

RX ファミリ

R01AN2839JJ0100

Rev.1.00

SRC モジュールを使用するサンプリングレート変換サンプルプログラム

2017.6.1

Firmware Integration Technology

要旨

本アプリケーションノートは、サンプリングレートコンバータ (SRC) と Firmware Integration Technology (FIT) に準拠した SRC モジュールを使用するサンプリングレート変換を行う 2 つのサンプルプログラムを提供します。これらはサンプリングレート変換に必要なすべての要素を包含した e2studio のプロジェクトの形式です。各サンプルプログラムには、ルネサスの評価ボード上で実際にサンプリングレート変換を行う処理を記述しています。

本書では、最初にサンプルプログラムの概要、次に e2studio でのサンプルプログラムの使い方を説明します。

対象デバイス

- ・ RX64M グループ

目次

1. 概要.....	2
1.1 添付プロジェクトについて	2
1.2 ハードウェアの条件	3
1.3 開発ツールの条件.....	3
2. サンプルプログラム.....	3
2.1 ソフトウェアの構成要素	3
2.2 PIO でのサンプリングレート変換.....	4
2.3 DMAC でのサンプリングレート変換	5
2.4 MCU リソース	6
2.5 PCM データバッファ	7
3. サンプルプログラムの実行.....	7
3.1 ハードウェアのセットアップ	7
3.2 e2studio を使用したサンプルプログラムの実行.....	7
4. 参考資料.....	14

1. 概要

本書に添付した 2 つのサンプルプログラムは、典型的なサンプリングレート変換処理における SRC 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。これら 2 つのプログラムは共に 8KHz から 44.1KHz へのサンプリングレート変換を行います。違いは SRC と PCM バッファの間のデータの転送方式です。一方はプログラム I/O (PIO) で、もう一方は DMAC で転送を行います。

PCM データフォーマット

入力および出力 PCM データのフォーマットを以下に示します。

- 入力 PCM データ：16bit ステレオ、サンプリング周波数：8KHz
- 出力 PCM データ：16bit ステレオ、サンプリング周波数：44.1KHz

テスト信号

2 つのサンプルプログラムは以下のテスト信号を含みます

- Fs: 8kHz、Lch: フルスケールの 500Hz 正弦波、Rch: フルスケールの 250 Hz の正弦波

1.1 添付プロジェクトについて

本書と 2 つのプロジェクトはファイル r01an2839jj****_src_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。

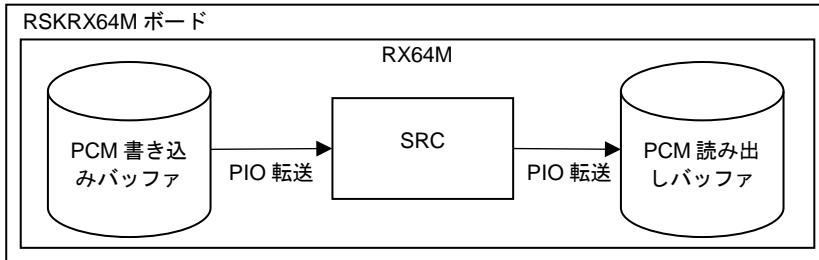
```
r01an2839jj****_src_rx + r01an2839jj****_rx.pdf
|
+ ---workspace
  rx64m_example_src.zip
  rx64m_example_src_dma.zip
```

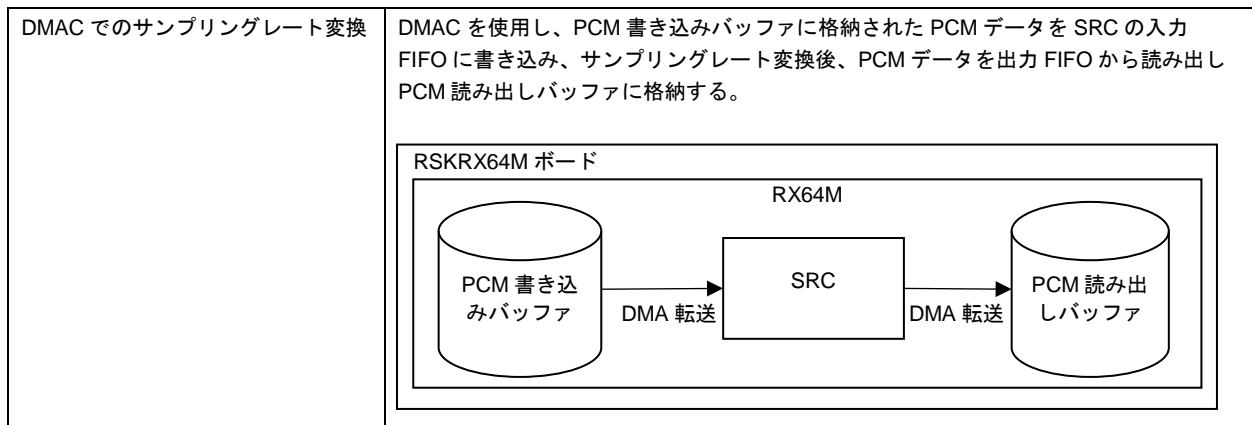
上記の 2 つの .zip ファイルは e2studio のプロジェクトのアーカイブファイルです。サンプリングレート変換の処理の種類に対応する .zip ファイルを表 1-1 に示します。これらのうち一つを選択し、e2studio でインポート、実行してください。各プロジェクトの処理とオーディオシグナルフローを表 1-2 に処理の種類ごとに示します。

表 1-1 サンプリングレート変換処理とプロジェクトのファイルの対応

処理の種類	プロジェクトのファイル
PIO でのサンプリングレート変換	rx64m_example_src.zip
DMAC でのサンプリングレート変換	rx64m_example_src_dma.zip

