RX ファミリ

SSI モジュールを使用する PCM データ転送サンプルプログラム Firmware Integration Technology

要旨

本アプリケーションノートは、Firmware Integration Technology (FIT) に準拠した SSI モジュールの使い方を説明します。PCM データ転送を行う 7 つのサンプルプログラムを同梱しており、ルネサスの評価ボード上で PCM データ転送を行う処理を記述しています。

本書に添付したサンプルプログラムは、2 種類に分かれます。一方はシリアルサウンドインタフェースを使用するためのユースケース別の基本プログラムです。もう一方は基本プログラムの応用で RX72N Envision Kit 上でシリアルサウンドインタフェースを使用するものです。オンボードの MEMS マイクから PCM 信号を入力することや、正弦波を出力することができます。

シリアルサウンドインタフェースには SSI と SSIE の 2 種類があり、サンプルプログラムは両方に対応しています。

サンプルプログラムは便宜上、RX64M または RX72N を対象としていますが、統合開発環境 e^2 studio やスマート・コンフィグレータを活用して、他のルネサス製 MCU に容易に応用することができます。

本書では、最初にサンプルプログラムの概要、次に e^2 studio でのサンプルプログラムの使い方を説明します。

表 1 基本プログラム

転送処理の種類	サンプルプログラム	対象デバイス
割り込みによる PCM データ送信	rx64m_example_ssi_tx	RX64M
割り込みによる PCM データ受信	rx64m_example_ssi_rx	RX64M
割り込みによる PCM データ受信と送信	rx64m_example_ssi_rx_tx	RX64M
DMAC による PCM データ送信	rx64m_example_ssi_dma_tx	RX64M
DMAC による PCM データ受信	rx64m_example_ssi_dma_rx	RX64M
DMAC による PCM データ受信と送信	rx64m_example_ssi_dma_rx_tx	RX64M

表 2 応用プログラム

転送処理の種類	サンプルプログラム	対象デバイス
DMAC による PCM データ送信と送信	rx72n_example_ssi_dma_rx_tx	RX72N

対象デバイス（SSI）

- RX64M グループ

対象デバイス（SSIE）

- RX72N グループ

目次

1. 概要 3
1.1 添付のサンプルプログラムについて 4
1.1.1 基本プログラム 4
1.1.2 応用プログラム 6

2. 動作確認条件 7

3. ソフトウェア説明 11
3.1 ソフトウェアの構成要素 11
3.2 SSI0 の割り込みを使用する PCM データ転送 11
3.3 DMAC を使用する PCM データ転送 13
3.4 PCM データバッファ 14

4. スマート・コンフィグレータによるプロジェクトの移行 15
4.1 デバイスの変更 16
4.2 クロック設定 17
4.3 SSI の設定と SSI で使用する端子設定 18
4.4 端子割り当ての変更 19
4.5 その他の変更 20

5. サンプルプログラムの実行 21
5.1 ハードウェアのセットアップ 21
5.1.1 基本プログラムを使用する場合 21
5.1.2 応用プログラムを使用する場合 22
5.2 サンプルプログラムのインポート 23
5.2.1 e² studio での手順 23
5.2.2 CS+での手順 24

6. 参考資料 25

改訂記録 26

1. 概要

本書に添付した全てのサンプルプログラムは、典型的な PCM データ転送（送信、受信および送受信）における SSI モジュールの使用方法を説明するために作成しています。基本プログラムは転送処理に SSI0 の割り込み、または DMAC を使用します。応用プログラムは転送処理に DMAC を使用します。本書に添付した全てのサンプルプログラムは、すべて下記のフォーマットで PCM データを転送します。

PCM データ転送フォーマット

- PCM データ転送フォーマット：I^2S（SCK：32bit/ch）
- PCM データ：16bit ステレオ（基本プログラム）、24bit ステレオ（応用プログラム）
- クロックマスタモード

テスト信号

送信処理のサンプルプログラムは以下のテスト波形データを内蔵しており、基本プログラムでは SSI0 から送信され、応用プログラムでは SSIE1 から送信されます。

- Fs:48kHz、Lch：フルスケールの 3kHz 正弦波、Rch：フルスケールの 1.5kHz 正弦波

基本プログラムは、マスタクロックが 12.2880MHz のときに、サンプリング周波数が 48kHz になるように設定されています。

応用プログラムは、マスタクロックが 24.576MHz のときに、サンプリング周波数が 48kHz になるように設定されています。

1.1　添付のサンプルプログラムについて

本書とサンプルプログラムはファイル r01an2825xx****_ssi_rx.zip に格納されています。解凍すると以下のフォルダ構成となります。 ****は、本アプリケーションノートのリビジョンを示します。

```
r01an2825xx****_ssi_rx
        |
        + r01an2825ej****_rx.pdf（本書）
        + r01an2825jj****_rx.pdf（本書）
        + workspace
                |
                + rx64m_example_ssi_tx（基本プログラム）
                + rx64m_example_ssi_rx（基本プログラム）
                + rx64m_example_ssi_rx_tx（基本プログラム）
                + rx64m_example_ssi_dma_tx（基本プログラム）
                + rx64m_example_ssi_dma_rx（基本プログラム）
                + rx64m_example_ssi_dma_rx_tx（基本プログラム）
                + rx72n_example_ssi_dma_rx_tx（応用プログラム）
```

上記の 7 つのサンプルプログラムは e^2 studio のプロジェクトです。

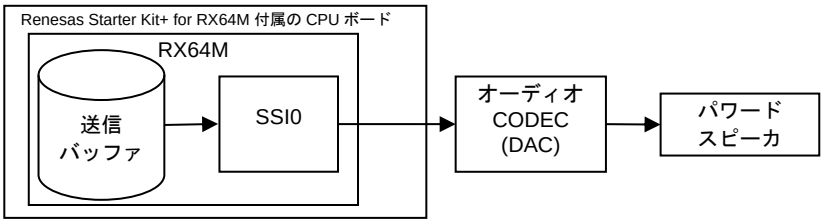
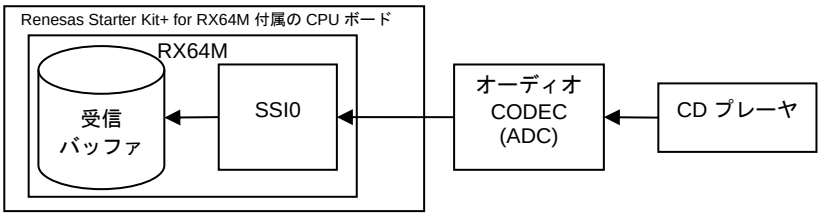
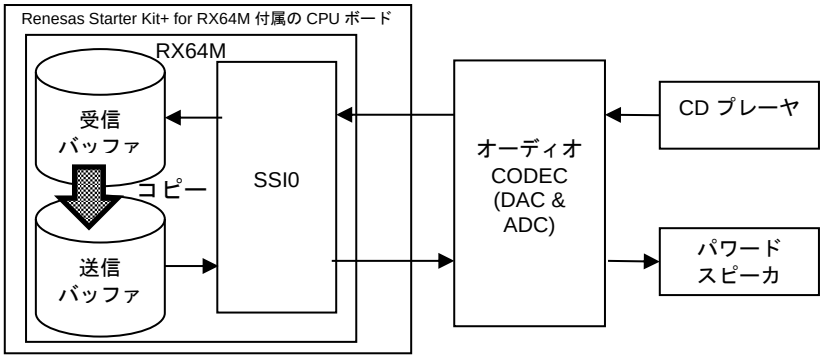
1.1.1　基本プログラム

PCM データ転送の処理の種類に対応するサンプルプログラムを表 1-1 に示します。これらのうち 1 つを選択し、e^2 studio でインポート、実行してください。想定するターゲットシステム上での各サンプルプログラムの処理とオーディオシグナルフローを表 1-2 に転送処理の種類別に示します。

表 1-1　転送処理とプロジェクトのファイルの対応（基本プログラム）

転送処理の種類	サンプルプログラム	対象デバイス
割り込みによる PCM データ送信	rx64m_example_ssi_tx	RX64M
割り込みによる PCM データ受信	rx64m_example_ssi_rx	RX64M
割り込みによる PCM データ受信と送信	rx64m_example_ssi_rx_tx	RX64M
DMAC による PCM データ送信	rx64m_example_ssi_dma_tx	RX64M
DMAC による PCM データ受信	rx64m_example_ssi_dma_rx	RX64M
DMAC による PCM データ受信と送信	rx64m_example_ssi_dma_rx_tx	RX64M

表 1-2　転送処理の種類ごとのオーディオシグナルフロー（基本プログラム）

転送処理の種類	想定外部デバイスとの処理とオーディオシグナルフロー
割り込みによる PCM データ送信、 DMAC による PCM データ送信	送信バッファに格納した正弦波を CODEC に送信する。 
割り込みによる PCM データ受信、 DMAC による PCM データ受信	CODEC からの信号を受信し、受信バッファに格納する。 
割り込みによる PCM データ受信と送信、 DMAC による PCM データ受信と送信	CODEC からの信号を受信し受信バッファに格納、送信バッファにコピーし、送信する。（ループバック処理） 

1.1.2　応用プログラム

応用プログラムは、スマート・コンフィグレータを使用して rx64m_example_ssi_dma_rx_tx を RX72N Envision Kit 用に変換したプロジェクトです。変換方法は「4.スマート・コンフィグレータによるプロジェクトの移行」を参照してください。

