

## RX ファミリ

R20AN0291JJ0101

Rev.1.01

音声録音・再生システム(独自 ADPCM コーデック) M3S-S2-Tiny モジュール 2016.10.01

を用いたサンプルプログラム

Firmware Integration Technology

### 要旨

本資料は、ADPCM エンコーダ/デコーダライブラリ(以下、S2 ライブラリ)のサンプルプログラムの情報を示します。

サンプルプログラムは各種 Renesas Starter Kit 上で動作します。音声再生・録音機能の実現のためには Renesas Starter Kit に対応したミドルウェア評価ボードが必要です。ミドルウェア評価ボードの詳細については[2章 開発環境]を参照してください。

### 動作確認デバイス

RX64M グループ、RX610 グループ、RX210 グループ、RX111 グループ、RX65N グループ

## 目次

1. アプリケーションノート構成 .....	3
2. 開発環境 .....	4
3. サンプルプログラム .....	5
3.1 操作方法 .....	5
3.2 動作環境 .....	5
3.3 FIT モジュール環境 .....	6
3.3.1 使用する FIT モジュール .....	6
3.3.2 CS+プロジェクトへの変換方法 .....	6
3.4 SW2 押下による音声再生プログラム .....	7
3.4.1 概要 .....	7
3.4.2 フロー .....	8
3.4.3 関数リスト .....	9
3.4.4 関数チャート .....	11
3.4.5 音声データ変更手順 .....	12
3.4.6 10 ビット音声データ .....	13
3.4.7 PWM 出力方法 .....	14
3.5 SW1 押下による音声録音プログラム .....	20
3.5.1 概要 .....	20
3.5.2 フロー .....	20
3.5.3 関数リスト .....	21
3.5.4 関数チャート .....	22
3.5.5 ADPCM データ取得手順 .....	23
3.5.6 サンプルプログラムの変更 .....	23
3.5.7 ADPCM データの検証 .....	23
4. ソフトウェア更新履歴 .....	23

## 1. アプリケーションノート構成

本アプリケーションノートは、以下のものから構成されています。

表 1 S2 ライブラリサンプルプログラムのアプリケーションノート構成

構成	内容
r20an0291jj0101.pdf	本書
readme_r22j.txt	サンプルプログラムの情報
workspace	
サンプルプログラム(sample)	
rx64m_rsk	RX64M 用サンプルプログラムディレクトリ (e <sup>2</sup> studio) (※)
rx610_rsk	RX610 用サンプルプログラムディレクトリ (e <sup>2</sup> studio) (※)
rx210_rsk	RX210 用サンプルプログラムディレクトリ (e <sup>2</sup> studio) (※)
rx111_rsk	RX111 用サンプルプログラムディレクトリ (e <sup>2</sup> studio) (※)
rx65n_rsk	RX65N 用サンプルプログラムディレクトリ (e <sup>2</sup> studio) (※)

(※) e<sup>2</sup> studio プロジェクトは CS+へ変換して動作することが可能です。

## 2. 開発環境

弊社の開発環境を以下に示します。

ユーザアプリケーション開発時は以下のバージョンより新しいものをご使用下さい。

### [ソフトウェアツール]

統合開発環境

e<sup>2</sup> studio V5.2

コーディングツール

Renesas RXC Toolchain V2.04.01

### [デバッグツール]

エミュレータデバッガ

E1 emulator

エミュレータソフトウェア

RX E1/E20 エミュレータデバッガ V.1.02.00

### [評価ボード]

- Renesas Starter Kit+ for RX64M CPU ボード (型名 : R0K50564MS100BE)
- Renesas Starter Kit for RX610 CPU ボード (型名 : R0K56108C000BE)
- Renesas Starter Kit for RX210 CPU ボード (型名 : R0K52108C000BE)
- Renesas Starter Kit for RX111 CPU ボード (型名 : R0K505111S000BE#WS)
- Renesas Starter Kit+ for RX65N CPU ボード (型名 : RTK5005651C01000BR)

ミドルウェア評価ボード

ルネサス製

ミドルウェア評価ボードのマニュアルは Renesas Web サイトより入手してください。

[http://japan.renesas.com/products/mpumcu/rx/Application\\_Notes.jsp](http://japan.renesas.com/products/mpumcu/rx/Application_Notes.jsp)

