher ダイアログが表示されますので、プロジェクトを格納するためのフォルダを指定してください。



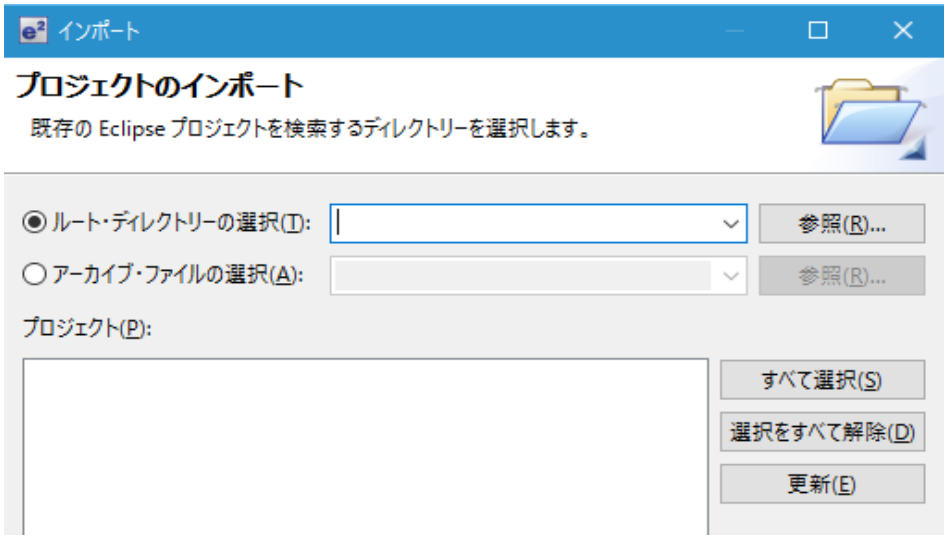
(2). プロジェクトをワークスペースへインポート

a)　[ファイル] --> [インポート]を選択してください。

b)　[一般] => [既存プロジェクトをワークスペースへ]を選択してください。



c)　プロジェクトファイル".cproject"が格納されたフォルダを”Select root directory”に入力してください。



d)　“Finish”をクリック

プロジェクトのワークスペースへのインポートが完了しました。同様の方法で他のプロジェクトを同一のワークスペースへインポートすることができます。

(3). “Build”ボタンをクリックし、実行プログラムを生成してください。

(4). デバッガへの接続を行い、実行プログラムをダウンロードしてください。“Run”ボタンをクリックすると、プログラムが実行されます。

4. サンプルアプリケーション

4.1 アプリケーション仕様

APL の主な機能を以下に示します。

1. MSC デバイスに対するエニュメレーション処理およびドライブ認識処理を行います。
2. 上記1完了後、ファイル'hmscdemo.txt'を一回、MSC デバイスに対し書き込みます。
3. 上記書き込み後、ファイル'hmscdemo.txt'のリード処理を繰り返します。

[Note]

ファイル'hmscdemo.txt'が格納された MSC デバイスを接続した場合、'hmscdemo.txt'が上書きされます。

4.2 アプリケーション処理概要 (Non-OS)

APL は、初期設定、メインループの 2 つの部分から構成されます。以下にそれぞれの処理概要を示します。

4.2.1 初期設定

初期設定では、MCU の端子設定、USB コントローラの初期設定およびアプリケーションプログラムの初期化処理を行います。

4.2.2 メインループ

このメインループでは、R_USB_GetEvent 関数の戻り値とステート変数を使ってプログラム制御を行っています。USB のイベント管理(アタッチ/デタッチ等)を、R_USB_GetEvent 関数の戻り値により行い、MSC デバイスに対するファイルライト/ファイルリードをステート変数の状態参照により行います。