PCM データ転送の処理の種類に対応するサンプルプログラムを表 1-3 に示します。サンプルプログラムの処理とオーディオシグナルフローを図 1-1 に示します。ソースコード上のマクロ定義により、図 1-1、表 1-4 に示すように SSIE1 の信号ソースとして、オンボードの MEMS マイクか正弦波かを選択できます。

表 1-3　転送処理とプロジェクトのファイルの対応（応用プログラム）

転送処理の種類	サンプルプログラム	対象デバイス
DMAC による PCM データ受信と送信	rx72n_example_ssi_dma_rx_tx	RX72N

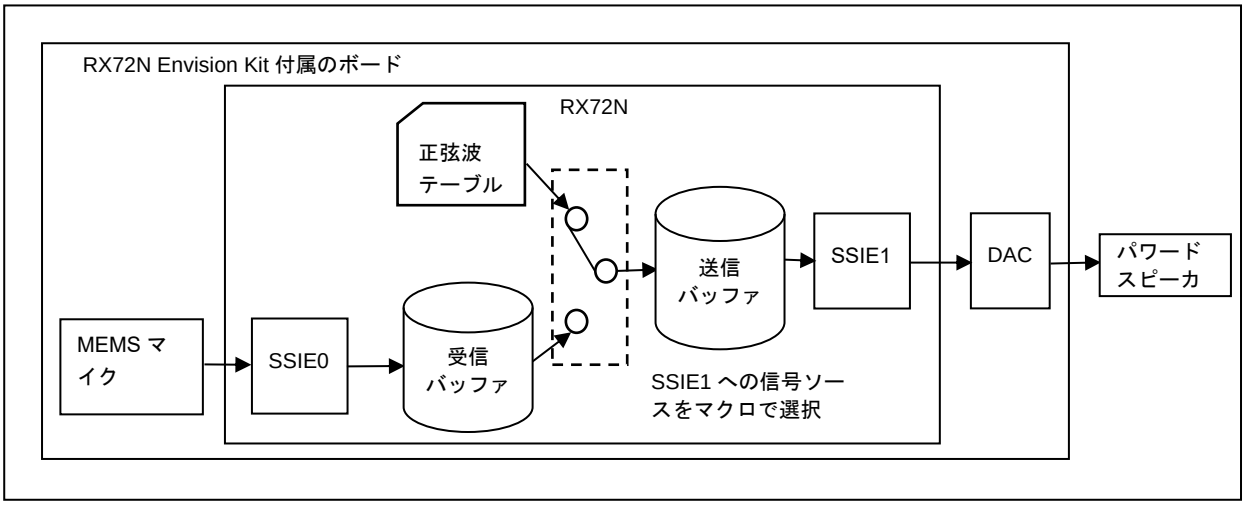


図 1-1　オーディオシグナルフロー（応用プログラム）

表 1-4　SSIE1 の信号ソース選択（応用プログラム）

マクロ定義	SSIE1 の信号ソース
#define OUTPUT_SIGNAL_FROM　SINE_WAVE	正弦波データ（デフォルト）
#define OUTPUT_SIGNAL_FROM　MEMS_MIC	オンボードの MEMS マイクからの入力信号

2. 動作確認条件

表 2-1、表 2-2 に動作確認条件を示します。図 2-1、図 2-2 にターゲットシステムの構成を示します。図 2-1 は Renesas Starter Kit+ for RX64M 付属の CPU ボードと外部の要素の接続を示しています。「RX64M グループ Renesas Starter Kit+ ユーザーズマニュアル」を参照し、ターゲットシステムを構成してください。

図 2-2 は RX72N Envision Kit 付属のボード上の RX72N と D/A コンバータなどオーディオ系部品との接続を示します。RX72N Envision Kit 付属のボードにはクロックジェネレータ IC など PCM データ転送のための部品が実装されています。このため部品の追加なしに RX72N から PCM データ転送を行うことができます。

表 2-1　動作確認条件（基本プログラム）

項目	内容
使用マイコン	R5F564MLCDFC（RX64M グループ)
動作周波数	・メインクロック ：24MHz ・PLL 回路:240MHz（メインクロックの 10 逓倍） ・システムクロック（ICLK）: 120MHz（PLL 回路の 2 分周） ・周辺モジュールクロック A（PCLKB）: 60MHz（PLL 回路の 4 分周） ・周辺モジュールクロック B（PCLKB）: 60MHz（PLL 回路の 4 分周） ・周辺モジュールクロック C（PCLKB）: 60MHz（PLL 回路の 4 分周） ・周辺モジュールクロック D（PCLKD）: 60MHz（PLL 回路の 4 分周） ・外部バスクロック（BCLK）: 60MHz（PLL 回路の 4 分周） ・FlashIF クロック（FCLK）: 60MHz（PLL 回路の 4 分周）
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e² studio Version 2020-04
C コンパイラ(注 1)	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V.3.02.00
	コンパイルオプション 統合開発環境のデフォルト設定
iodefine.h のバージョン	Version 1.0A
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
SSI	SSI0 を使用
サンプルコードのバージョン	Version 2.00
使用ボード	Renesas Starter Kit+ for RX64M（製品型名： R0K50564MSxxxBE）付属の CPU ボード
測定器等	Audio Precision SYS2722 と PSIA-2722 をクロックジェネレータとオーディオ CODEC として使用

注1.　元のプロジェクトで指定するツールチェーン(C コンパイラ）と同一のバージョンがインポートする先にない場合は、ツールチェーンが選択されない状態になり、エラーが発生します。プロジェクトの設定画面でツールチェーンの選択状態を確認してください。設定方法は、FAQ 3000404 を参照してください。

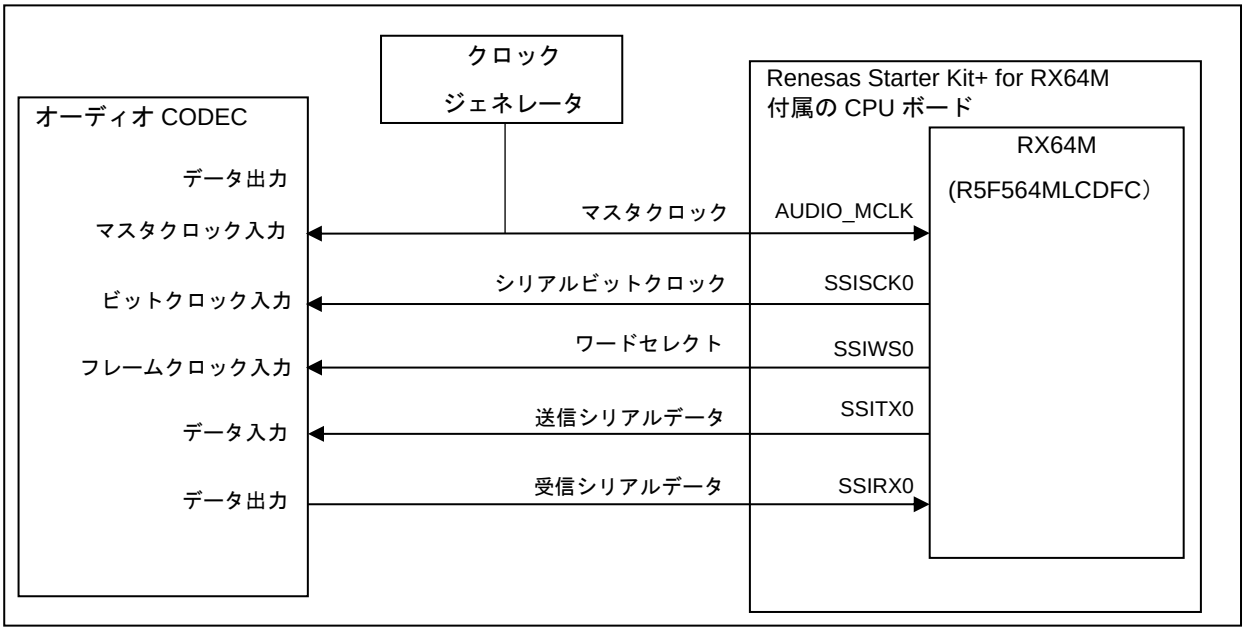


図 2-1　ターゲットシステムの構成（基本プログラム）

表 2-2　動作確認条件（応用プログラム）

項目	内容
使用マイコン	R5F572NDHDFB（RX72N グループ)
動作周波数	・メインクロック ：16MHz ・PLL 回路:240MHz（メインクロックの 15 逓倍） ・システムクロック（ICLK）: 240MHz（PLL 回路の 1 分周） ・周辺モジュールクロック A（PCLKB）: 60MHz（PLL 回路の 4 分周） ・周辺モジュールクロック B（PCLKB）: 60MHz（PLL 回路の 4 分周） ・周辺モジュールクロック C（PCLKB）: 60MHz（PLL 回路の 4 分周） ・周辺モジュールクロック D（PCLKD）: 60MHz（PLL 回路の 4 分周） ・外部バスクロック（BCLK）: 60MHz（PLL 回路の 4 分周） ・FlashIF クロック（FCLK）: 60MHz（PLL 回路の 4 分周）
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e^2 studio Version 2020-04
C コンパイラ(注 1)	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V.3.02.00
	コンパイルオプション 統合開発環境のデフォルト設定
iodefine.h のバージョン	Version 1.00C
エンディアン	リトルエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
SSI	SSIE0 と SSIE1 を使用
サンプルコードのバージョン	Version 2.00
使用ボード	RX72N Envision Kit（製品型名： RTK5RX72N0C00000BJ）付属のボード
測定器等	クロックジェネレータ IC(5X35023) D/A コンバータ(D2-41051-QR) MEMS マイク(ICS-43434)