ドキュメント No : R21AN0004JJ

ドキュメントタイトル : RX600 シリーズ Renesas Starter Kit for RX610

ミドルウェア評価ボードの使用方法

音声出力アンプ

(株)北斗電子様 (型名 : AMP386A)

音声を再生する際の外付けの音声出力用ボードとして利用可能です。

詳細は、以下の Web サイトをご参照ください。

<http://www.hokutodenshi.co.jp/7/amp386a.htm>

### 3. サンプルプログラム

本章ではサンプルプログラムについて説明します。サンプルプログラムは e<sup>2</sup> studio 形式のワークスペース  
になっています。マイコンと周辺機器の初期化プログラムをご使用のシステムに合わせて修正してくださ  
い。

#### 3.1 操作方法

メイン関数では、ボード上のスイッチ(SW)の入力を待ちます。  
SW 押下時の動作は以下の表のとおりです。

表 2 SW 押下時の操作

操作	内容
SW2 押下 (3.3 章 SW2 押下による音声再生プログラムを参 照)	音声再生 ROM データに記録されている ADPCM をデコー ドして PWM 出力または D/A 出力で再生します。
SW1 押下 (3.4 章 SW1 押下による音声録音プログラムを参 照)	音声録音 A/D 入力から取得した PCM データを ADPCM データにエンコードして RAM に保存します。

#### 3.2 動作環境

サンプルプログラムは各種 Renesas Starter Kit に同梱されている CPU Board(以降、CPU ボード)で動作しま  
す。  
各サンプルプログラム構成は以下の表のとおりです。

表 3 サンプルプログラム構成

サンプルプログラム	PWM 出力信号(※)	DA 出力信号(※)	入力アナログ信号
RX64M	P24(MTIOC4A)	P03(DA0) 12bitD/A	P43(AN003) 12bitA/D
RX610	P21(TIOCA3)	P66(DA0) 10bitD/A	P40(AN0) 10bitA/D
RX210	P24(TIOC4A)	P03(DA0) 10bitD/A	P45(AN005) 12bitA/D
RX111	P17(MTIOC3A)	P03(DA0) 8bitD/A	P43(AN003) 12bitA/D
RX65N	P24(MTIOC4A)	P03(DA0) 12bitD/A	P43(AN003) 12bitA/D

※PWM 出力と D/A 出力はサンプルプログラム上で切り替えます。

ユーザはこれらの端子をミドルウェア評価ボードと接続する必要があります。これらの端子は CPU ボー  
ドのアプリケーションヘッダとマイクロコントローラヘッダを使用してアクセスすることができます。詳細  
はミドルウェア評価ボードのアプリケーションノートを参照してください。

### 3.3 FIT モジュール環境

サンプルプログラム は、e<sup>2</sup> studio に含まれている FIT モジュール組み込み機能を用いてプロジェクトを作成しています。

#### 3.3.1 使用する FIT モジュール

使用している FIT モジュールは以下の通りです。

RX ファミリ 音声録音・再生システム(独自 ADPCM コーデック) M3S-S2-Tiny モジュール  
Firmware Integration Technology

(R20AN0037)

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)

上記 FIT モジュールの詳細は、各 FIT モジュールに含まれるドキュメントを参照して下さい。

#### 3.3.2 CS+プロジェクトへの変換方法

e<sup>2</sup> studio のプロジェクトは、プロジェクト内に含まれている rpc ファイルを用いて、CS+プロジェクトへコンバートすることができます。以下、rx64m\_rsk を例に説明します。

変換方法を以下に示します。

1. CS+を起動し、「e<sup>2</sup> studio / CubeSuite / …」の「GO」ボタンを押します。
2. 「e<sup>2</sup> studio プロジェクト・ファイル(\*.rpc)」を選択して、\*.rpc ファイルを開きます。
3. 「プロジェクト変換設定」ウインドウが開き、ツリー上でプロジェクトを選択します。
4. ツリーの右側のプロジェクト設定で、使用するマイクロコントローラを「RX64M」->「R5F564MLDxFC」を選択して「OK」を押します。CS+は変換されたプロジェクトを出力します。
5. 「プロジェクトツリー」から「CC-RX」を選択します。
6. 「共通オプション」タブ->の「CPU」->「命令セット・アーキテクチャ」を「RXv2 アーキテクチャ」に設定します。
7. 「リンク・オプション」タブ->の「入力」->「バイナリ・データ・ファイル」の設定を以下のように変更します。