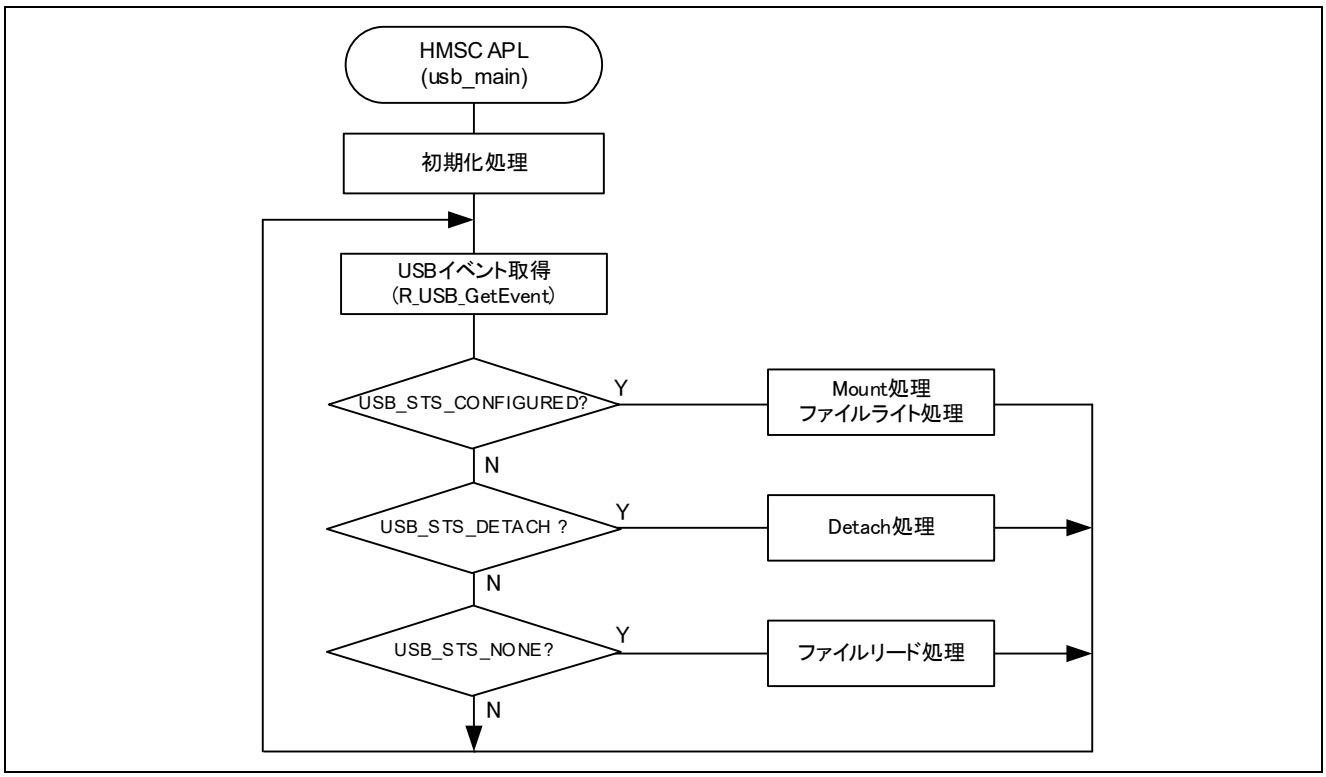


Figure 4-1 メインループ処理

1. Mount 処理 / ファイルライト処理 (USB_STS_CONFIGURED)

RSK に MSC デバイスが ATTACH され、Enumeration およびドライブ認識処理完了後に R_USB_GetEvent 関数をコールすると戻り値に USB_STS_CONFIGURED がセットされます。アプリケーションプログラムでは、戻り値に USB_STS_CONFIGURED がセットされていることを確認すると、Mount 処理およびファイルライト処理を行った後、ステート管理用変数に対し STATE_FILE_READ を設定します。

2. ファイルリード処理 (USB_STS_NONE)

USB 関連のイベントが無い状態で、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_NONE がセットされます。アプリケーションプログラムでは、戻り値に USB_STS_NONE がセットされていることを確認すると以下のファイルリード処理を行います。

(1). ファイルリード処理

FAT API を使って MSC デバイスに対しファイルリード処理を行います。MSC デバイスがデタッチされるまで MSC デバイスに対するファイルリード処理が継続されます。

3. Detach 処理 (USB_STS_DETACH)

RSK から MSC デバイスが DETACH された後、R_USB_GetEvent 関数をコールすると戻り値に USB_STS_DETACH がセットされます。アプリケーションプログラムでは、戻り値に USB_STS_DETACH がセットされていることを確認すると、ステート管理用変数に対し STATE_DETACH を設定します。

4.3 アプリケーション処理概要 (RTOS)

APL は、2 つのタスクで構成されています。以下にそれぞれの処理概要を示します。

4.3.1 usb_apl_task

1. RSK 起動後、MCU の端子設定、USB コントローラの初期設定およびアプリケーションプログラムの初期化処理を行います。
2. RSK に MSC デバイスが ATTACH され、Enumeration およびドライブ認識処理が完了すると、USB ドライバはコールバック関数(usb_apl_callback)をコールします。コールバック関数(usb_apl_callback)では、リアルタイム OS の機能を使って USB 完了イベントをアプリケーションタスクに通知します。
3. アプリケーションタスクではコールバック関数から通知された USB 完了イベント等の情報をリアルタイム OS の機能を使って取得します。
4. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_CONFIGURED の場合、APL はこの USB 完了イベントをもとに、MSC デバイスの Mount 処理および MSC デバイスに対するファイルライト処理を行います。
5. 上記2で取得した USB 完了イベント(usb_ctrl_t 構造体:メンバ event)が USB_STS_DETACH の場合、APL は、ステート管理用変数の初期化を行います。