注1. 元のプロジェクトで指定するツールチェーン(C コンパイラ）と同一のバージョンがインポートする先にない場合は、ツールチェーンが選択されない状態になり、エラーが発生します。プロジェクトの設定画面でツールチェーンの選択状態を確認してください。設定方法は、FAQ 3000404 を参照してください。

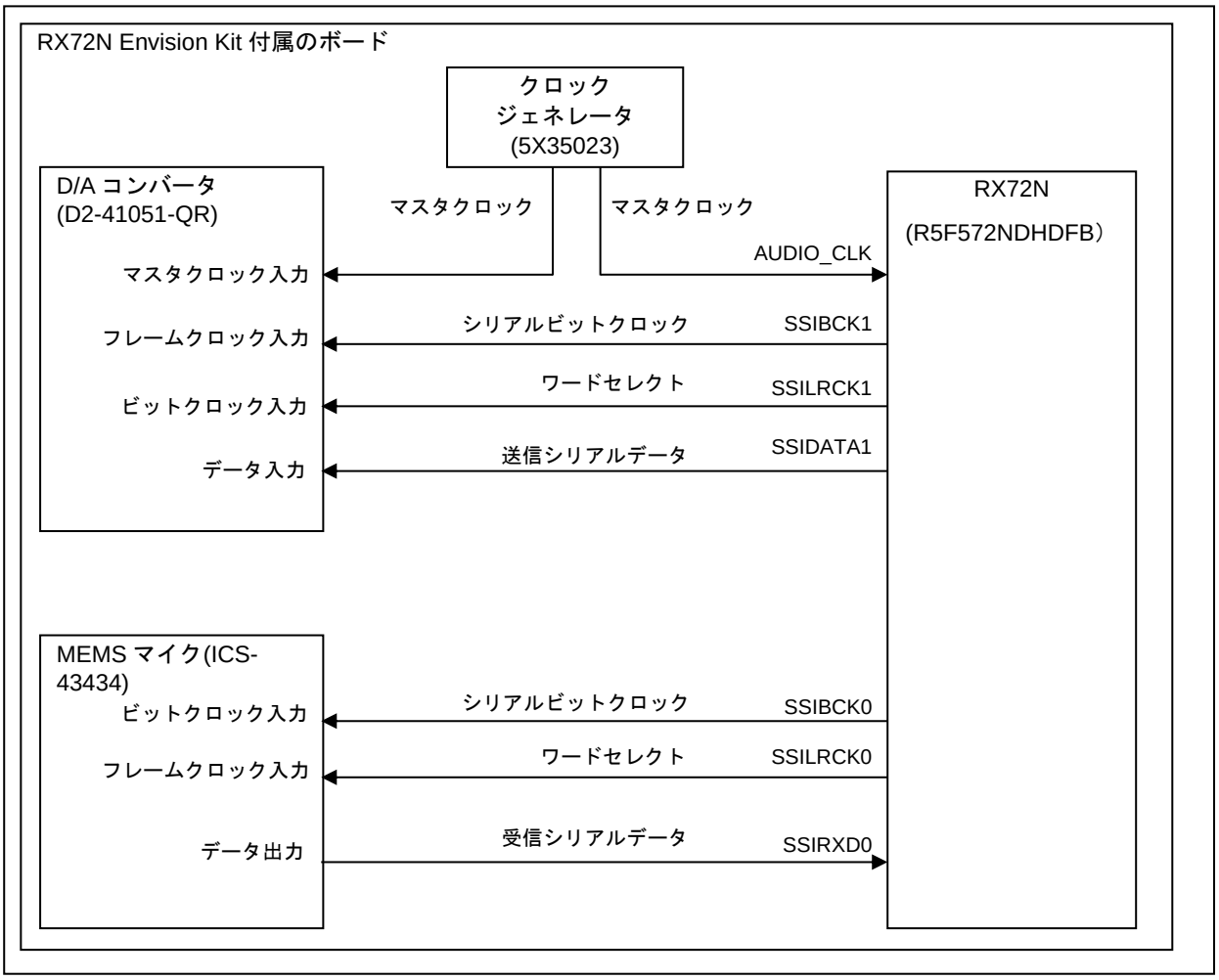


図 2-2　ターゲットシステムの構成（応用プログラム）

3. ソフトウェア説明

3.1 ソフトウェアの構成要素

サンプルプログラムのソフトウェアの構成要素を

表 3-1 と表 3-2 に示します。システム設定と SSI モジュールを用いる PCM データ転送処理は main()に記述されています。従って、SSI モジュールによる PCM データ送信を理解するには main()を含むサンプルプログラムのファイルをご覧ください。各 FIT モジュールの詳細はそれぞれのアプリケーションノートをご覧ください。（「6. 参考資料」参照）

なお本章では基本プログラムでソフトウェアの動作を説明します。

表 3-1　ソフトウェアの構成要素（基本プログラム）

転送処理の種類	main()を含むサンプルプログラム	FIT モジュール
割り込みによる PCM データ送信	r_main_ssi_tx.c (Rev.2.00)	BSP モジュール (Rev.5.52) SSI モジュール (Rev.2.02) MPC モジュール (Rev.3.40) GPIO モジュール (Rev.3.40)
割り込みによる PCM データ受信	r_main_ssi_rx.c (Rev.2.00)	
割り込みによる PCM データ受信と送信	r_main_ssi_rx_tx.c (Rev.2.00)	
DMAC による PCM データ送信	r_main_ssi_dma_tx.c (Rev.2.00)	BSP モジュール (Rev.5.52) SSI モジュール (Rev.2.02) DMAC モジュール (Rev.2.30) MPC モジュール (Rev.3.40) GPIO モジュール (Rev.3.40)
DMAC による PCM データ受信	r_main_ssi_dma_rx.c (Rev.2.00)	
DMAC による PCM データ受信と送信	r_main_ssi_dma_rx_tx.c (Rev.2.00)	

「PCM データ送信」と「DMAC による PCM データ送信」は、PCM データテーブルファイル「r_test_wave.h」を含みます。

表 3-2　ソフトウェアの構成要素（応用プログラム）

転送処理の種類	main()を含むサンプルプログラム	FIT モジュール
DMAC による PCM データ送信と送信	r_main_ssi_dma_rx_tx.c(Rev.2.00)	BSP モジュール (Rev.5.52) SSI モジュール (Rev.2.02) DMAC モジュール (Rev.2.30) MPC モジュール (Rev.3.40) GPIO モジュール (Rev.3.40)

PCM データテーブルファイル「r_test_wave.h」を含みます。

3.2 SSI0 の割り込みを使用する PCM データ転送

図 3-1 は、SSI0 の割り込みを使用する PCM データ転送処理の主な構成要素を示します。使用する要素は処理内容ごとに異なります。PCM データ送受信処理の場合、すべての要素を使用します。一方、PCM データ送信処理の場合は送信用の要素のみ使用します。PCM データ受信処理の場合も同様です。