[変更前]

```
%ProjectFolder%/../%ProjectName%/adpcm/sound_adpcm. dat (ADPCM_ADDR1)
```

```
%MainProjectDir%/../%ProjectName%/adpcm/sound_adpcm. dat (ADPCM_ADDR1)
```

[変更後]

不要な"}"と不要なパスを削除します。

```
%ProjectFolder%/../%ProjectName%/adpcm/sound_adpcm. dat (ADPCM_ADDR1)
```

8. プロジェクトをビルドして、ビルド完了します。
9. ユーザは環境に合わせてデバッグツールの設定を行ってください。その後、ユーザはサンプルプログラムの動作を確認することができます。

## 3.4 SW2 押下による音声再生プログラム

### 3.4.1 概要

SW2 を押すと、音声再生されます。音声は PWM または DA 出力を使用して再生します。

decode\_main 関数により、ADPCM データが伸張され、PCM データをリングバッファに格納します。タイマ割り込みによってリングバッファ内の PCM/DA データを出力します。この処理はすべての ADPCM データが出力されるまで続きます。PCM データは  $90.7 \mu s$  間隔でタイマ割り込みによってバッファから取り出され、PWM 出力に転送されます。音声データのサンプリング周波数は 11.025kHz です。

**【注】**  $90.7 \mu s \approx 11.025kHz$

音声データのサイズと記憶領域は、以下のマクロで定義します。

音声データのサイズ (ADPCM データサイズの 2 倍) :マクロ名 PCM\_DATA\_SIZE1

記憶領域の開始アドレス :マクロ名 ADPCM\_ADDR1

D/A、PWM 出力の選択は、r\_s2\_driver.h の SOUND\_OUTPUT\_MODULE マクロに対して以下のどちらかを定義することにより決定します。

D/A で音声出力する場合 :SOUND\_DA

PWM で音声出力する場合 :SOUND\_PWM

このサンプルプログラムは、PCM データバッファアンダーランを防止するためにメインループで PCM データのバッファリングをしています。これは実用的なユーザシステムではありません。ユーザは必要に応じて、定期的な割り込みまたは RTOS による PCM データのバッファリングを実装してください。