表 1-2 処理の種類ごとのオーディオシグナルフロー

処理の種類	RSKR64M ボードでのオーディオシグナルフロー
PIO でのサンプリングレート変換	<p>PCM 書き込みバッファに格納された PCM データを SRC の入力 FIFO に書き込み、サンプリングレート変換後、PCM データを出力 FIFO から読み出し PCM 読み出しバッファに格納する。</p>  <pre> graph LR subgraph RSKRX64M_Board [RSKR64M ボード] subgraph RX64M WriteBuf[(PCM 書き込みバッファ)] -- PIO 転送 --> SRC[Src] SRC -- PIO 転送 --> ReadBuf[(PCM 読み出しバッファ)] end end </pre>



1.2 ハードウェアの条件

「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。

ターゲットシステムの構成要素

- ターゲットボード： RSKRX64M CPU ボード
Renesas Starter Kit+ for RX64M (RSK+RX64M) 付属の評価ボードです。
ボード型名： R0K50564MC000BE
- ターゲット MCU： RX64M (R5F564MLCDFC)

1.3 開発ツールの条件

- 統合開発環境 e2studio、バージョン 3.1.2.10、ルネサスエレクトロニクス製
- RX ファミリ コンパイラパッケージ、バージョン 2.0.2.00、ルネサスエレクトロニクス製
- E1 エミュレータ、ルネサスエレクトロニクス製

2. サンプルプログラム

2.1 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 2-1 に示します。main()は、システム設定とサンプリングレート変換処理を実行します。main()は MCU の周辺機能を FIT モジュールを使って間接的に制御します。従って FIT モジュールの API を使用するサンプリングレート変換処理を理解するには、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。（「4. 参考資料」参照）

表 2-1 処理の種類ごとのソフトウェアの構成要素

処理の種類	main()を含むサンプルプログラム	FIT モジュール
PIO でのサンプリングレート変換	r_main_src.c (Rev.1.00)	BSP モジュール (Rev.2.80) SRC モジュール (Rev.1.11)
DMAC でのサンプリングレート変換	r_main_src_dma.c (Rev.1.00)	BSP モジュール (Rev.2.80) SRC モジュール (Rev.1.11) DMAC モジュール (Rev.1.02)

上記 2 つのサンプルプログラムは PCM データテーブルファイル「r_test_wave.h」を含みます。

2.2 PIO でのサンプリングレート変換

図 2-1 に PIO で PCM データの転送を行うサンプリングレート変換処理の主な構成要素を示します。この処理はメインルーチンのみからなり、割り込みサービスなどの他のルーチンはありません。main() はシステムの設定とサンプリングレート変換の処理を実行します。ポーリングにより SRC を制御するシステムです。

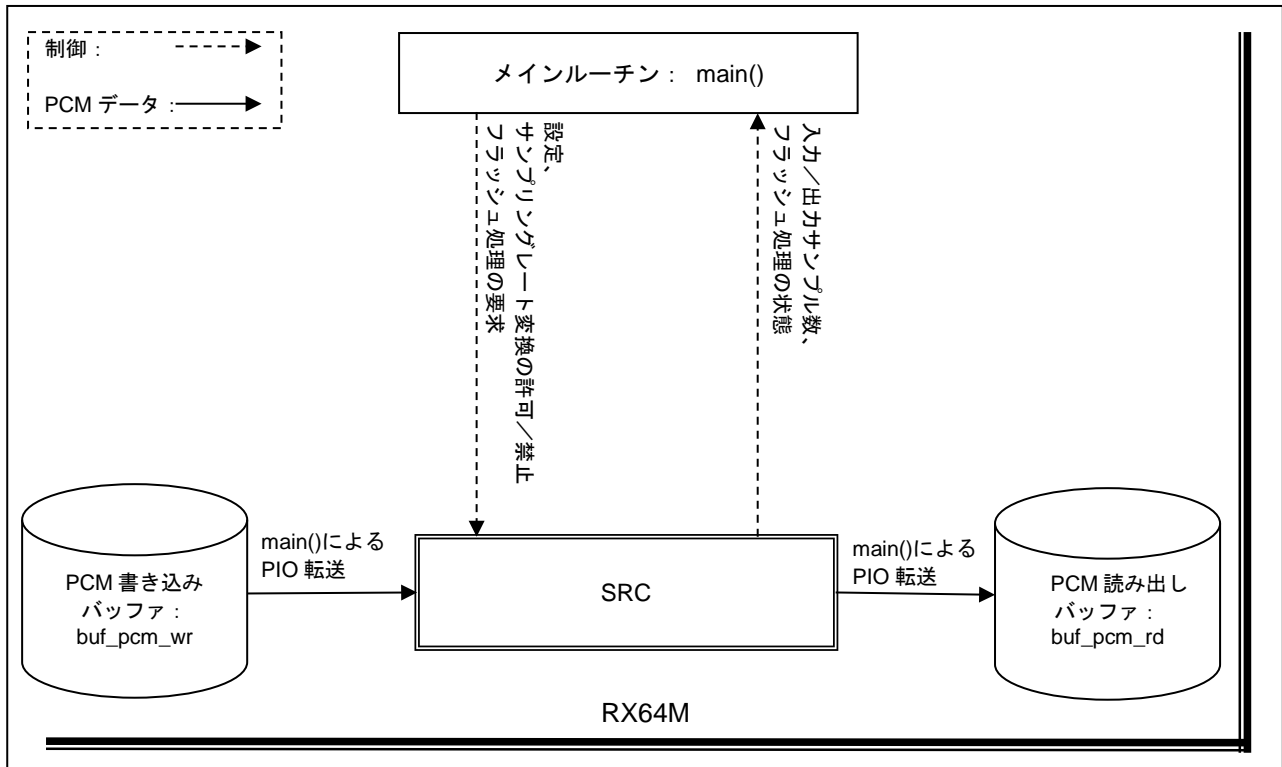


図 2-1 PIO でのサンプリングレート変換処理

図 2-1 に照らし、PIO で PCM データの転送を行うサンプリングレート変換の手順を以下に説明します。

1) 周辺機能の設定

main() は、SRC を SRC モジュールの API を使用し設定します。

2) PCM 書き込みバッファ、PCM 読み出しバッファの情報の設定

main() は、PCM 書き込みバッファと PCM 読み出しバッファの情報を設定します。この情報はバッファの開始アドレスとサンプル数からなります。この情報は転送の単位を表し、後続の説明のように PCM データの書き込みおよび読み出しの処理に使用されます。