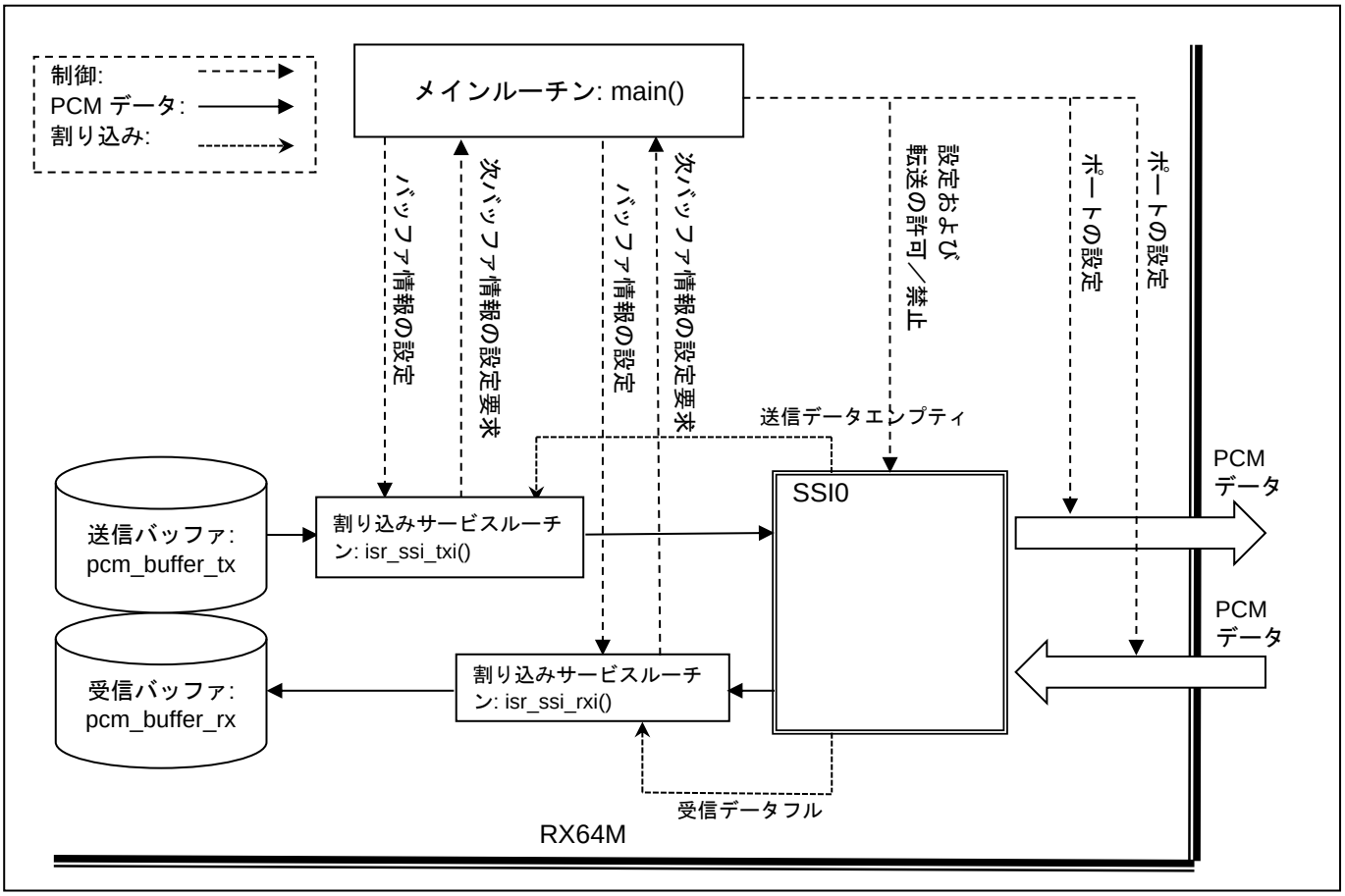


図 3-1　SSI0 の割り込みを使用する PCM データ転送

図 3-1 に照らし、SSI0 の割り込みを使用する PCM データ転送処理の手順を以下に説明します。

1) 周辺機能の設定

main() は、周辺機能 GPIO、MPC および SSI を各 FIT モジュールの API を使用し設定します。

2) 送信および受信バッファの情報の設定

main()は、送信および受信バッファの情報を設定します。この情報はサンプル数と開始アドレスからなり、転送の単位を表します。後続の説明のようにこの情報は PCM データの書き込みおよび読み出し処理に使用されます。

3) 転送の開始

main()は、SSI0 に PCM データ転送の開始を許可します。

4) PCM データ転送中の処理

転送を開始すると、SSI0 から送信データエンプティと受信データフル割り込みが発生します。送信エンプティ割り込みが発生すると、対応する割り込みサービスルーチン isr_ssi_txi()が R_SSI_Write()を実行し送信バッファに格納されている PCM データが SSI0 の送信 FIFO に書き込まれます。これは割り込みによる周期的な処理です。手順 2)で設定されたすべての PCM データが転送されたら、isr_ssi_txi()は main() に次のバッファ情報の設定を要求します。このように isr_ssi_txi()の実行とバッファ情報の更新を繰り返すことで PCM データは途切れることなく持続的に SSI0 から出力されます。
受信データフル割り込みが発生すると、対応する割り込みサービスルーチン isr_ssi_rxi()が R_SSI_Read()を実行し SSI0 の受信 FIFO の PCM データを読み出し受信バッファに格納します。これは割り込みによる周期的な処理です。手順 2)で設定されたすべての PCM データが転送されたら、isr_ssi_rxi()は main() に次のバッファ情報の設定を要求します。このように isr_ssi_rxi()の実行とバッファ情報の更新を繰り返すことで PCM データは途切れることなく持続的に SSI0 から入力されます。

表 3-3　割り込みを使用する場合の MCU リソース

MCU リソース	転送処理の種類（割り込みを使用する場合）		
	送信	受信	送受信
SSI	SSI0	SSI0	SSI0
SSI0 端子	SSISCK0, SSIWS0, SSITXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSIRXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSITXD0, SSIRXD0, AUDIO_MCLK
割り込み	送信データエンプティ	受信データフル	送信データエンプティ、受信データフル

3.3　DMAC を使用する PCM データ転送

図 3-2 は、DMAC を使用する PCM データ転送の主な構成要素を示します。使用する要素は処理内容ごとに異なります。PCM データ送受信処理の場合、すべての要素を使用します。一方、PCM データ送信処理の場合は送信用の要素のみ使用します。PCM データ受信処理の場合も同様です。

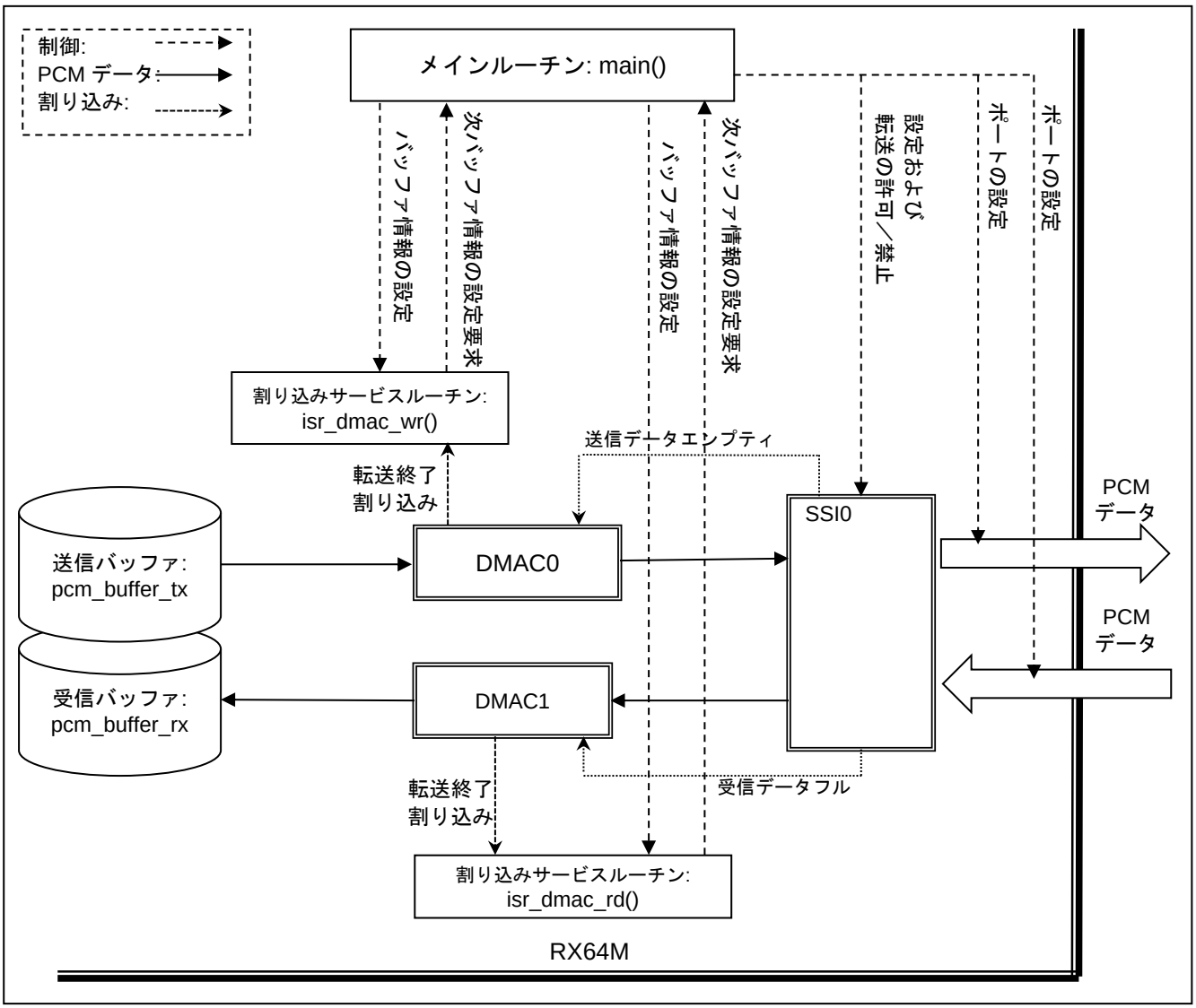


図 3-2　DMAC を使用する PCM データ転送処理

図 3-2 に照らし、DMAC を使用する PCM データ転送処理の手順を以下に説明します。

1)　周辺機能の設定

main() は、周辺機能 GPIO、MPC、DMAC および SSI を各 FIT モジュールの API を使用し設定します。

2) 送信および受信バッファの情報の設定

main()は、送信および受信バッファの情報を設定します。情報はサンプル数と開始アドレスからなり、転送の単位を表します。後続の説明のように、この情報は DMAC の処理に使用されます。

3) 転送の開始

main()は、DMA 転送を許可したあと SSI0 に PCM データ転送の開始を許可します。

4) PCM データ転送中の処理

DMAC0 は送信バッファから SSI0 の送信 FIFO に PCM データを転送し、DMAC1 は SSI0 の受信 FIFO から受信バッファに PCM データを転送します。
DMAC0 転送終了割り込みが発生すると、割り込みサービスルーチン isr_dmac_wr()が実行されます。これは手順 2)で設定したサンプル数の転送が終了したことを示し、isr_dmac_wr()は main()に次のバッファ情報の設定を要求します。このように isr_dmac_wr()の実行とバッファ情報が更新を繰り返すことで PCM データは途切れることなく持続的に SSI0 から出力されます。
DMAC1 転送終了割り込みが発生すると、割り込みサービスルーチン isr_dmac_rd()が実行されます。これは手順 2)で設定したサンプル数の転送が終了したことを示し、isr_dmac_rd()は main()に次のバッファ情報の設定を要求します。このように isr_dmac_rd()の実行とバッファ情報が更新を繰り返すことで PCM データは途切れることなく持続的に SSI0 から入力されます。