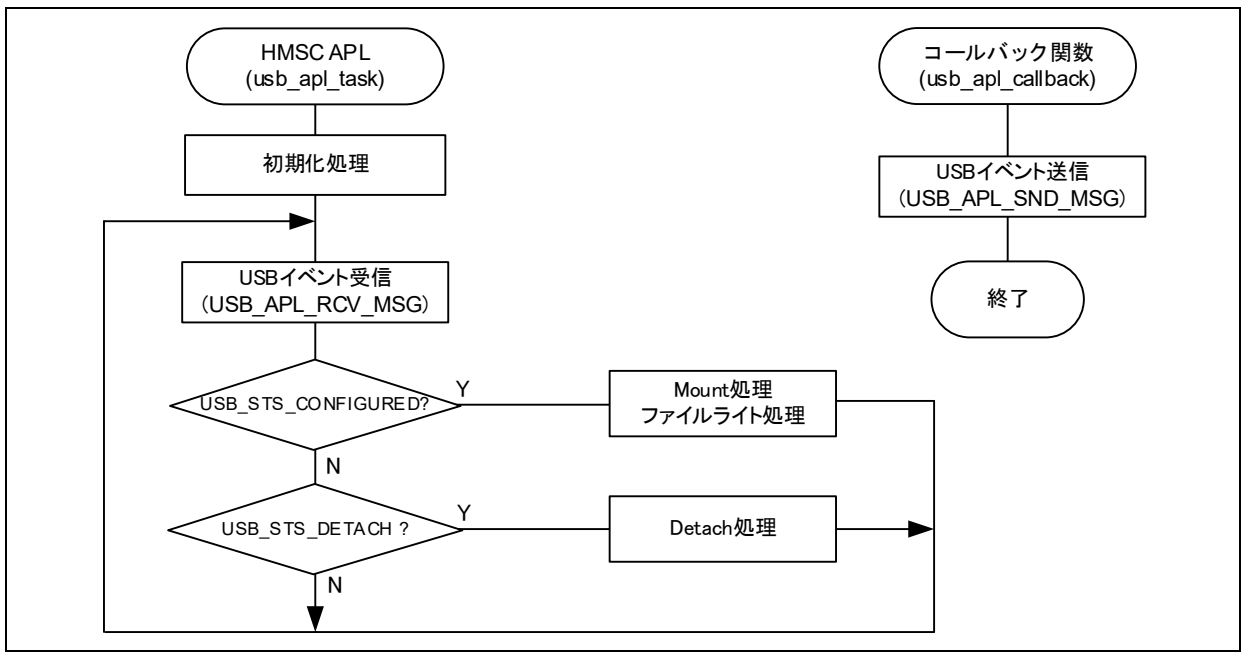


Figure 4-2 usb_apl_task

4.3.2 file_read_task

アプリケーションタスク(usb_apl_task と file_read_task)のうち、usb_apl_task が待ち状態の間、file_read_task が処理されます。このタスクでは、MSC デバイスに書き込まれたファイル(hmscdemo.txt)のファイルリード処理を行います。

4.4 アプリケーションプログラム用コンフィグレーションファイル (r_usb_hmsc_apl_config.h)

以下の各定義に対する設定を行ってください。

1. USE_USBIP 定義

使用する USB モジュールのモジュール番号を指定してください。

```
#define   USE_USBIP   USE_USBIP0                  // USB0 モジュールを使用する場合
#define   USE_USBIP   USE_USBIP1                  // USB1 モジュールを使用する場合
#define   USE_USBIP   (USE_USBIP0|USE_USBIP1)     // USB0 と USB1 モジュールを使用する場合
```

2. USB_SUPPORT_MULTI 定義

USB Hub 等を使用し、同時に複数の MSC デバイスを接続する場合、USB_SUPPORT_MULTI 定義に対し、USB_APL_ENABLE を指定してください。複数の MSC デバイスを接続しない場合は、USB_SUPPORT_MULTI 定義に対し USB_APL_DISABLE を指定してください。

```
#define   USB_SUPPORT_MULTI   USB_APL_DISABLE   // MSC デバイスの複数接続なし
#define   USB_SUPPORT_MULTI   USB_APL_ENABLE    // MSC デバイスの複数接続あり
```

3. USB_SUPPORT_RTOS 定義

リアルタイム OS を使用するかどうかを指定します。リアルタイム OS を使用する場合は、USB_SUPPORT_RTOS 定義に対し USB_APL_ENABLE を指定してください。

```
#define   USB_SUPPORT_RTOS   USB_APL_DISABLE   // RTOS 非使用
#define   USB_SUPPORT_RTOS   USB_APL_ENABLE    // RTOS 使用
```

4. 注意事項

上記はアプリケーションプログラム用のコンフィグレーション設定です。上記の設定の他に USB ドライバのコンフィグレーション設定が必要です。USB ドライバのコンフィグレーション設定については、「USB Basic Host and Peripheral Driver Firmware Integration Technology アプリーションノート」(Document No. R01AN2025JJ)を参照してください。