3) サンプリングレート変換の開始

main() は、入力および出力のサンプリングレートを指定し、SRC にサンプリングレート変換動作を許可します。

4) サンプリングレート変換

main() はサンプリング変換動作中、以下の a) と c) を繰り返し非同期に実行します。

- main() は SRC の入力 FIFO に変換前の PCM データを書き込みます。
- SRC は変換動作を開始、変換後の PCM データを出力 FIFO に出力します。
- main() は SRC の出力 FIFO から変換後の PCM データを読み出します。

PCM データ書き込み転送

上記の処理 a) は、R_SRC_Write() を使用する PCM 書き込みバッファから SRC の入力 FIFO へのデータ転送です。バッファのアドレスとサンプル数は手順 2) で設定した値となります。main() は

R_SRC_Write()の実行の都度、その戻り値を使ってこれらの値を更新します。戻り値は実際に転送することができたサンプル数を示します。

PCM データ読み出し転送

上記の処理 c)は、R_SRC_Read()を使用する SRC の出力 FIFO から PCM 読み出しバッファへのデータ転送です。バッファのアドレスとサンプル数は手順 2)で設定した値となります。main()は R_SRC_Read()の実行の都度、その戻り値によってこれらの値を更新します。戻り値は実際に転送することができたサンプル数を示します。

5) 変換終了

指定サンプル数 (PCM 書き込みバッファ 1 ブロック分) の書き込み後、main()は処理 a)を停止、変換を終了するため R_SRC_Stop()を使用しフラッシュ処理を SRC に要求します。main()は変換後のすべての PCM データを読み出し、R_SRC_Read()の戻り値がフラッシュ処理の完了を示すまで処理 c)を続けます。サンプリングレート変換後のすべての PCM データは PCM 読み出しバッファに格納されています。

2.3 DMAC でのサンプリングレート変換

図 2-2 に DMAC で PCM データの転送を行うサンプリングレート変換処理の主な構成要素を示します。この処理はメインルーチンのみからなり、割り込みサービスなどの他のルーチンはありません。main()はシステムの設定とサンプリングレート変換の実行をします。ポーリングで SRC を制御するシステムです。

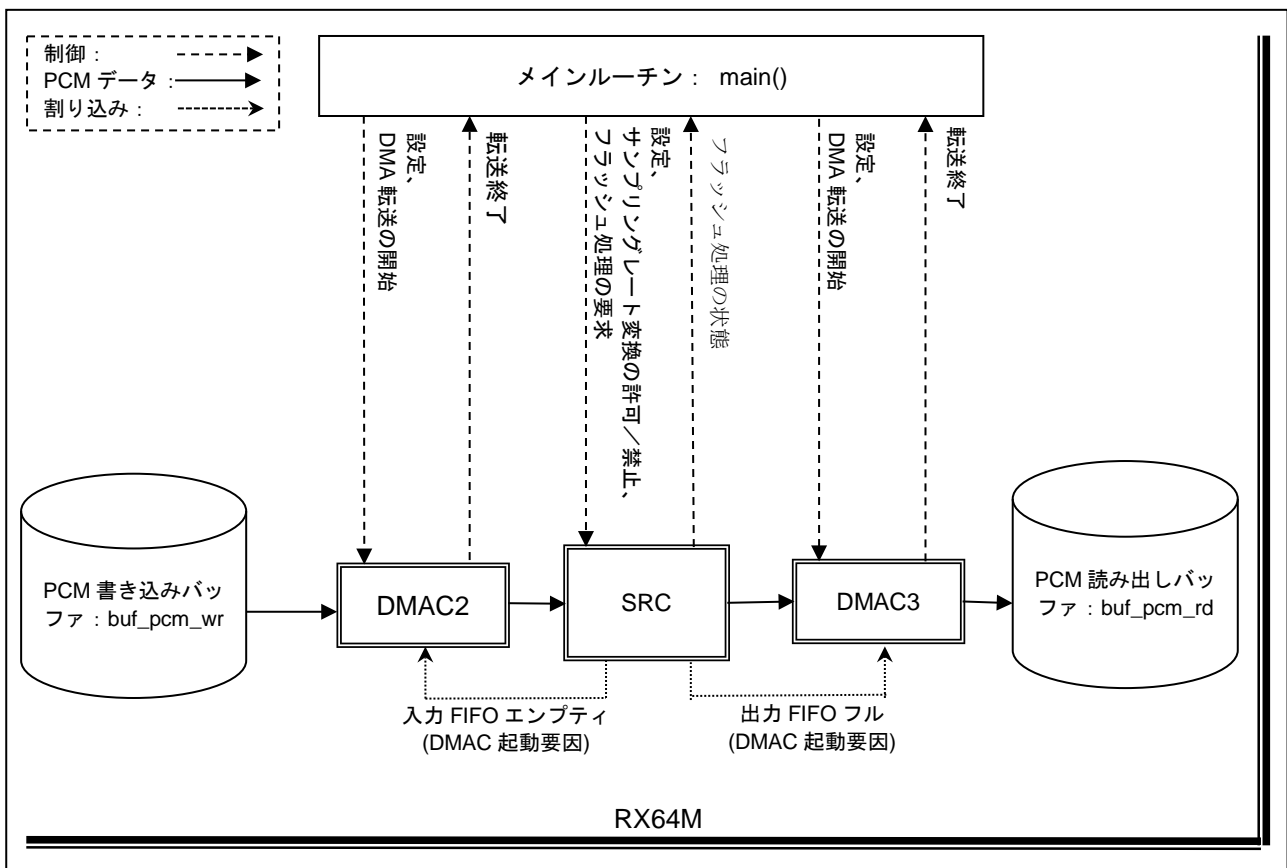


図 2-2 DMAC を使用するサンプリングレート変換処理

図 2-2 に照らし、DMAC で PCM データの転送を行うサンプリングレート変換の手順を以下に説明します。

1) 周辺機能の設定

main() は、DMAC と SRC を各 FIT モジュールの API を使用し設定します。

2) DMAC 転送の情報の設定

main()は、PCM データ転送のため DMAC モジュールの API を使って DMAC2 と DMAC3 を設定します。転送の単位は PCM バッファ 1 ブロックとします。main()は PCM 書き込みバッファと PCM 読み出しバッファそれぞれの開始アドレスとサンプル数を設定し、DMAC2 と DMAC3 の転送を許可します。