表 3-4　DMAC を使用する場合の MCU リソース

MCU リソース	転送処理の種類（DMAC を使用する場合）		
	送信	受信	送受信
SSI	SSI0	SSI0	SSI0
SSI0 端子	SSISCK0, SSIWS0, SSITXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSIRXD0, AUDIO_MCLK	SSISCK0, SSIWS0, SSITXD0, SSIRXD0, AUDIO_MCLK
割り込み	DMAC0 転送終了	DMAC1 転送終了	DMAC0 転送終了、DMAC1 転送終了
DMAC	チャネル 0	チャネル 1	チャネル 0、チャネル 1

3.4　PCM データバッファ

図 3-1 と図 3-2 に示す送信バッファと受信バッファは、送信または受信する PCM データを格納します。バッファは int16_t 型の 2 次元配列です。行はブロック数を示し、列は 1 ブロックあたりのサンプル数を示します。

- ブロック数：表 3-5 参照
- 1 ブロックのサンプル数：1024 x 2
- PCM データの格納順序：左チャネル、右チャネルの順

表 3-5　PCM データバッファのブロック数

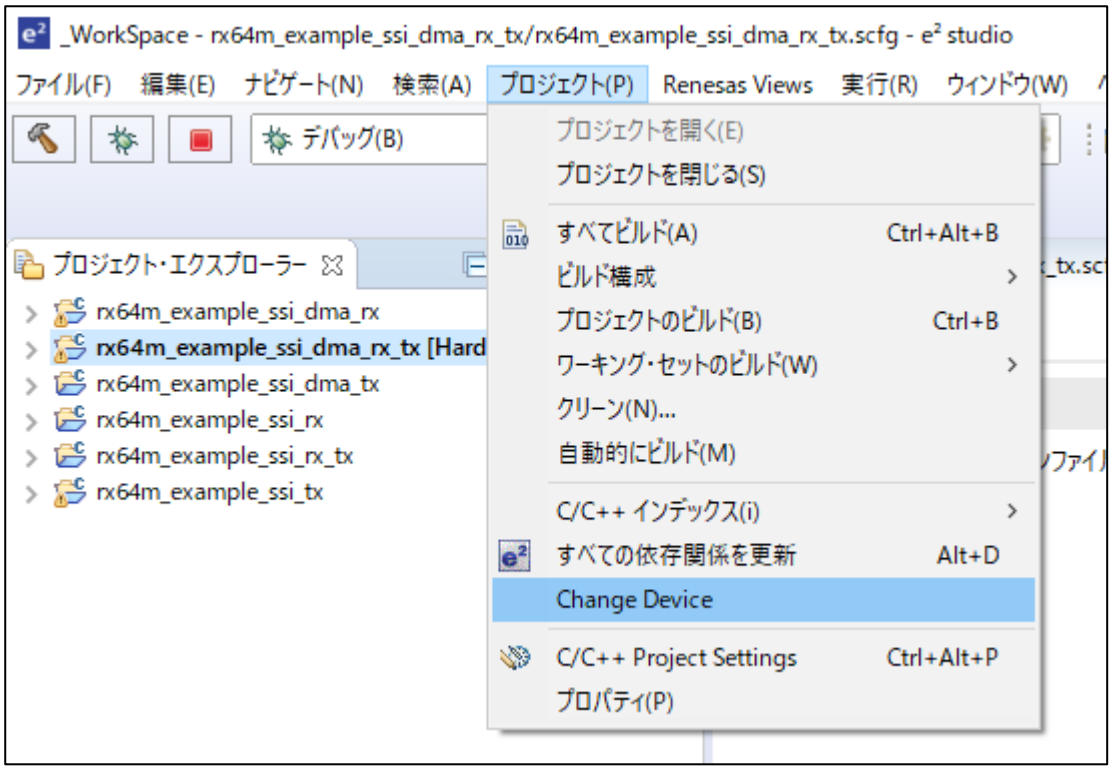
	転送処理の種類		
	送信	受信	送受信
ブロック数	2	2	3

4. スマート・コンフィグレータによるプロジェクトの移行

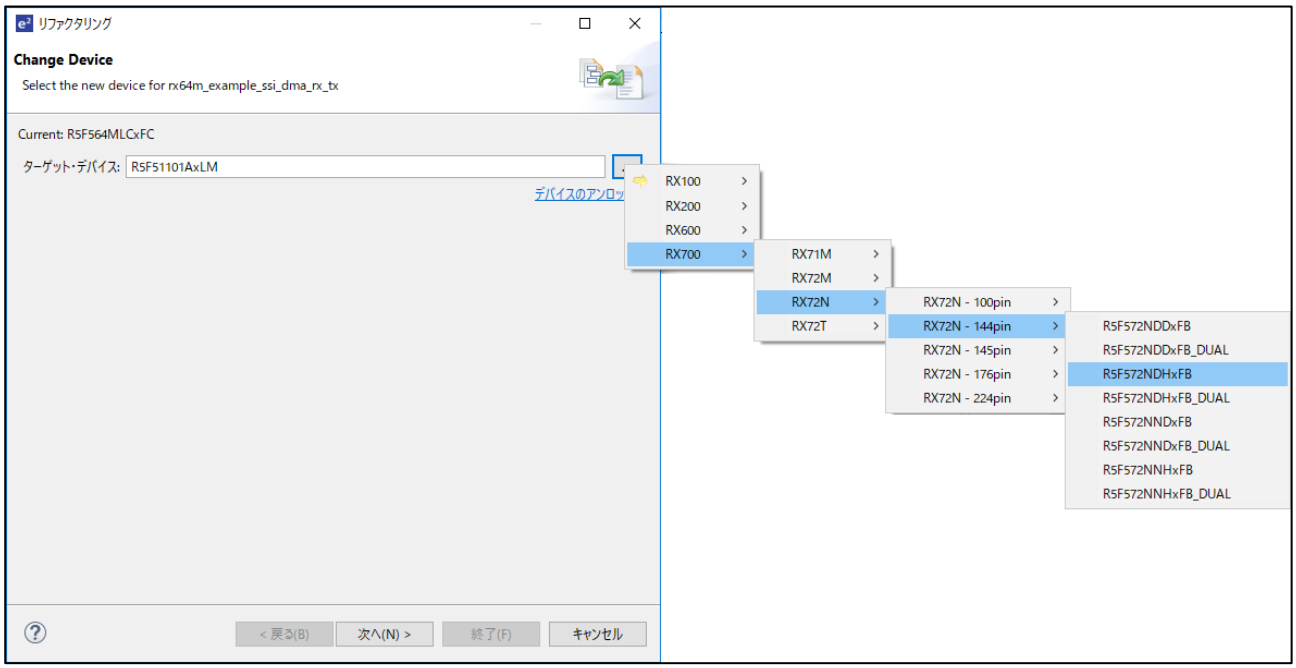
スマート・コンフィグレータを使用すると、異なるデバイス間でプロジェクト設定の移行をすることができます。本章では rx64m_example_ssi_dma_rx_tx のプロジェクトを RX72N Envision Kit 用のプロジェクトに移行するための手順を説明します。また、移行した後のプロジェクトとして rx72n_example_ssi_dma_rx_tx を添付していますので参考にしてください。

4.1　デバイスの変更

プロジェクトを選択し、［プロジェクト］メニューから［Change Device］を選択します。



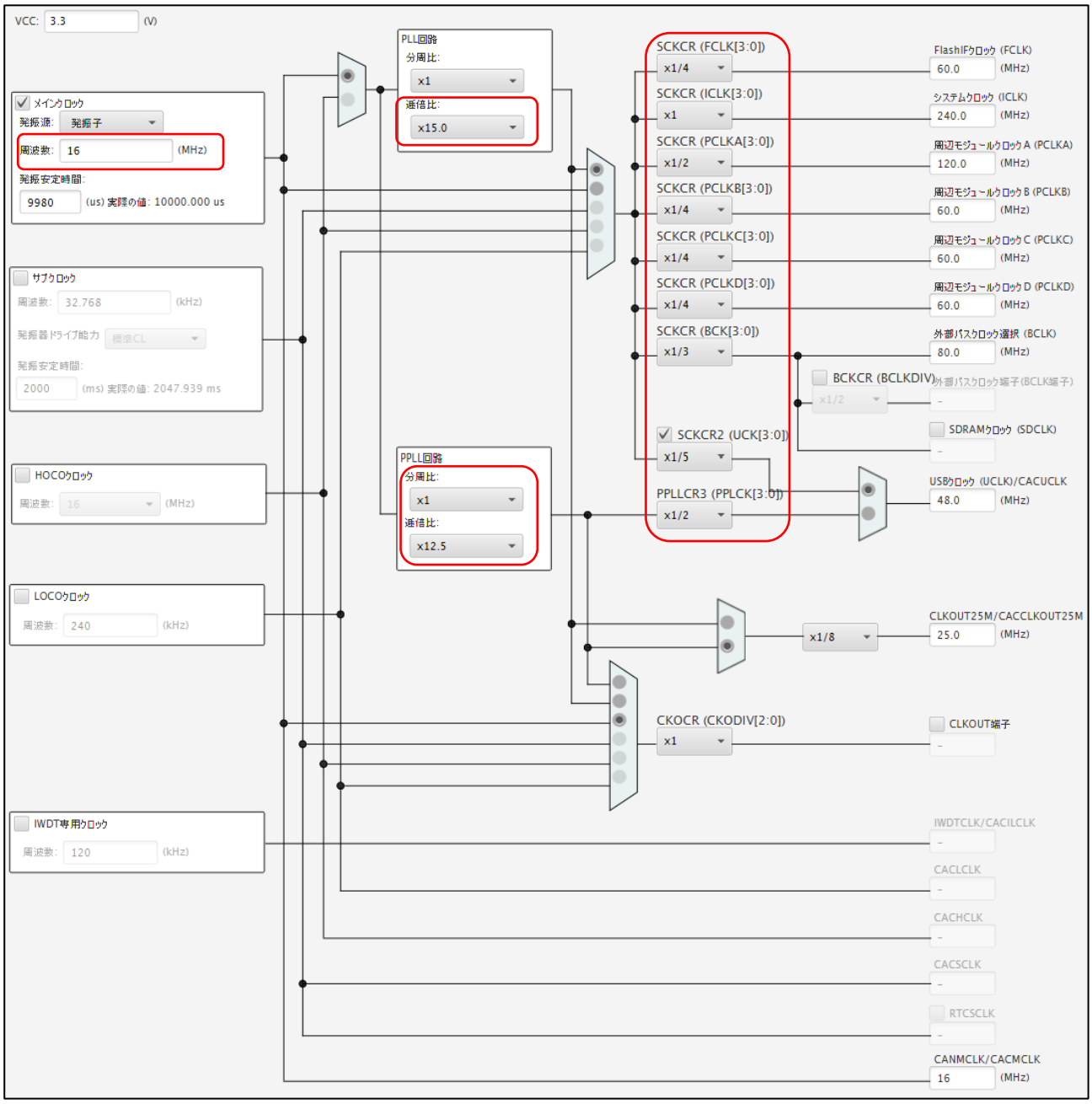
［デバイス選択］ ボタンをクリックして、”R5F572NDHxFB”を選択して［次へ］をクリックします。