3.4.2 フロー

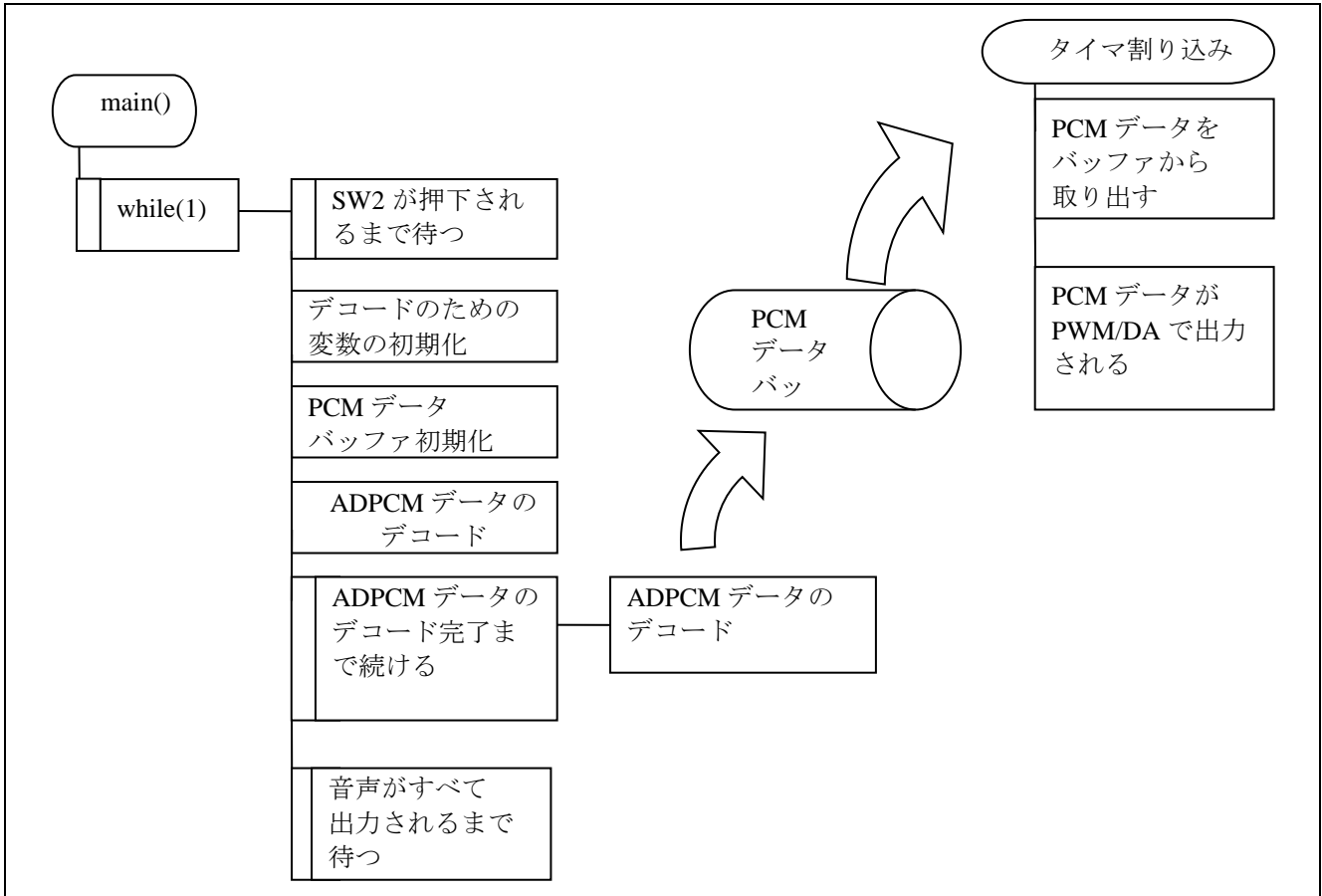


図 1 フロー



### 3.4.3 関数リスト

表 4 関数リスト

No.	関数名	概要
1	main	メイン関数です。
1.1	R_StartSelect	スイッチの入力を判定します。
1.2	R_decode_main	ROM 上にある ADPCM データをデコードし、PCM データをリングバッファに格納します。
1.2.1	decode_interval_timer_init	サンプリング周波数の周期タイマの初期設定を行います。
1.2.2	SOUND_MODULE_INIT(※)	PWM/DA 出力の初期設定を行います。
1.2.3	R_InitDeclInfo	デコード情報構造(DECINFO)を初期化します。
1.2.3.1	R_adpcm_initDec	デコードするための作業領域を初期化します。 - ライブラリ関数です。
1.2.4	R_InitRingBuffer	デコードデータを格納するためのリングバッファを初期化します。
1.2.5	R_DecodeProc	ADPCM データを PCM データに変換し、デコードされたデータをリングバッファに格納します。ADPCM 入力ポイントがここで更新されます。
1.2.5.1	R_adpcm_RefreshDec	入力 ADPCM データアドレスと出力 PCM データアドレスを更新します。 - ライブラリ関数です。
1.2.5.2	R_adpcm_decode	4 ビット ADPCM データを 16 ビット PCM データに変換します。 - ライブラリ関数です。
1.2.5.3	R_RingBufferSetData	データを書き込む前にリングバッファの空きをチェックします。
1.2.5.3.1	R_RingBufferPush	デコードされた PCM データをリングバッファに格納します。
1.2.5.3.1.1	SOUND_OUTPUT_DATA_CNV(※)	PCM データ(16 ビット)を PWM 出力データに変換します。
1.2.6	SOUND_MODULE_START(※)	PWM/DA 出力を開始します。
1.2.7	decode_interval_timer_start	サンプリング周波数の周期タイマを開始します。
1.2.8	SOUND_MODULE_STOP(※)	PWM/DA 出力を停止します。
1.2.9	decode_interval_timer_stop	サンプリング周波数の周期タイマを停止します。
1.2.10	SOUND_MODULE_SLEEP(※)	PWM/DA 出力を終了します。
1.2.11	decode_interval_timer_sleep	サンプリング周波数の周期タイマを終了します。
2.	R_interrupt_decode_timer	タイマ割り込みです。PWM を使用してサンプリング周波数のサイクルごとにデコードされた PCM データを出力します。
2.1.	R_SetPCMdata	音声データを PWM/DA で出力します。