3) サンプリングレート変換の開始

main()は入力と出力のサンプリングレートを指定し、SRC にサンプリングレート変換動作を許可します。

4) サンプリングレート変換

DMAC2 と DMAC3 は、SRC の入力 FIFO エンプティと出力 FIFO フルを起動要因として転送を行います。サンプリングレート変換中、起動要因の発生により DMAC2 と DMAC3 による転送は繰り返し非同期に発生します。

- a) DMAC2 は変換前の PCM データを SRC の入力 FIFO に転送します。
- b) SRC は変換動作を開始、変換後の PCM データを出力 FIFO に出力します。
- c) DMAC3 は変換後の PCM データを SRC の出力 FIFO から転送します。

DMAC2 による PCM データ書き込み転送

上記の処理 a)は、DMAC2 を使用する PCM 書き込みバッファから SRC の入力 FIFO へのデータ転送です。バッファのアドレスとサンプル数は手順 2)で設定した値となります。すべての PCM データが転送されると、main()は次の転送のため DMAC2 を設定し転送を許可します。

DMAC3 による PCM データ読み出し転送

上記の処理 c)は、DMAC3 を使用する SRC の出力 FIFO から PCM 読み出しバッファへのデータ転送です。バッファのアドレスとサンプル数は手順 2)で設定した値となります。main()はすべての PCM データが転送されると、main()は次の転送のため DMAC3 を設定し転送を許可します。

5) 変換終了

指定回数 (PCM 書き込みバッファ 2 ブロック分) の DMAC2 による PCM データ書き込み転送後、main()は DMAC2 による処理 a)を停止させ、変換を終了するために R_SRC_Stop()を使用しフラッシュ処理を SRC に要求します。main()は、変換後のすべての PCM データを読み出すまで DMAC3 による c)の処理を続けます。サンプリングレート変換の完了は R_SRC_CheckFlush()を使用して確認します。R_SRC_CheckFlush()の戻り値がフラッシュ処理の完了を示したあと、main()は DMAC3 の転送を強制的に停止させます。サンプリングレート変換後のすべての PCM データは PCM 読み出しバッファに格納されています。

2.4 MCU リソース

表 2-2 にサンプルプログラムで使用する MCU リソースを示します。

表 2-2 MCU リソース

	サンプルプログラムごとの MCU リソース	
	PIO でのサンプリングレート変換	DMAC でのサンプリングレート変換
SRC	SRC	SRC
DMAC 起動要因	-	SRC の入力 FIFO エンプティ (DMAC2) 、 SRC の出力 FIFO フル (DMAC3)
DMAC	-	DMAC2、DMAC3

2.5 PCM データバッファ

図 2-1 と図 2-2 に示す PCM 書き込みバッファと PCM 読み出しバッファは、サンプリングレート変換前または変換後の PCM データを格納します。バッファは `uint16_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。表 2-3 にサンプリングレート変換処理別の PCM バッファサイズをまとめます。

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

表 2-3 PCM データバッファサイズ

		PIO でのサンプリングレート変換	DMAC でのサンプリングレート変換
PCM 書き込み バッファ	ブロック数	1	2
	1 ブロックのサンプル数	1024 x 2	1024 x 2
PCM 読み出し バッファ	ブロック数	6	12
	1 ブロックのサンプル数	1024 x 2	1024 x 2

3. サンプルプログラムの実行

3.1 ハードウェアのセットアップ

図 3-1 に示す動作環境を参照し、以下の手順に則り動作環境をセットアップしてください。

1. ターゲットシステムの RSKRX64M CPU ボードに E1 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。
2. E1 エミュレータをホスト PC に USB ケーブルで接続します。