4.5 複数の MSC デバイスを接続する場合

USB Hub 等を使用し、複数の MSC デバイスを接続するアプリケーションプログラムを開発する場合は、以下の参考プログラムを参照してください。

1. r_usb_hmsc_apl_multi.c

5. クラスドライバ概要

5.1 クラスリクエスト

Table 5-1に HMSC が対応しているクラスリクエストを示します。

Table 5-1 クラスリクエスト

リクエスト	説明
GetMaxLun	サポートする最大ユニット数を取得する。
MassStrageReset	プロトコルエラーを解除する。

5.2 ストレージコマンド

HMSC は USB マスストレージデバイスへアクセスするために必須となるストレージコマンドと、サンプルのストレージコマンドに対応しています。

HMSC は USB マスストレージクラスの SFF-8070i に対応しています。

Table 5-2　サポートするストレージコマンド

コマンド名	コード	説明	対応（※）
TEST_UNIT_READY	0x00	ペリフェラル機器の状態確認	○
REQUEST_SENSE	0x03	ペリフェラル機器の状態取得	○
FORMAT_UNIT	0x04	論理ユニットのフォーマット	×
INQUIRY	0x12	論理ユニットのパラメータ情報取得	○
MODE_SELECT6	0x15	パラメータ指定	×
MODE_SENSE6	0x1A	論理ユニットのパラメータ取得	×
START_STOP_UNIT	0x1B	論理ユニットのアクセス許可／禁止	×
PREVENT_ALLOW	0x1E	メディアの取り出し許可／禁止	○
READ_FORMAT_CAPACITY	0x23	フォーマット可能な容量取得	○
READ_CAPACITY	0x25	論理ユニットの容量情報取得	○
READ10	0x28	データ読み出し	○
WRITE10	0x2A	データ書き込み	○
SEEK	0x2B	論理ブロックアドレスに移動	×
WRITE_AND_VERIFY	0x2E	確認付きデータ書き込み「	×
VERIFY10	0x2F	データ確認	×
MODE_SELECT10	0x55	パラメータ指定	○
MODE_SENSE10	0x5A	論理ユニットのパラメータ取得	○

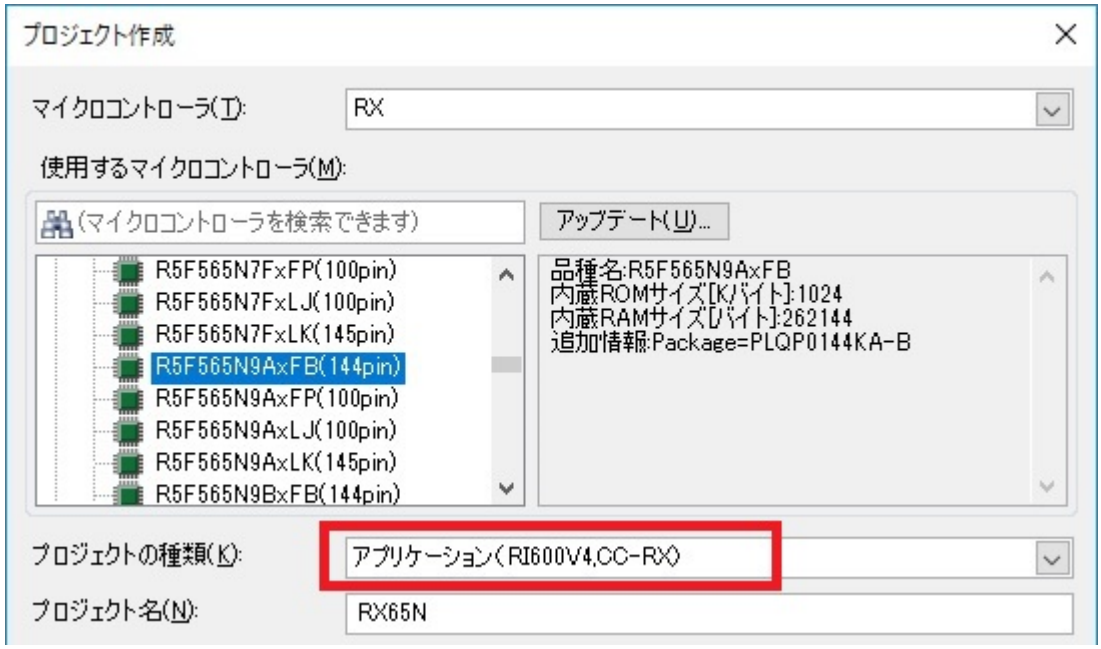
※　○:実装　×:未実装

6. RI600V4 プロジェクトを CS+で使用する場合

パッケージ内の RI600V4 用プロジェクトは CS+をサポートしていません。RI600V4 用プロジェクトを CS+で使用する場合、以下の手順に従って CS+用のプロジェクトを作成する必要があります。

6.1 CS+上で新規プロジェクトを作成

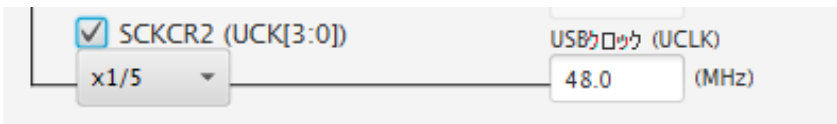
プロジェクトの種類には、「アプリケーション(RI600V4, CC-RX)」を選択してください。