4.2　クロック設定

RX72N Envision Kit に合わせてクロック設定をします。

プロジェクト・エクスプローラーから rx64m_example_ssi_dma_rx_tx.scfg を開き、スマート・コンフィグレータを表示させ、クロックタブを選択した後、以下の設定をしてください。 (変更箇所を赤枠で示します。)



4.3　SSI の設定と SSI で使用する端子設定

SSI の設定と、SSI で使用する端子の設定をします。

プロジェクト・エクスプローラーから rx64m_example_ssi_dma_rx_tx.scfg を開き、スマート・コンフィグレータを表示させ、コンポーネントタブを選択し、r_ssi_api_rx を選択した後、以下の手順で SSI の設定と SSI で使用する端子設定をしてください。項目、設定内容は設定画面の表記で記載しています。

表 4-1　SSI の設定と SSI で使用する端子設定

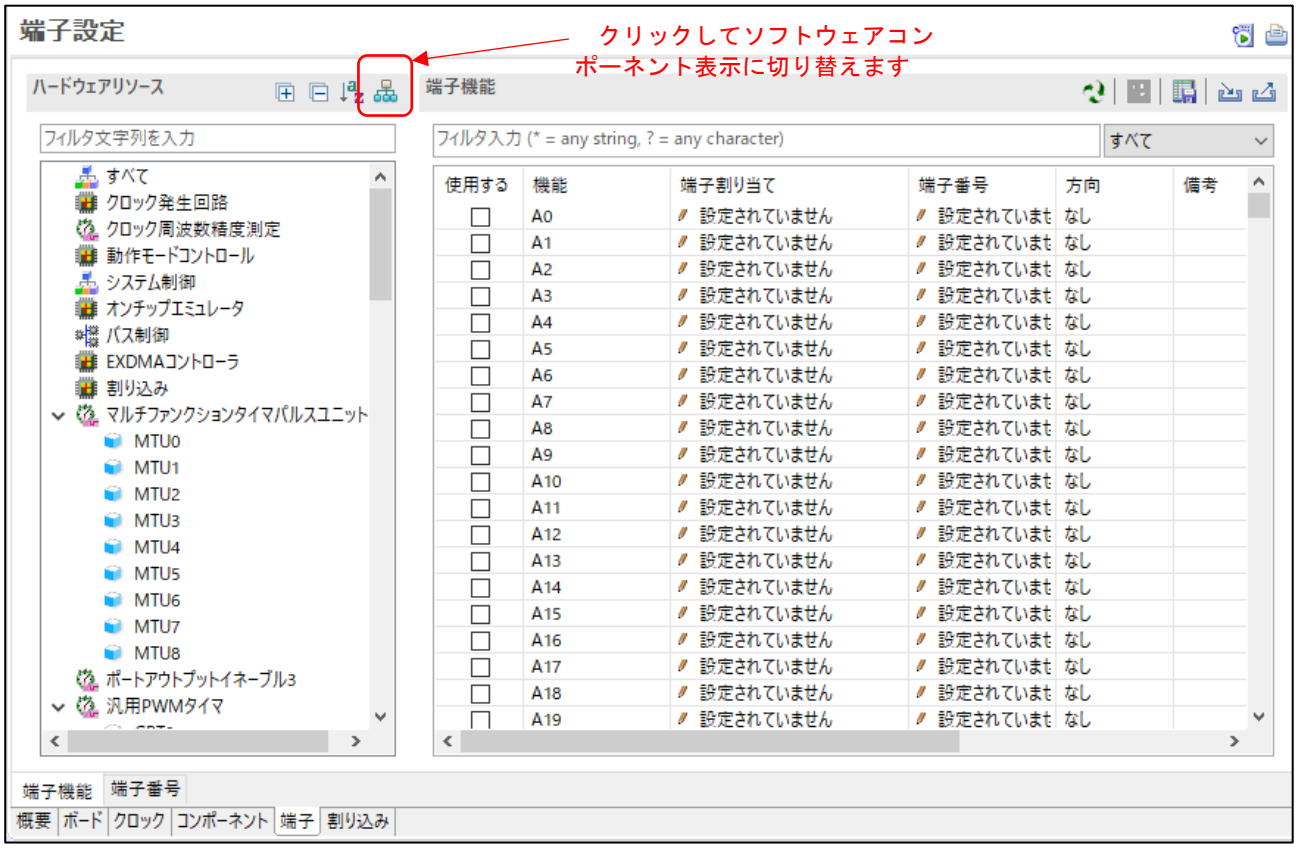
プロパティ	設定内容
Use Standard configuration	Standard
SSI or SSIE	SSIE
Ch0 Transmit / Receive mode	SSIE to Receive data
Ch1 Transmit / Receive mode	SSIE to Transmit data
Ch0 Audio serial interface format	I2S
Ch1 Audio serial interface format	I2S
Ch0 PCM data width	24 bits
Ch1 PCM data width	24 bits
Ch0 Bit Clock	64 fs
Ch1 Bit Clock	64 fs
Master Clock	512 u
Ch0 Clock Supply Mode	Master mode
Ch1 Clock Supply Mode	Master mode
Ch0 Tx Empty trigger	8
Ch1 Tx Empty trigger	8
Ch0 Rx Full trigger	8
Ch1 Rx Full trigger	8
Ch0 Byte Swap	Byte swapping is disabled
Ch1 Byte Swap	Byte swapping is disabled
SSIE0	チェックする
SSIBCK0 端子	使用する
SSILRCK0 端子	使用する
SSITXD0 端子	使用しない
SSIRXD0 端子	使用する
AUDIO_CLK 端子	使用する
SSIE1	チェックする
SSIBCK1 端子	使用する
SSILRCK1 端子	使用する
SSIDATA1 端子	使用する
AUDIO_CLK 端子	使用する

4.4　端子割り当ての変更

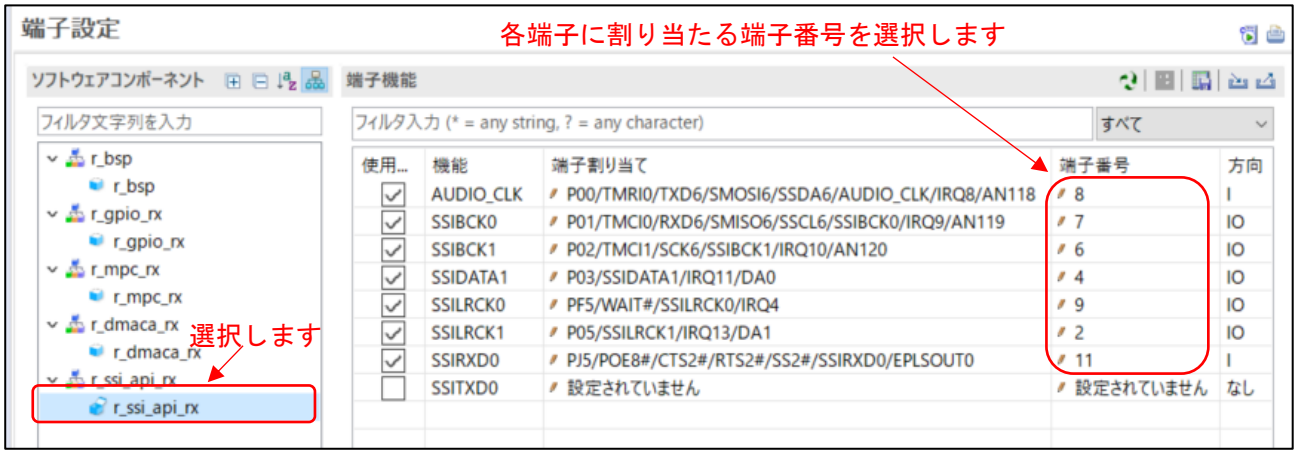
SSI で使用する端子の割り当てを変更します。

プロジェクト・エクスプローラーから rx64m_example_ssi_dma_rx_tx.scfg を開き、スマート・コンフィグレータを表示させ、端子タブを選択した後、以下の設定をしてください。