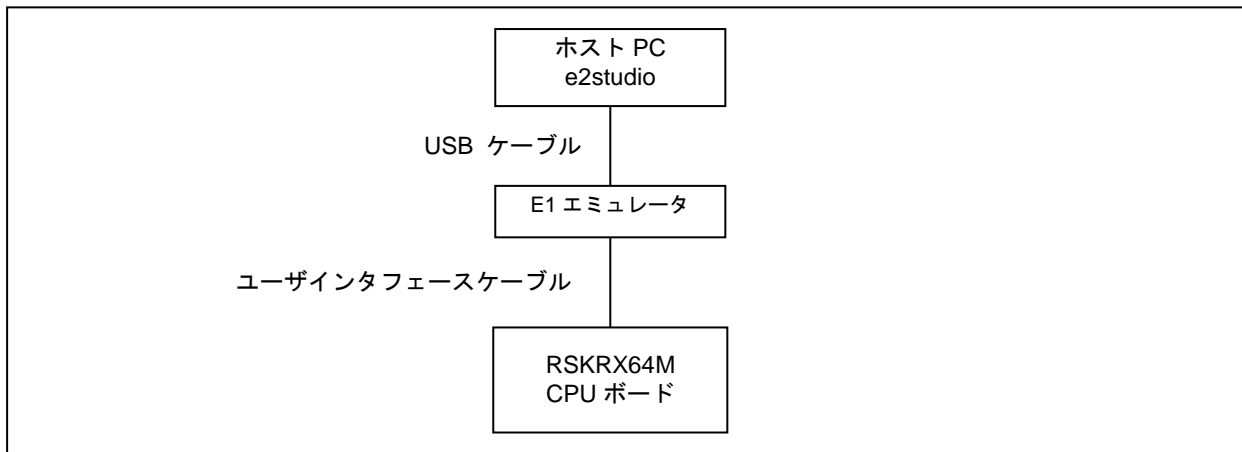


図 3-1 動作環境

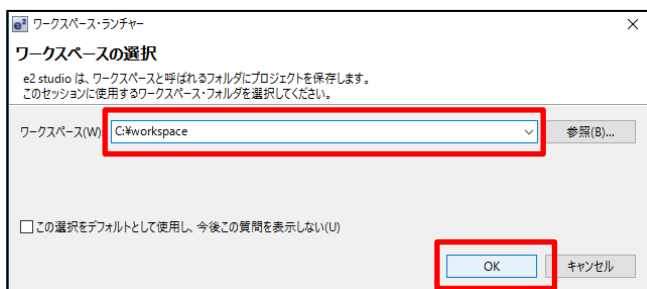
3.2 e2studio を使用したサンプルプログラムの実行

3.2.1 サンプルプロジェクトの e2studio へのインポート

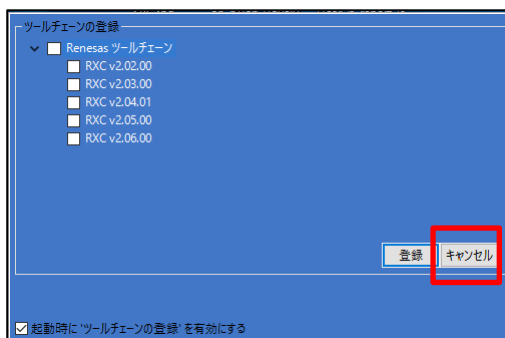
ここでは、例としてサンプルプロジェクト「rx64m_example_src」をインポートする方法を示します。

- 1) e2studio を起動します。

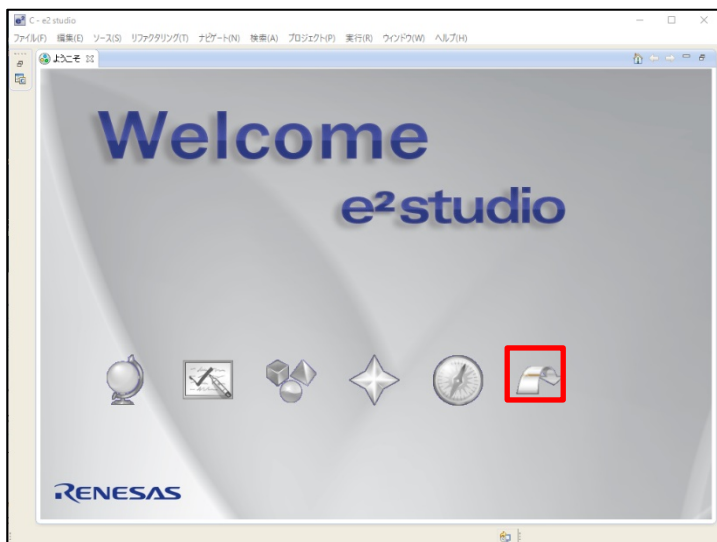
- 2) ワークスペース・ランチャーが表示されるので、プロジェクトを格納するフォルダ（例：C:\workspace）をダイアログボックスで指定し、「OK」をクリックします。



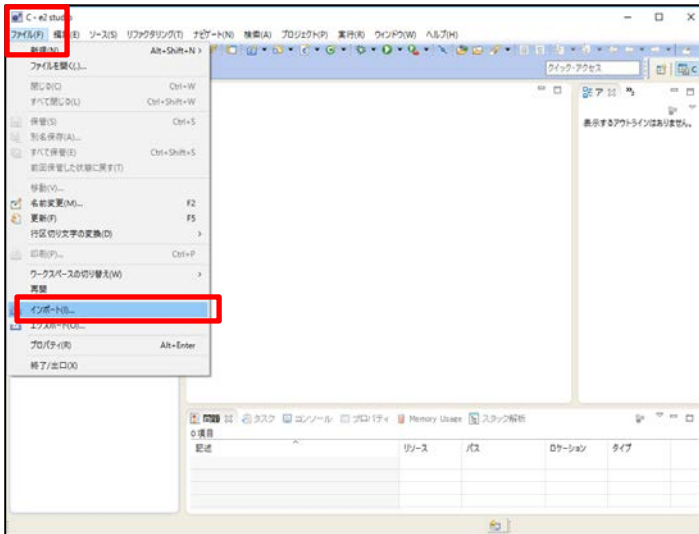
- 3) ツールチェーンの登録が表示された場合、「キャンセル」をクリックしてください。



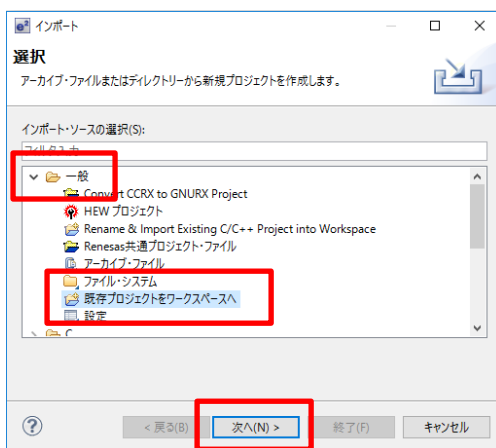
- 4) 「Workbench」アイコンをクリックしてください。



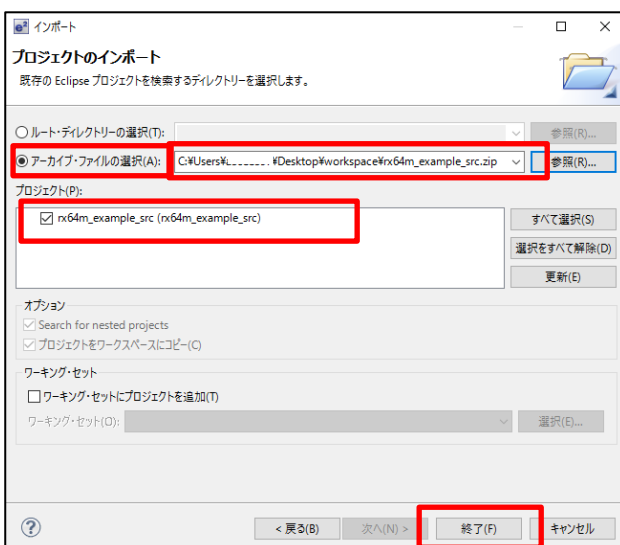
5) 「ファイル(F)」プルダウンの「インポート(I)...」をクリックしてください。