6.2 スマートコンフィグレータを起動

1. クロック設定 (「クロック」タブを選択)

USB クロック(UCLK)に 48MHz が設定されるよう関連クロックを設定してください。



2. コンポーネント設定 (「コンポーネント」タブを選択)

(1). FIT モジュールをインポート

以下の FIT モジュールを選択し、「終了」ボタンを押してください

a. r_usb_hmsc
b. r_tfat_rx
c. r_tfat_driver_rx
d. r_sys_timer_rx
e. r_cmt_rx

Note:

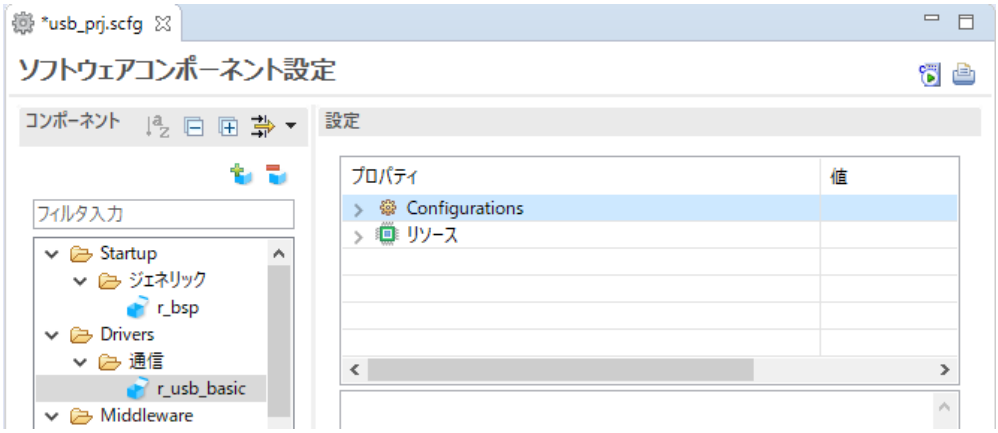
a. r_usb_basic モジュールも同時に組み込まれます。
b. DTC/DMA を使用する場合、r_dtc_rx/r_dmaca_rx モジュールも選択してください。

(2). コンフィグレーション

a. r_bsp

DTC 使用時、Heap size を変更してください。設定値は DTC FIT モジュールのドキュメントを参照してください。

b. r_usb_basic



(a). Configurations

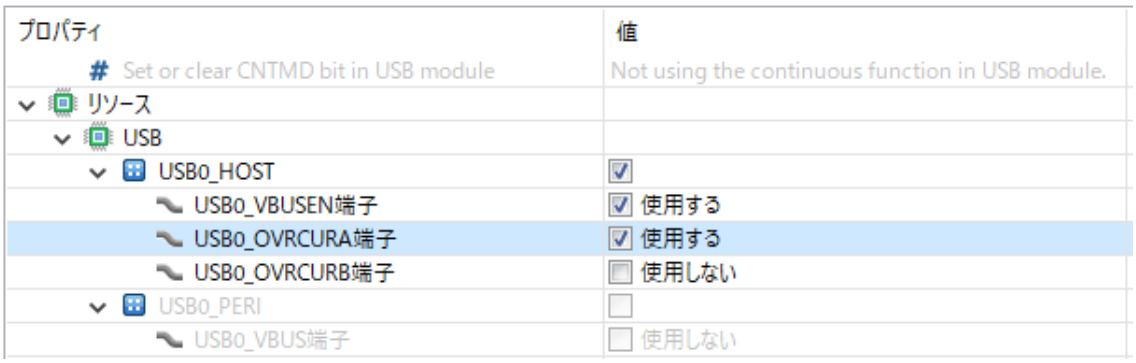
USB Basic Host and Peripheral Driver Firmware Integration Technology アプリケーションノート(ドキュメント No.R01AN2025)の「コンフィグレーション」章を参照いただきますようお願いします。

(b). リソース

以下の端子をチェックしてください。

i. USBx_VBUSEN 端子

ii. USBx_OVRCURA 端子または USBx_OVRCURB 端子



c. r_usb_tfat_driver_rx

(a). Number of USB drives

同時接続するMSCデバイスの数を設定してください。 (1から4まで)