端子タブを選択し、ソフトウェアコンポーネント表示に切り替えます。



ソフトウェアコンポーネントの中から r_ssi_api_rx を選択し、各端子に割り当たる端子番号を選択します。



4.5　その他の変更

rx64m_example_ssi_dma_rx_tx では SSI0 で送信と受信をしていますが、RX72N Envision Kit 用のプロジェクトに移行するため、「r_main_ssi_dma_rx_tx.c」を以下のように変更し、SSIE0 で受信、SSIE1 で送信となるようにしています。

表 4-2　r_main_ssi_dma_rx_tx.c の変更内容

変更箇所	変更内容
PCM データバッファ	バッファの型を int32_t に変更
main()	SSIE0 と SSIE1 を使用するように変更
	SSIE1 の信号ソースとして正弦波データを使用するか、MEMS マイクからの入力信号を使用するかを選択する処理を追加
isr_ssi_status_error()	SSIE1 のエラー判定処理を追加
init_port_ssi0()	RX72N Envision Kit にあわせた SSIE0 と SSIE1 の端子設定に変更
init_dmac_wr() isr_dmac_wr() init_dmac_rd() isr_dmac_rd()	DMA の転送回数を変更

5. サンプルプログラムの実行

5.1 ハードウェアのセットアップ

5.1.1 基本プログラムを使用する場合

図 5-1 に基本プログラムを使用する場合の動作環境の例を示します。図 2-1 と図 5-1 を参考に動作環境を構成してください。基本プログラムでは、Renesas Starter Kit+ for RX64M を使用します。

Renesas Starter Kit+ for RX64M 付属の AC アダプタを使い下記手順に従ってセットアップしてください。

1. パワードスピーカをオーディオ CODEC に接続します。パワードスピーカにボリュームが付いている場合は、ボリュームを最小にした状態で接続し、プログラム実行後に、ボリュームを適切な値に設定してください。
2. ターゲットシステムの Renesas Starter Kit+ for RX64M 同梱の CPU ボードに E2 Lite エミュレータをユーザインタフェースケーブルで接続します。
3. Renesas Starter Kit+ for RX64M 同梱の CPU ボードに AC アダプタを接続し、電源を供給します。
4. E2 Lite エミュレータをホスト PC に USB ケーブルで接続します。

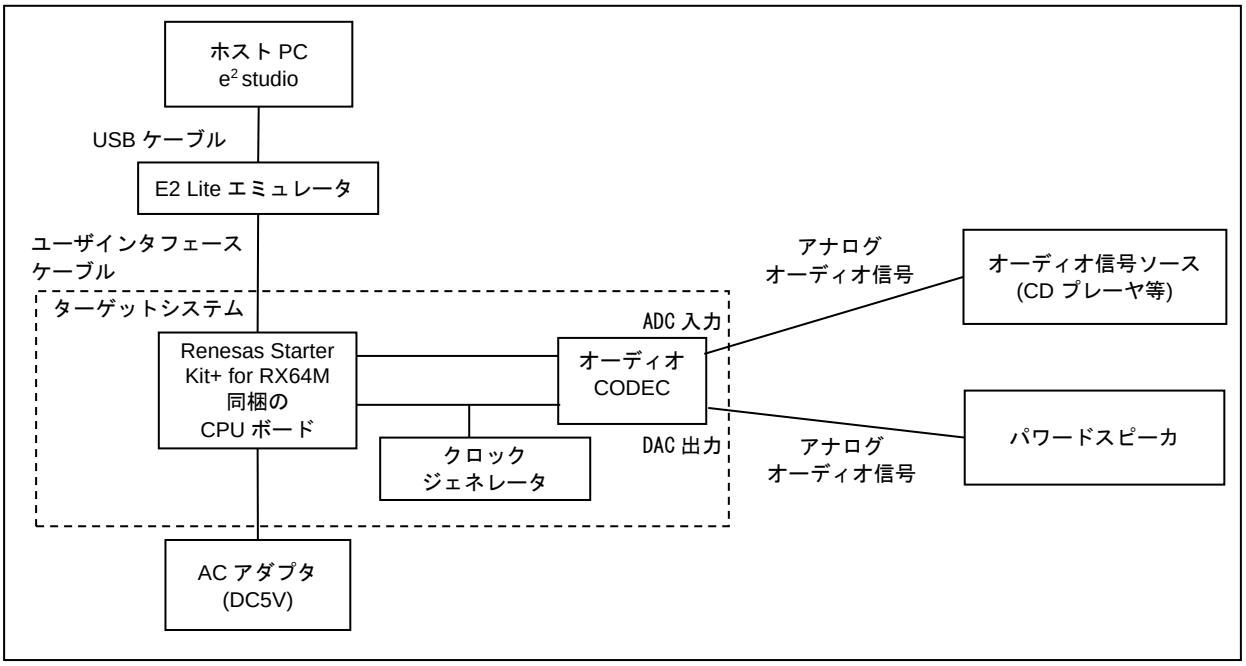


図 5-1　動作環境の例（基本プログラム）

5.1.2　応用プログラムを使用する場合

図 5-2 に応用プログラムでの動作環境の例を示します。応用プログラムでは RX72N Envision Kit を使用します。図 2-2 と図 5-2 を参考に動作環境を構成してください。

下記手順に従ってセットアップしてください。

1. パワードスピーカのボリュームを最小にして、RX72N Envision Kit 同梱のボードの音声出力ジャックに接続します。パワードスピーカにボリュームが付いている場合は、ボリュームを最小にした状態で接続し、プログラム実行後に、ボリュームを適切な値に設定してください。
2. ターゲットシステムの RX72N Envision Kit 同梱のボードに USB ケーブルで接続します。ボードの電源は USB コネクタから供給されます。
3. D/A コンバータへの入力は、サンプルプログラムのマクロで正弦波かオンボードの MEMS マイクかを選択可能です。デフォルトは正弦波です。

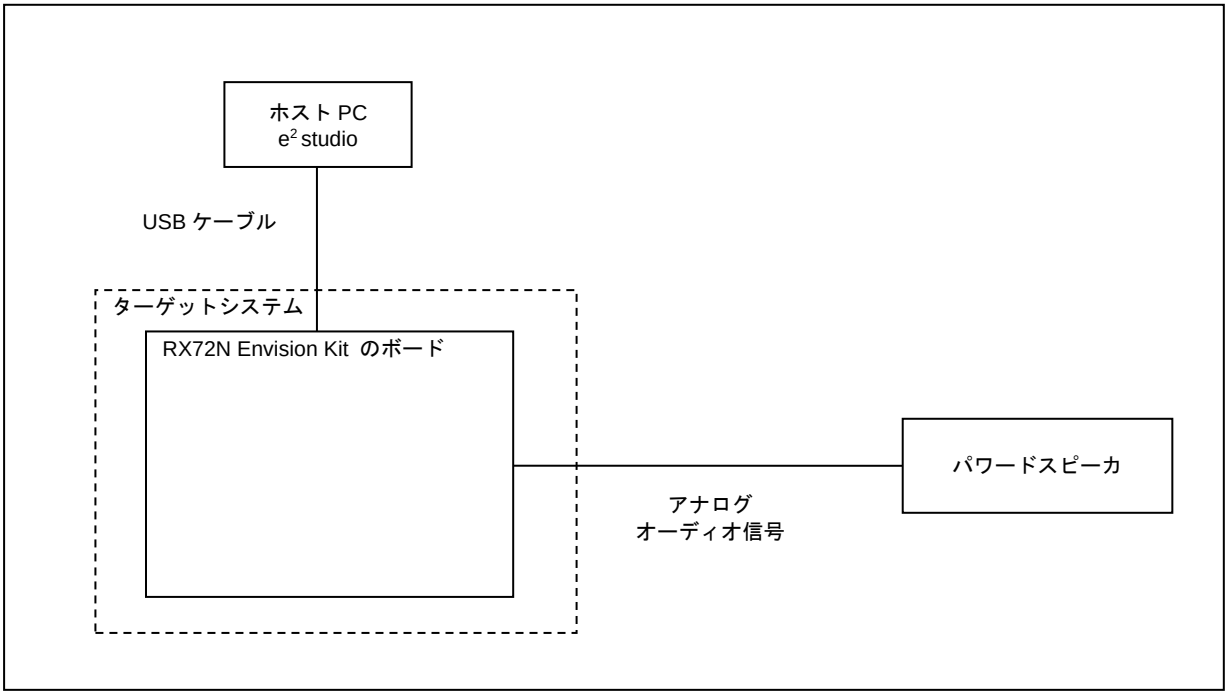


図 5-2　動作環境の例（応用プログラム）

5.2　サンプルプログラムのインポート

サンプルプログラムは e² studio のプロジェクト形式で提供しています。本章では、 e² studio および CS+ へプロジェクトをインポートする方法を示します。インポート完了後、ビルドおよびデバッグの設定を確認してください。

5.2.1　e² studio での手順

e² studio でご使用になる際は、下記の手順で e² studio にインポートしてください。

(使用する e² studio のバージョンによっては画面が異なる場合があります。)