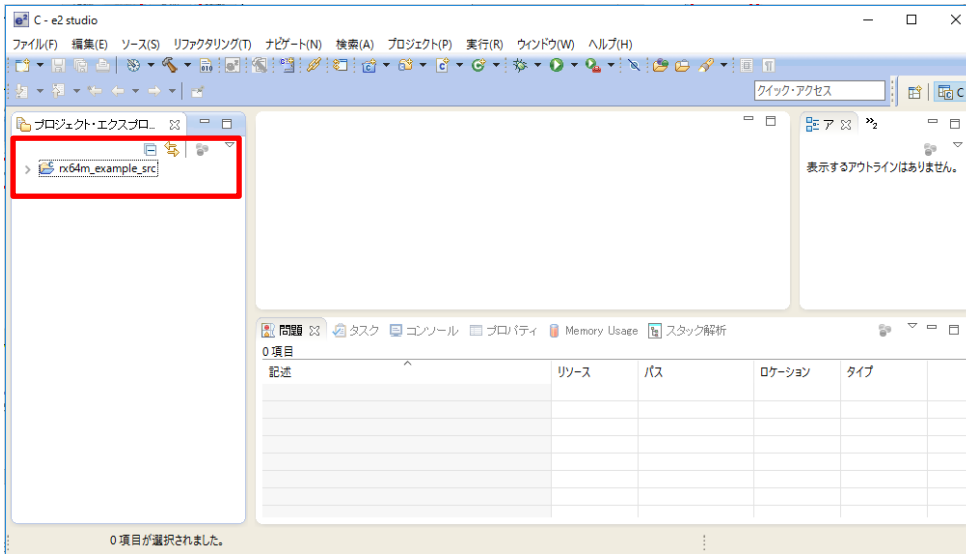
6) 「一般」の「既存プロジェクトをワークスペースへ」を選択し「次へ(N) >」をクリックしてください。



7) 「アーカイブファイルの選択」を選択しアーカイブファイル「rx64m_example_src.zip」を指定、「rx64m_example_src (rx64m_example_src)」にチェックを入れ、「終了(F)」をクリックします。



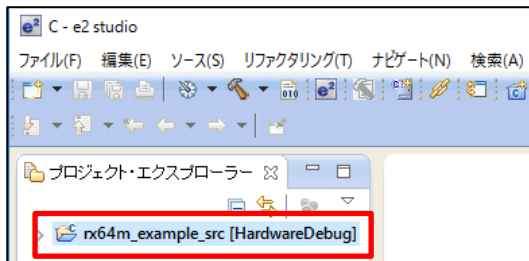
- 8) プロジェクト「rx64m_example_src」がインポートされ、プロジェクトエクスプローラペインに表示されます。



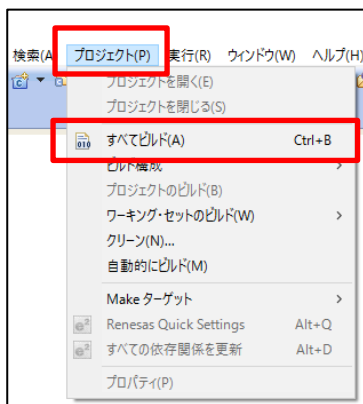
3.2.2. サンプルプロジェクトのビルド

ここでは、例としてプロジェクト「rx64m_example_src」をビルドする方法を示します。

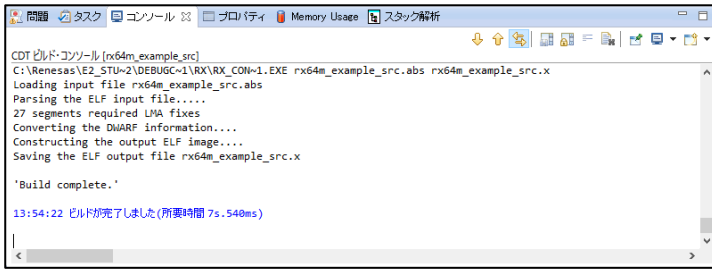
- 1) rx64m_example_src [HardwareDebug] をクリックします。



- 2) プルダウン「プロジェクト(P)」の[すべてビルド(A)]をクリックします。



3) ビルドが完了すると、「ビルドが完了しました」がコンソールに表示されます。



```
CDT ビルド:コンソール [rx64m_example_src]
C:\Renesas\E2_STU~2\DEBUGC~1\RX\RX_CON=1.EXE rx64m_example_src.abs rx64m_example_src.x
Loading input file rx64m_example_src.abs
Parsing the ELF input file.....
27 segments required LMA fixes
Converting the DWARF information....
Constructing the output ELF image....
Saving the ELF output file rx64m_example_src.x

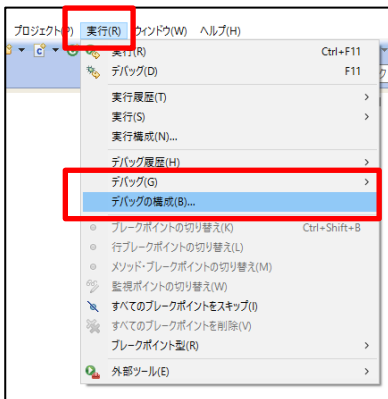
'Build complete.'
```

13:54:22 ビルドが完了しました(所要時間 7s.540ms)