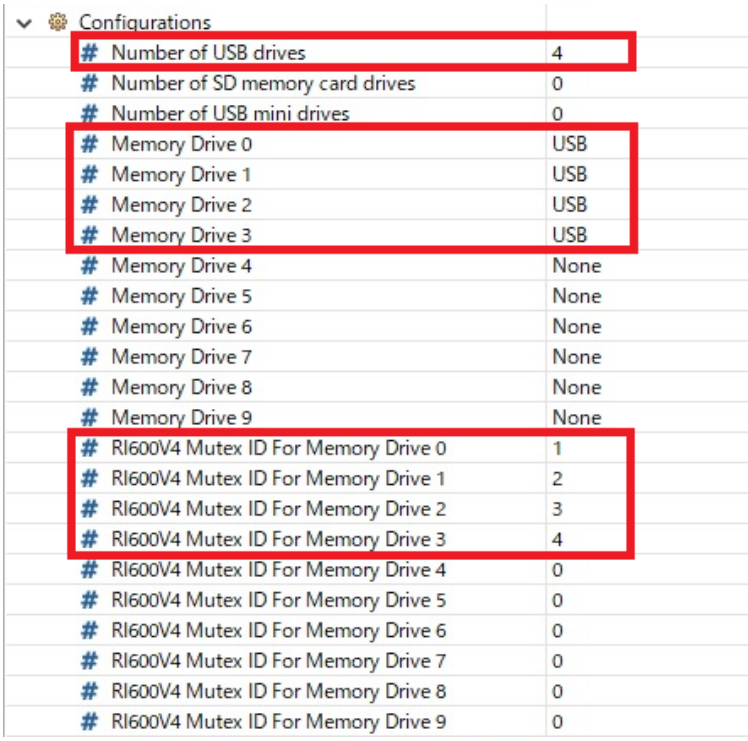
(b). Memory Drive

"Memory Driver 0"から順に"USB"を設定してください。"USB"を設定する数は上記(a)で設定した数と一致しなければなりません。

(c). RI600V4 Mutex ID For Memory Drive

RI600V4コンフィグレーションファイル(.cfgファイル)内で定義したMutex ID番号を設定してください。(1から4まで)

下記は、同時接続するMSCデバイス数が4の場合の設定例です。赤枠部分を同時接続するMSCデバイス数に応じて設定してください。



3.　端子設定 (「端子」タブを選択)

お客様のシステムに合った USB 端子のポート選択を行ってください。

4.　その他の設定

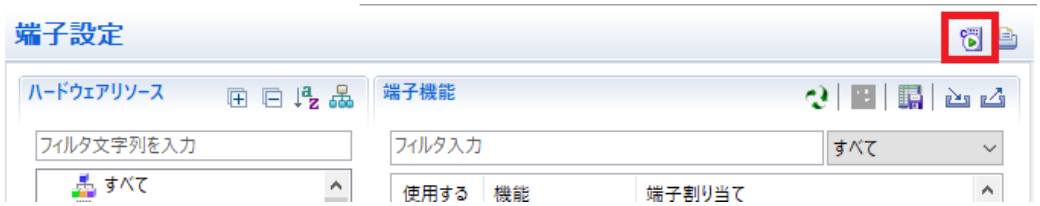
"r_tfat_rx¥src¥ffconf.h"ファイル内の「FF_VOLUMES」定義に対し、同時にサポート可能なドライブ数(1 から 4 まで)を設定してください。

例)

```
#define　　FF_VOLUMES　　4
```

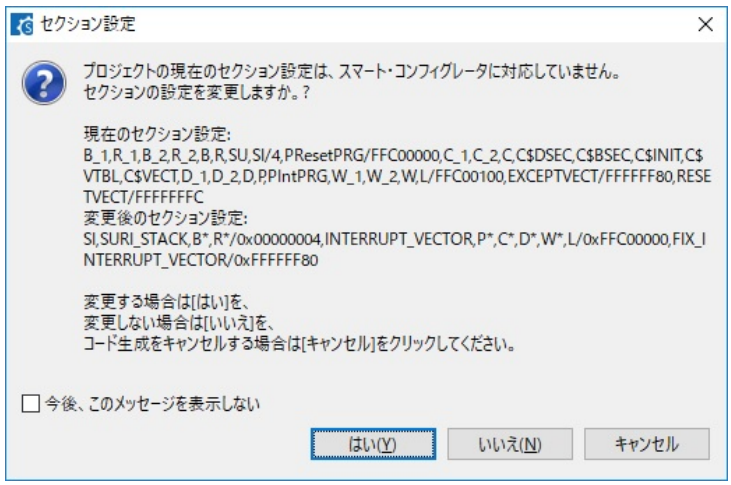
5. コード生成

「コードの生成」ボタンをクリックすると、スマートコンフィグレータは<ProjectDir>¥src¥smc_gen フォルダに USB FIT モジュールのソースコードおよび端子設定のコードを生成します。



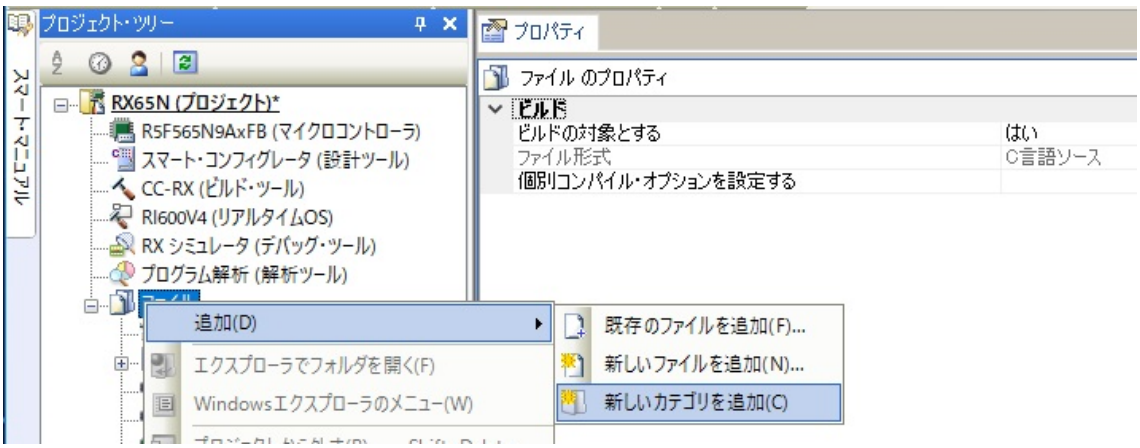
Note:

以下のダイアログが出力されますので、「はい(Y)」を選択してください。



6.3 アプリケーションプログラムおよびコンフィグレーションファイルの追加

1. 本パッケージ内の demo_src フォルダを<ProjectDir>¥src フォルダにコピーしてください。
2. 本パッケージ内の RI600V4 用コンフィグレーションファイル(.cfg ファイル)を<ProjectDir>フォルダにコピーしてください。
3. プロジェクトツリー内の「ファイル」を選択、右クリック。「追加」→「新しいカテゴリを追加」を選択し、アプリケーションプログラムを格納するカテゴリを作成してください。次に「既存のファイルを追加」を選択し、上記2でコピーしたアプリケーションプログラムおよびコンフィグレーションファイルを登録してください。



Note:

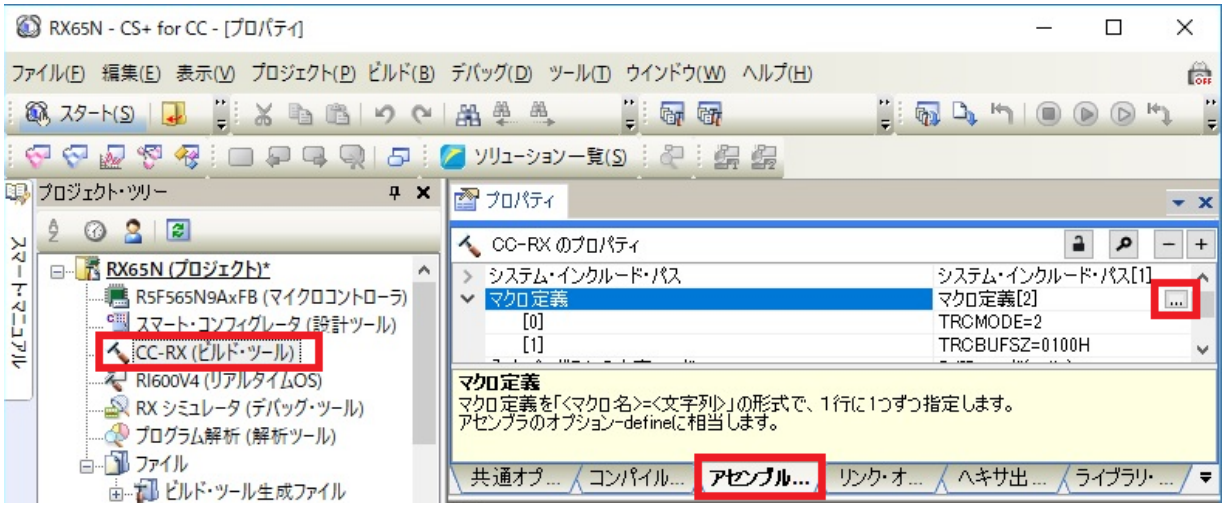
新規に生成された task.c ファイルおよび sample.cfg は<ProjectDir>フォルダから削除してください。

6.4 マクロ定義削除

新規生成したプロジェクトには以下のマクロが定義されていますので、これらのマクロ定義を削除してください。

CC-RX(ビルド・ツール) -> 「アセンブルオプション」タブを選択し、以下のマクロ定義を削除。

1. TRCMODE = 2
2. TRCBUFSZ = 0100H



6.5 ビルド実行

ビルドを実行し、実行プログラムを生成してください。

7. e² studio 用プロジェクトを CS+で使用する場合

HMSC のプロジェクトは、統合環境 e² studio で作成されています。HMSC を CS+で動作させる場合は、下記の手順にて読み込んでください。

[Note]

1. 「プロジェクト変換設定」ウィンドウ内の「変換直前のプロジェクト構成ファイルをまとめてバックアップする」のチェックを外してください。
2. RI600V4 をご使用の場合、以下の方法をサポートしていません。「**6. RI600V4プロジェクトをCS+で使用する場合**」を参照してください。