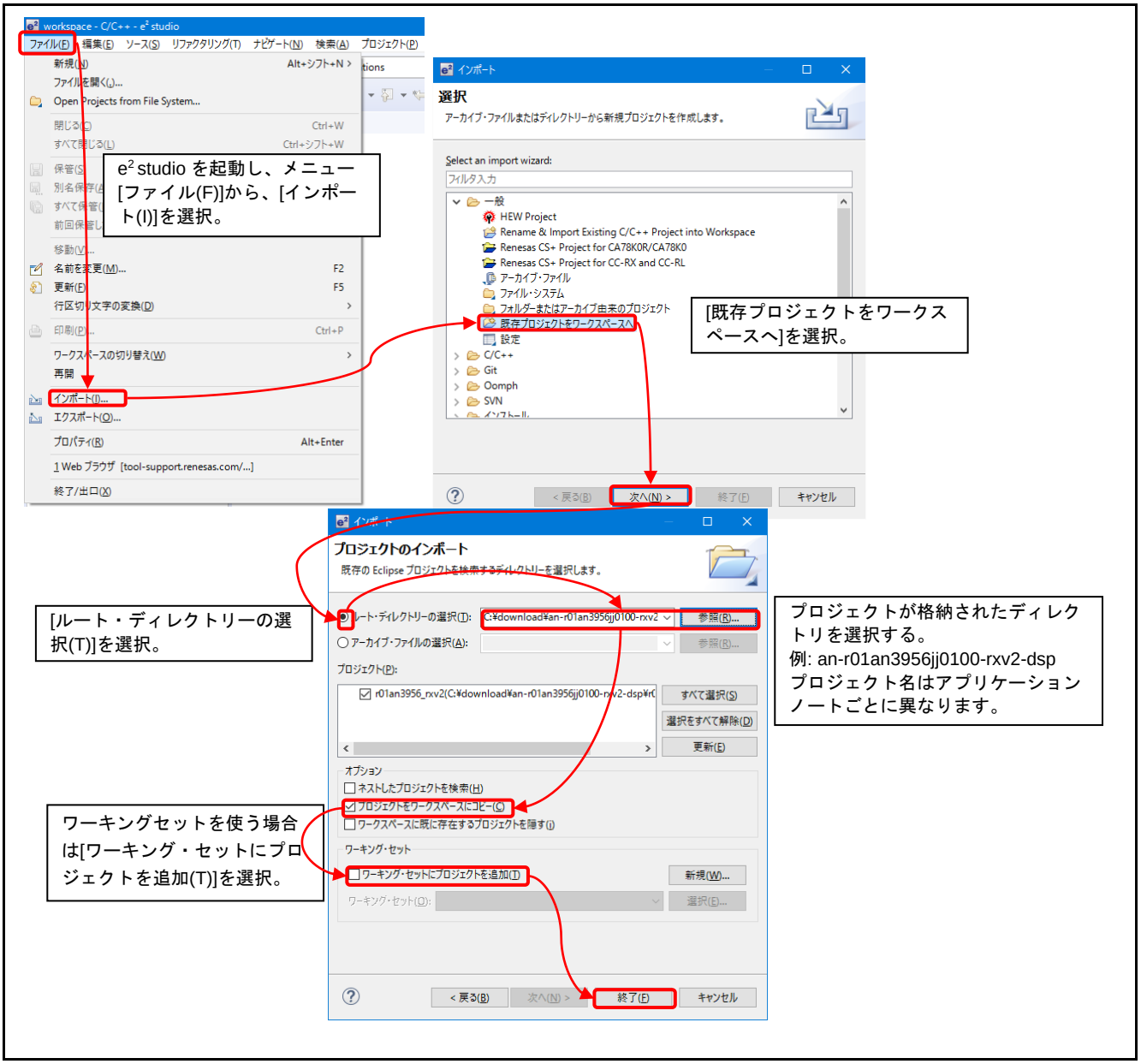


図 5-3　プロジェクトを e2 studio にインポートする方法

5.2.2　CS+での手順

CS+でご使用になる際は、下記の手順で CS+にインポートしてください。

(使用する CS+のバージョンによっては画面が異なる場合があります。)

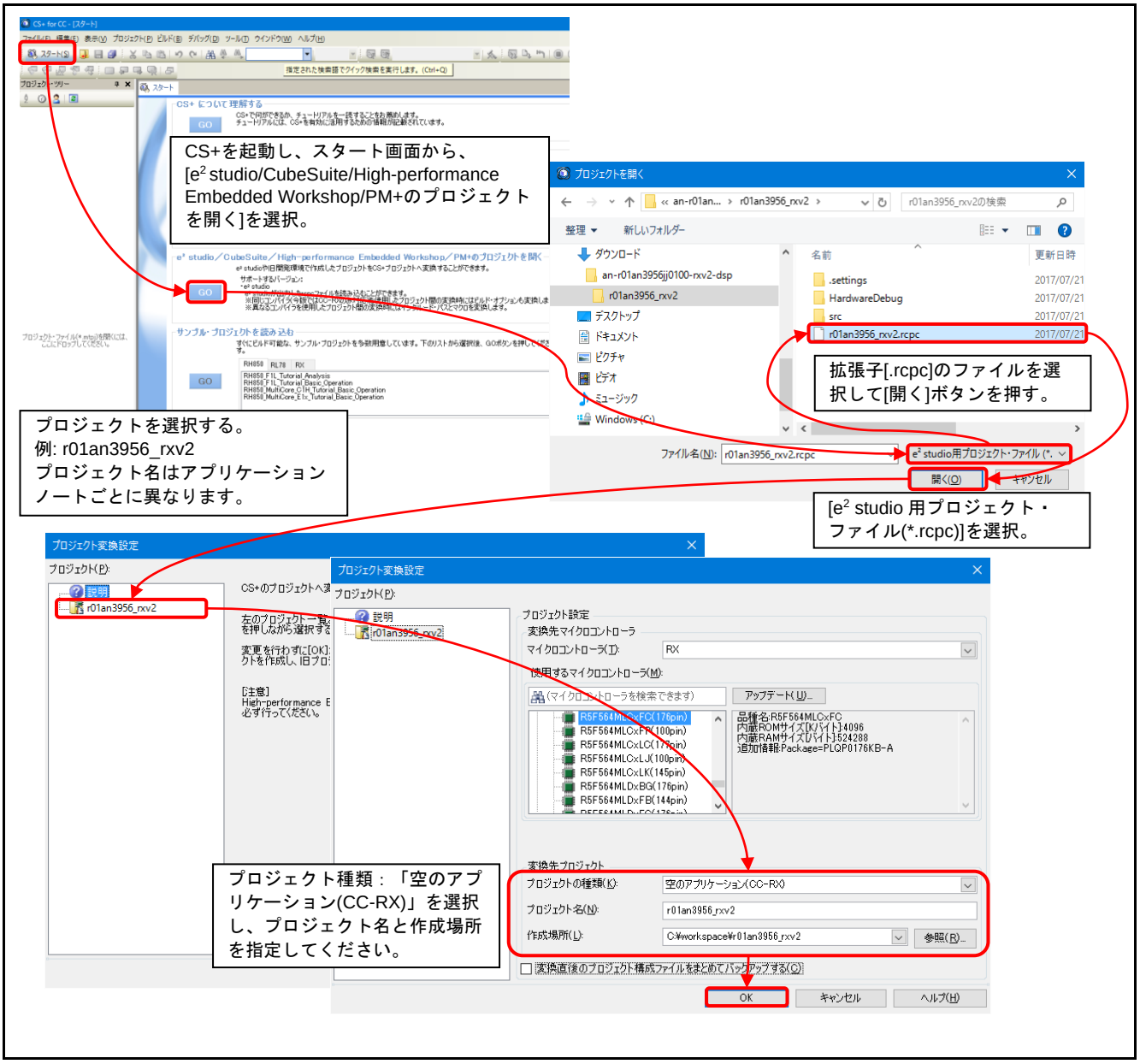


図 5-4　プロジェクトを CS+にインポートする方法

6. 参考資料

Firmware Integration Technology アプリケーションノート

RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)

RX ファミリ SSI モジュール Firmware Integration Technology (R01AN2150)

RX ファミリ DMA コントローラ DMACA 制御モジュール Firmware Integration Technology (R01AN2063)

RX ファミリ GPIO モジュール Firmware Integration Technology (R01AN1721)

RX ファミリ MPC モジュール Firmware Integration Technology (R01AN1724)

（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）

Renesas Starter Kit+ for RX64M ユーザーズマニュアル

RX64M グループ Renesas Starter Kit+ ユーザーズマニュアル (CubeSuite+) (20UT2590)

RX64M Group Renesas Starter Kit+ User's Manual For e^2 studio (R20UT2593)

（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）

RX72N Envision Kit ユーザーズマニュアル

RX72N グループ Envision Kit ユーザーズマニュアル (R20UT4788)

（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）

ユーザーズ マニュアル：ハードウェア

RX64M グループ ユーザーズマニュアル ハードウェア編 (R01UH0377)

RX72N グループ ユーザーズマニュアル ハードウェア編 (R01UH0824)

（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）

テクニカルアップデート、ツールニュース

（最新の情報をルネサス エレクトロニクスホームページから入手してください）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2015/6/5	-	初版発行
	2017/6/1	-	日本語版発行
2.00	Jun.30.20	-	以下の対象デバイスを追加 RX72N

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、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、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。
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。
6. 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。
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、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。
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。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. 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。
10. 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。
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。
12. 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.4.0-1　2017.11)

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。

© 2020 Renesas Electronics Corporation. All rights reserved.