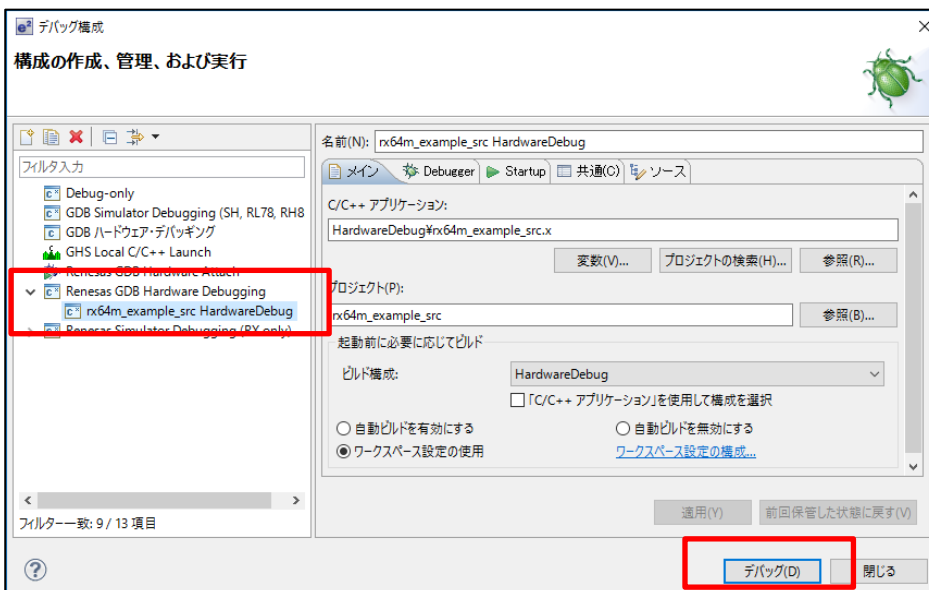
3.2.3. ターゲットシステムとの接続、ダウンロードおよびプログラムの実行

ここでは、e2studio をターゲットシステムと接続、ロードモジュールをダウンロードし、「rx64m_example_src」を実行する方法を説明します。

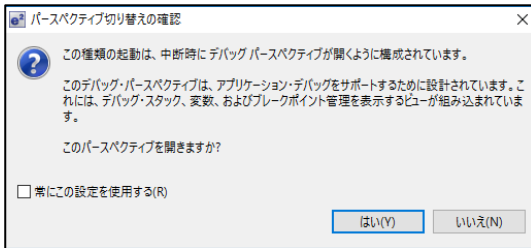
1) プルダウン「実行(R)」の「デバッグの構成(B)...」をクリックします。



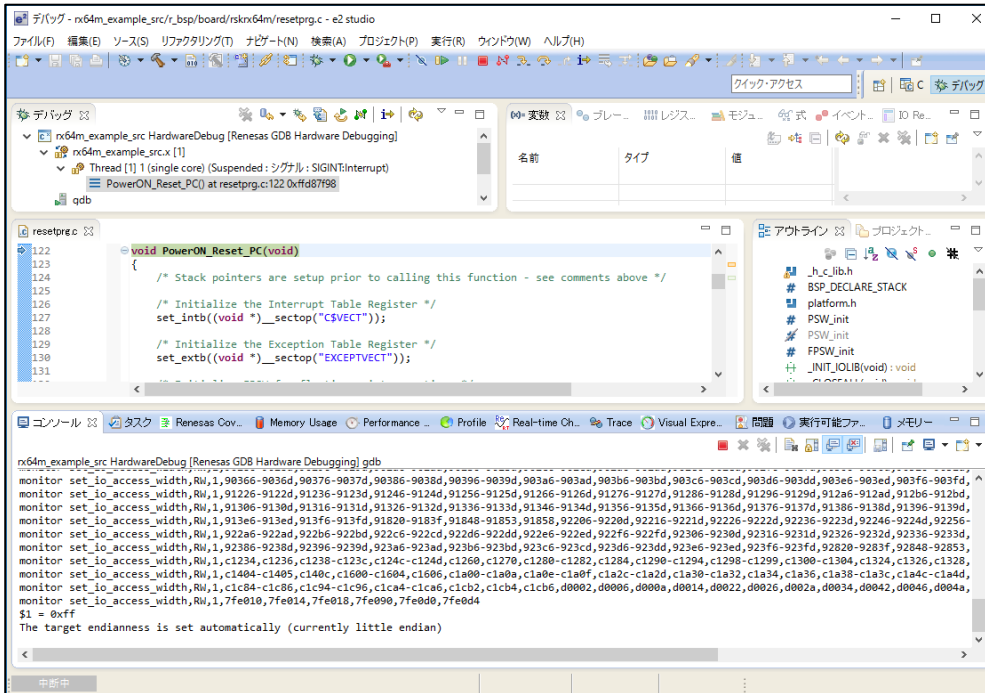
2) 「Renesas GDB Hardware Debugging」の「rx64m_example_src HardwareDebug」を選択し、「デバッグ(D)」をクリックします。



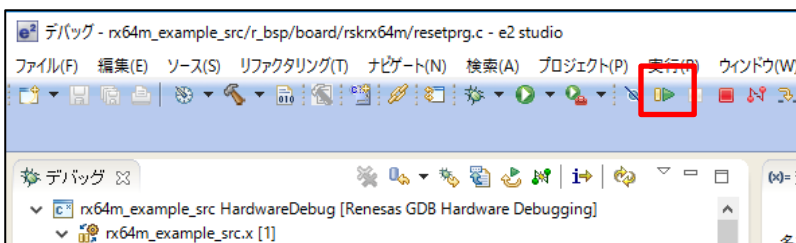
- 3) パースペクティブの切り替えを確認するメッセージが表示されるので「はい(Y)」をクリックします。