(※) 関数を定義したマクロです。SOUND\_OUTPUT\_MODULE マクロの設定値により実行する関数が変わります。

表 5 関数マクロ一覧

関数マクロ名	SOUND_OUTPUT_MODULE が SOUND_DA を定義している場合	SOUND_OUTPUT_MODULE が SOUND_PWM を定義している場合
SOUND_MODULE_INIT	decode_da_converter_init	decode_pwm_init
SOUND_OUTPUT_DATA_CNV	R_Convert16to10(RX610, RX210) R_Convert16to12(RX64M, RX65N) R_Convert16to8(RX111)	R_Convert16toPWM
SOUND_MODULE_START	decode_da_converter_start	decode_pwm_start
SOUND_MODULE_STOP	decode_da_converter_stop	decode_pwm_stop
SOUND_MODULE_SLEEP	decode_da_converter_sleep	decode_pwm_sleep

3.4.4 関数チャート

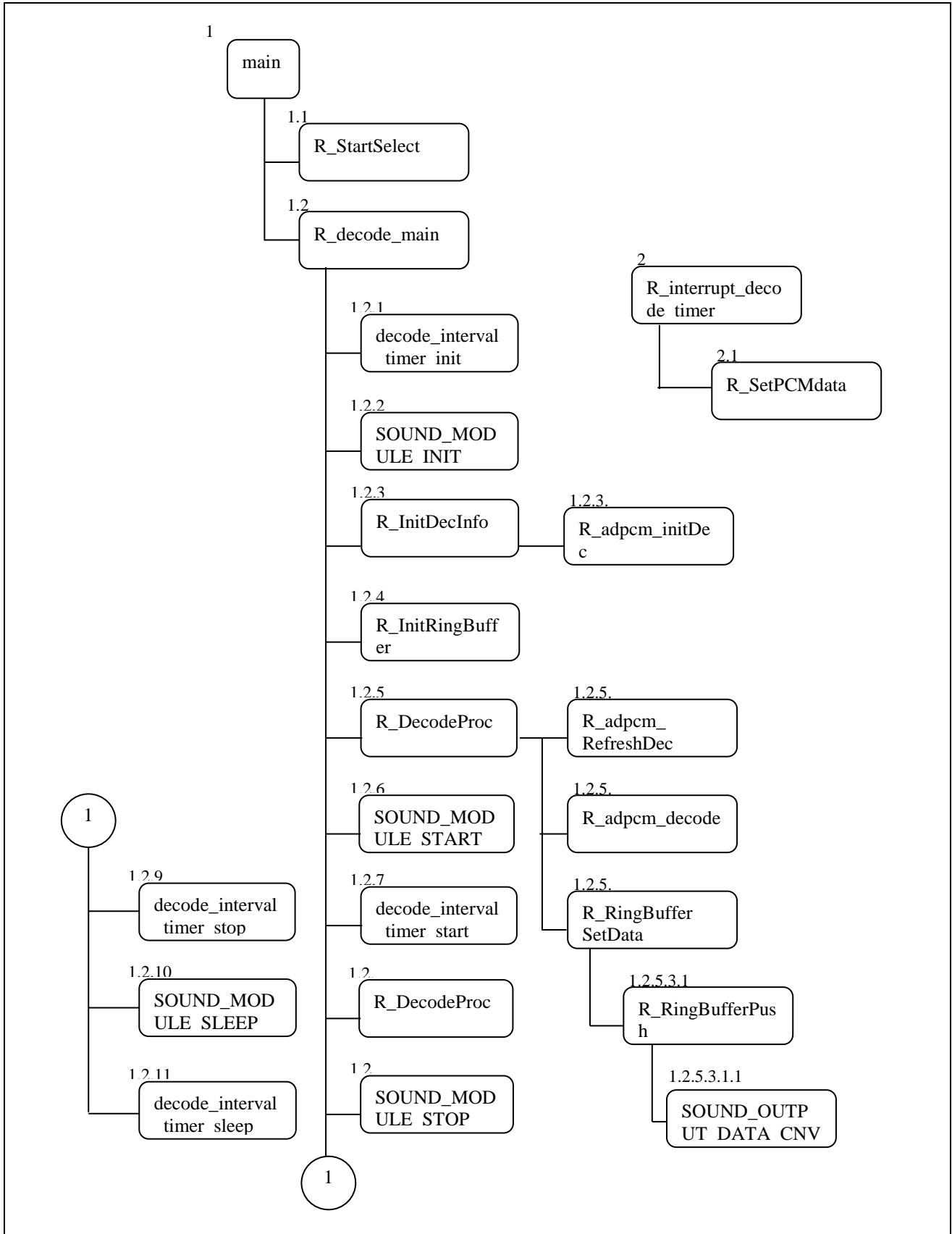


図 2 関数チャート

### 3.4.5 音声データ変更手順

サンプルプログラムにはサンプルの音声データが含まれます。この音声データをユーザ作成のものに変更する手順を以下に示します。

#### (1) 音声データの作成

以下の表に示す仕様に従って音声データを作成します。

表 6 音声データ仕様

量子化ビット	16 ビット
チャンネル	モノラル
サンプリング周波数	11.025kHz
ファイル形式	WAV

#### (2) 音声データの圧縮

ADPCM\_TOOL(ADPCM.exe)を使用して Wave(.wav)ファイルから ADPCM データ(バイナリ形式)を作成します。使用方法については、ツール"ADPCM\_TOOL"のマニュアルを参照してください。

#### (3) サンプルプログラムの変更

##### ・ PCM\_DATA\_SIZE1

マクロ PCM\_DATA\_SIZE1 の値は、サンプルプログラムの"r\_s2\_driver.h"ファイルに設定しています。この値は、使用する音声データのサイズに従って設定します。設定値は ADPCM データのサイズの 2 倍の値を設定してください。