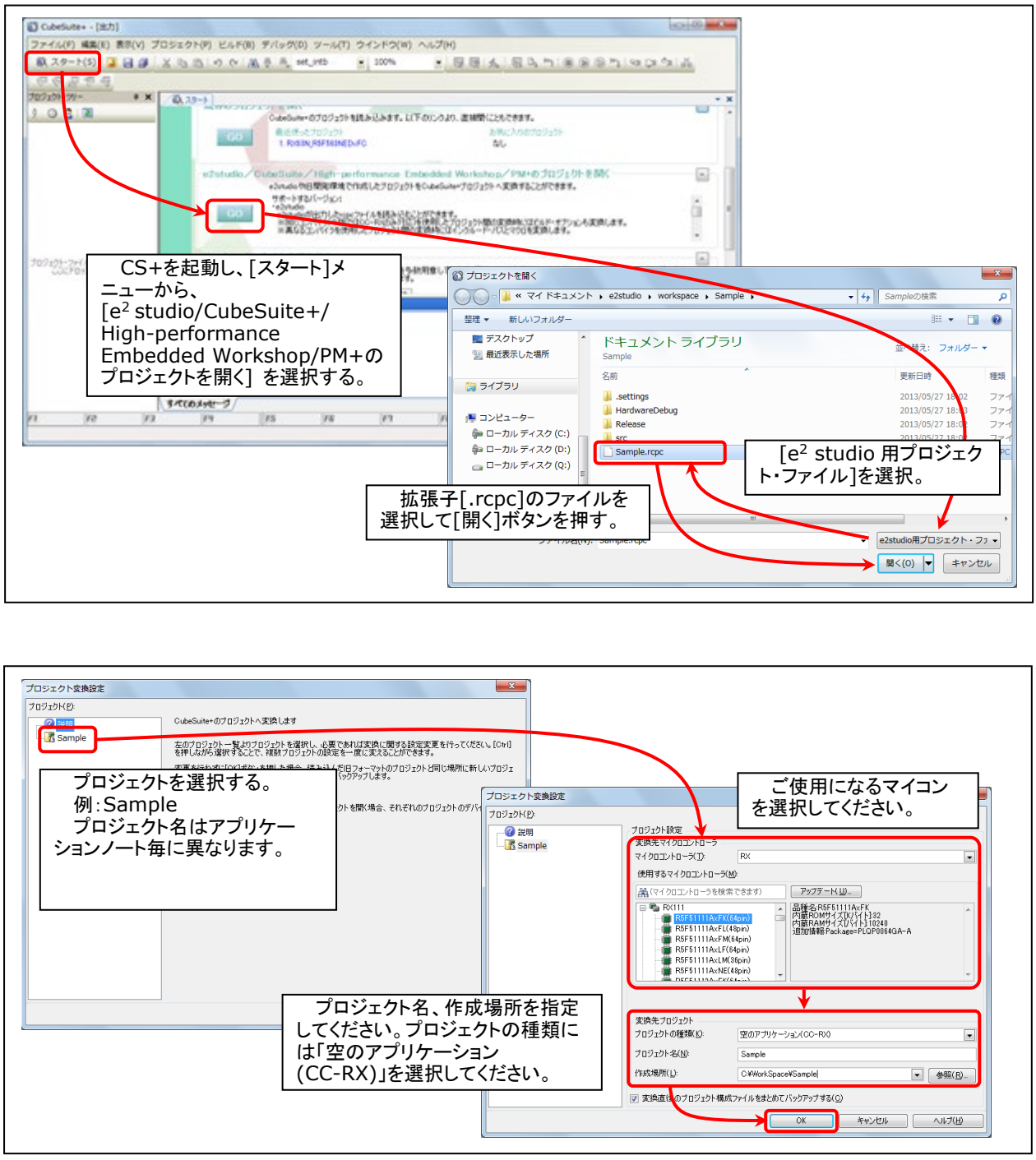


Figure 7-1　e² studio 用プロジェクトの CS+読み込み方法

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/

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	Oct 16, 2014	—	初版発行
1.10	Dec 26, 2014	—	RX71M を対象デバイスに追加。
1.11	Sep 30, 2015	—	RX63N/RX631 を対象デバイスに追加。
1.20	Sep 30, 2016	—	1. 対象デバイスに RX65N/RX651 を追加 2. DMA 転送をサポート 3. USB Host and Peripheral Interface Driver アプリケーションノートに対応
1.21	Mar 31, 2017	—	USB Basic driver のリビジョンを Up しました。
1.22	Sep 30, 2017	—	RX65N/RX651-2M をサポート
1.23	Mar 31, 2018	—	USB Basic driver のリビジョンを Up しました。
1.24	Dec 28, 2018	—	リアルタイム OS をサポート
1.25	Apr 16, 2019	—	RX66T/RX72T を対象デバイスに追加。
1.27	Jul 31, 2019	—	1. RX72M を対象デバイスに追加。 2. 対象デバイスから RX63N を削除しました。
1.30	Mar 1, 2020	—	1. リアルタイム OS(uITRON:RI600V4)をサポートしました。 2. 対象デバイスに RX72N/RX66N を追加
1.31	Mar 1, 2021	—	対象デバイスに RX671 を追加

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。

© 2021 Renesas Electronics Corporation. All rights reserved.