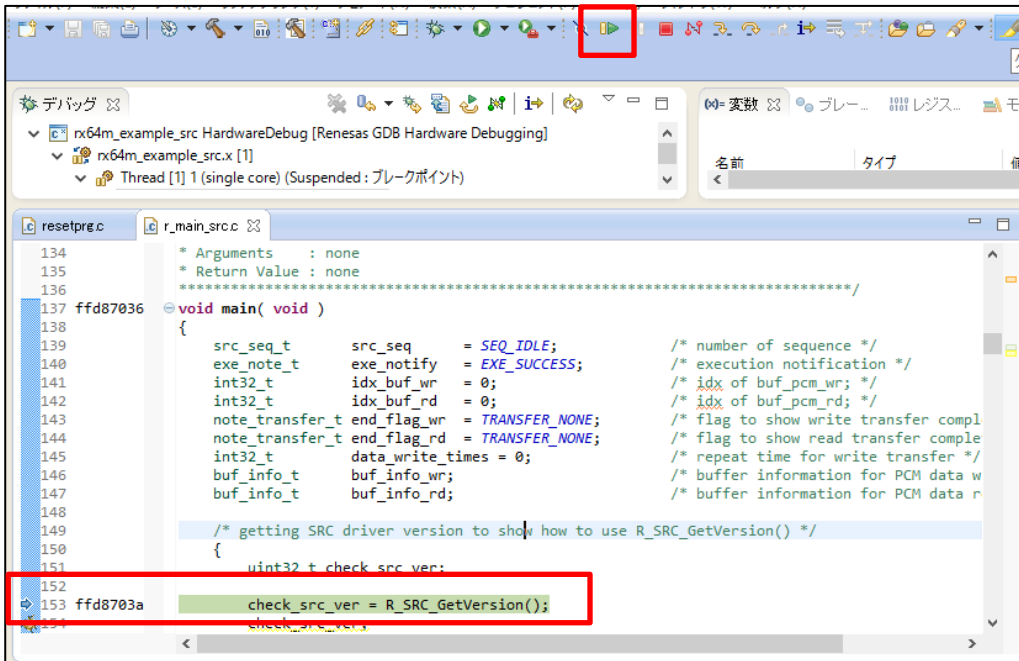
- 4) デバッグパースペクティブに切り替わります。



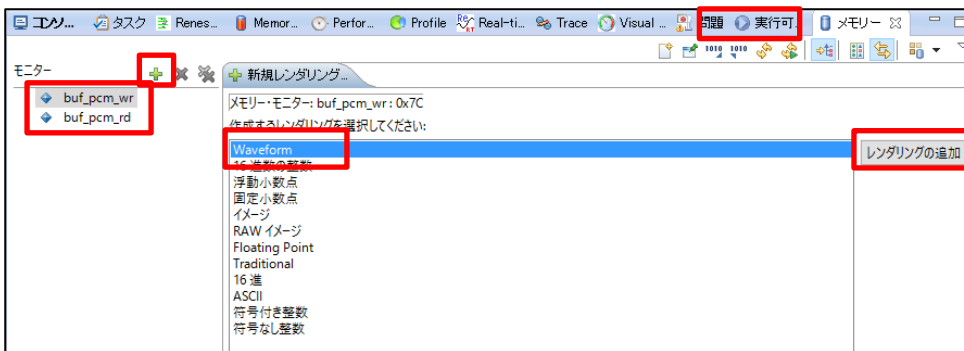
- 5) レジュームアイコンをクリックしサンプルプログラムを実行します。



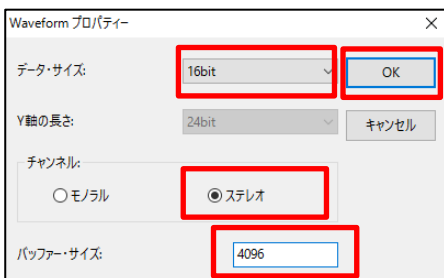
- 6) main()の最初の行で停止するので、レジュームアイコンを再度クリックし実行再開します。サンプリングレート変換処理が開始され、変換後のPCMデータがPCM読み出しバッファに格納されます。



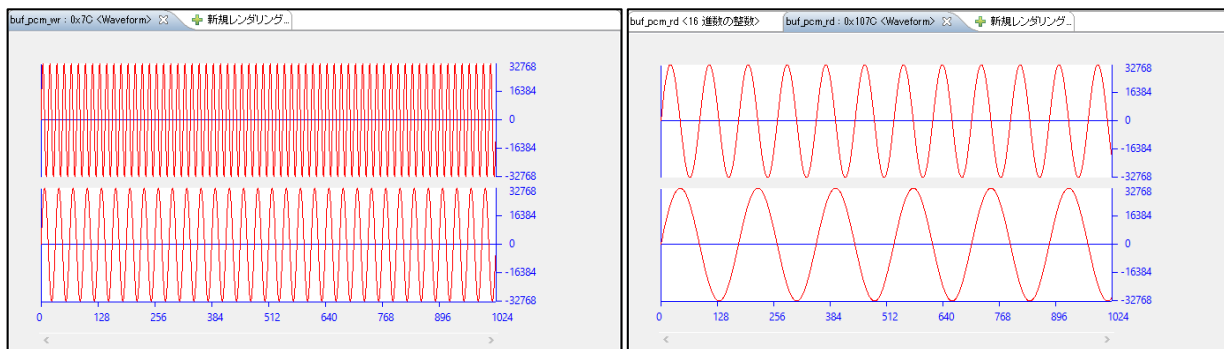
- 7) サンプリングレート変換前後のPCMデータはPCM書き込みバッファ、PCM読み出しバッファに格納されます。「メモリビュー」の波形表示機能を使用し、変換前後のPCMデータの波形を見ることができます。「+」をクリックし「buf_pcm_wr」、「buf_pcm_rd」をモニターに加えます。「Wavehome」を選択し「レンダリングの追加」をクリックします。



- 8) 「Waveform プロパティ」は以下のように設定します。



9) サンプリングレート変換前、変換後の PCM データの波形が以下のように表示されます。



4. 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 SRC モジュール Firmware Integration Technology (R01AN2090)

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)

(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)

(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。)

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
1.00	2015/6/5	-	初版発行
	2017/6/1	-	日本語版発行

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 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くと、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まれます。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、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。
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、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。
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にご使用ください。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。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。
10. 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。
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。

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