[たとえば、ADPCM データのサイズが 0x1ef1 バイトである場合は、設定は次のようになります。]

```
#define PCM_DATA_SIZE1 0x3de2 /* size of sound data after decode(compressed data) */
```

##### ・ ADPCM\_ADDR1

ADPCM データの位置に従ってサンプルプログラムの"r\_s2\_driver.h"ファイルにアドレス ADPCM\_ADDR1 の値を設定します。

[ADPCM データが 0xFFFF80000 番地から始まる場合]

```
#define ADPCM_ADDR1 0xFFFF8000 /* top address of ADPCM sound data storage */
```

#### (4) 音声データのダウンロード

ADPCM データを ADPCM\_ADDR1 に設定されたアドレスにダウンロードします。ADPCM データは ADPCM\_TOOL を使用して作成したデータです。ダウンロード手順については、統合開発環境のマニュアルを参照してください。

### 3.4.6 10 ビット音声データ

R\_adpcm\_decode 関数によって伸張された音声データの長さは 16 ビットです。ご使用のシステムに従って、音声データのビット長を変更してください。RX610、RX210 のサンプルプログラムでは、R\_Convert16to10()関数を使用して 16 ビット長の PCM データを 10 ビット長に変換します。(10 ビット D/A コンバータなどを使用するときに、これを使用することができます。)

引数：変換前の 16 ビット PCM データ

戻り値：変換後の 10 ビット D/A データ

```
return (uint16_t)(((data + 0x20) >> 6) + 512);
```

### 3.4.7 PWM 出力方法

PWM の設定および動作方法について説明します。

- <仕様>

タイマを 2 チャンネル使用し、周期タイマ用、PWM 機能用として使用します。周期タイマの割り込みでは PWM 機能用タイマのタイマジェネラルレジスタに PCM データから計算されたデューティ値を設定します。

デューティ値は以下の計算式で求めます。(r\_s2\_decode.c 関数名: R\_Convert16toPWM)

```
return (uint16_t)((((int32_t)( data + 32768 ) * PWM_SAMPLING) >> 16);
```

— 計算式の意味

— ・ +32768 の意味

入力値(data)の補正。入力値(data)は int16\_t 型、出力値は uint16\_t 型

— ・ PWM\_SAMPLING と >>16 の意味

入力値(data): 0xffff(タイマカウント最大値) = x: PWM\_SAMPLING(100kHz を数える為のタイマカウント値)

$x = (\text{入力値}(\text{data}) * \text{PWM\_SAMPLING}) / 0xffff$

$= (\text{入力値}(\text{data}) * \text{PWM\_SAMPLING}) \gg 16;$

(1) [RX64M の場合]

- <利用資源>
- マルチファンクションタイマパルスユニット 3(MTU3a) 2ch(MTU1,MTU4)  
出力端子 1 本( MTIOC4A )

表 7 PWM 出力方法

チャンネル	項目	内容
1	用途	音声データのサンプリング周期生成
	端子出力制御	なし
	動作モード	通常動作
	動作クロック	120MHz
	周期	音声のサンプリング周波数(≒11.025kHz) TGRA レジスタとのコンペアマッチを周期とする
	割り込み	使用する
4	用途	PWM 機能
	端子出力制御	あり(MTIOC4A)
	動作モード	PWM モード 1 TGRA レジスタと TGRC レジスタはバッファ動作
	動作クロック	120MHz
	PWM 周期	100kHz(チャンネル 1 より短い周期) = PWM_SAMPLING TRGB とのコンペアマッチを PWM 周期とする
	PWM 出力設定	TRGA とのコンペアマッチで"L"出力、 TRGB とのコンペアマッチで"H"出力
	TRGC から TRGA への 転送タイミング	TCNT レジスタのクリア(=TRGB とのコンペアマッチ)
	デューティ値の更新タイミング	チャンネル 1 割り込み処理で更新
	割り込み	使用しない

<制限事項>

CPU ボードで RX64M のサンプルプログラムの D/A 出力の動作を確認する場合は、CPU ボードの 0Ω 抵抗の実装を変更する必要があります。

表 8 実装変更

P03 信号切替	信号名	抵抗	
		R189	R277
D/A 出力の動作確認時の実装	DA0	実装する	実装しない
デフォルト実装	LED0	実装しない	実装する

(2) [RX610 の場合]

- <利用資源>
- 16 ビットタイマパルスユニット(TPU) 2ch(TPU1,TPU3)  
出力端子 1 本(TIOCA3)

表 9 PWM 出力方法

チャンネル	項目	内容
1	用途	音声データのサンプリング周期生成
	端子出力制御	なし
	動作モード	通常動作
	動作クロック	50MHz
	周期	音声のサンプリング周波数(≒11.025kHz) TGRA レジスタとのコンペアマッチを周期とする
	割り込み	使用する
3	用途	PWM 機能
	端子出力制御	あり(TIOCA3)
	動作モード	PWM モード 1 TGRA レジスタと TGRC レジスタはバッファ動作
	動作クロック	50MHz
	PWM 周期	100kHz(チャンネル 1 より短い周期) = PWM_SAMPLING TRGB とのコンペアマッチを PWM 周期とする
	PWM 出力設定	TRGA とのコンペアマッチで"L"出力、 TRGB とのコンペアマッチで"H"出力
	TRGC から TRGA への 転送タイミング	TCNT と TRGA のコンペアマッチ
	デューティ値の更新タイミング	チャンネル 1 割り込み処理で更新
	割り込み	使用しない



(3) [RX210 の場合]

- <利用資源>
- マルチファンクションタイマパルスユニット 2(MTU2a) 2ch(MTU1,MTU4)  
出力端子 1 本(TIOC4A)

表 10 PWM 出力方法

チャンネル	項目	内容
1	用途	音声データのサンプリング周期生成
	端子出力制御	なし
	動作モード	通常動作
	動作クロック	25MHz
	周期	音声のサンプリング周波数(≒11.025kHz) TGRA レジスタとのコンペアマッチを周期とする
	割り込み	使用する
4	用途	PWM 機能
	端子出力制御	あり(TIOC4A)
	動作モード	PWM モード 1 TGRA レジスタと TGRC レジスタはバッファ動作
	動作クロック	25MHz
	PWM 周期	100kHz(チャンネル 1 より短い周期) = PWM_SAMPLING TRGB とのコンペアマッチを PWM 周期とする
	PWM 出力設定	TRGA とのコンペアマッチで"L"出力、 TRGB とのコンペアマッチで"H"出力
	TRGC から TRGA への 転送タイミング	TCNT レジスタのクリア(=TRGB とのコンペアマッチ)
	デューティ値の更新タイミング	チャンネル 1 割り込み処理で更新
	割り込み	使用しない

(4) [RX111 の場合]

- <利用資源>
- マルチファンクションタイマパルスユニット 2(MTU2a) 2ch(MTU1,MTU3)  
 出力端子 1 本( MTIOC3A )

表 11 PWM 出力方法

チャンネル	項目	内容
1	用途	音声データのサンプリング周期生成
	端子出力制御	なし
	動作モード	通常動作
	動作クロック	24MHz
	周期	音声のサンプリング周波数(≒11.025kHz) TGRA レジスタとのコンペアマッチを周期とする
	割り込み	使用する
3	用途	PWM 機能
	端子出力制御	あり(MTIOC3A)
	動作モード	PWM モード 1 TGRA レジスタと TGRC レジスタはバッファ動作
	動作クロック	24MHz
	PWM 周期	100kHz(チャンネル 1 より短い周期) = PWM_SAMPLING TRGB とのコンペアマッチを PWM 周期とする
	PWM 出力設定	TRGA とのコンペアマッチで"L"出力、 TRGB とのコンペアマッチで"H"出力
	TRGC から TRGA への 転送タイミング	TCNT レジスタのクリア(=TRGB とのコンペアマッチ)
	デューティ値の更新タイミング	チャンネル 1 割り込み処理で更新
	割り込み	使用しない

(5) [RX65N の場合]

- <利用資源>
- マルチファンクションタイマパルスユニット 3(MTU3a) 2ch(MTU1,MTU4)  
出力端子 1 本( MTIOC4A )

表 12 PWM 出力方法

チャネル	項目	内容
1	用途	音声データのサンプリング周期生成
	端子出力制御	なし
	動作モード	通常動作
	動作クロック	120MHz
	周期	音声のサンプリング周波数(≒11.025kHz) TGRA レジスタとのコンペアマッチを周期とする
	割り込み	使用する
4	用途	PWM 機能
	端子出力制御	あり(MTIOC4A)
	動作モード	PWM モード 1 TGRA レジスタと TGRC レジスタはバッファ動作
	動作クロック	120MHz
	PWM 周期	100kHz(チャネル 1 より短い周期) = PWM_SAMPLING TRGB とのコンペアマッチを PWM 周期とする
	PWM 出力設定	TRGA とのコンペアマッチで"L"出力、 TRGB とのコンペアマッチで"H"出力
	TRGC から TRGA への 転送タイミング	TCNT レジスタのクリア(=TRGB とのコンペアマッチ)
	デューティ値の更新タイミング	チャネル 1 割り込み処理で更新
	割り込み	使用しない

<制限事項>

CPU ボードで RX65N のサンプルプログラムの D/A 出力の動作を確認する場合は、CPU ボードの 0Ω 抵抗の実装を変更する必要があります。

表 13 実装変更

P03 信号切替	信号名	抵抗	
		R192	R306
D/A 出力の動作確認時の実装	DA0	実装する	実装しない
デフォルト実装	LED0	実装しない	実装する

### 3.5 SW1 押下による音声録音プログラム

#### 3.5.1 概要

ユーザが SW1 を押下すると、音声を記録します。音声は AD コンバータを使用して記録します。

R\_encode\_main 関数により、エンコーダを初期化し、タイマをスタートします。その後、タイマ割り込みによって  $125 \mu s$  ごとに入力信号からサンプリングします(サンプリング周波数 8kHz)。入力データは ADPCM 変換のために S2 ライブラリ関数に渡され、エンコードされた ADPCM データは RAM に格納されます。エンコードは ADPCM データが MAX\_DATA\_LENGTH に達するまで行います。

ADPCM データのサイズは以下のマクロで定義されます。

ADPCM データのサイズ      マクロ名 : MAX\_DATA\_LENGTH

#### 3.5.2 フロー

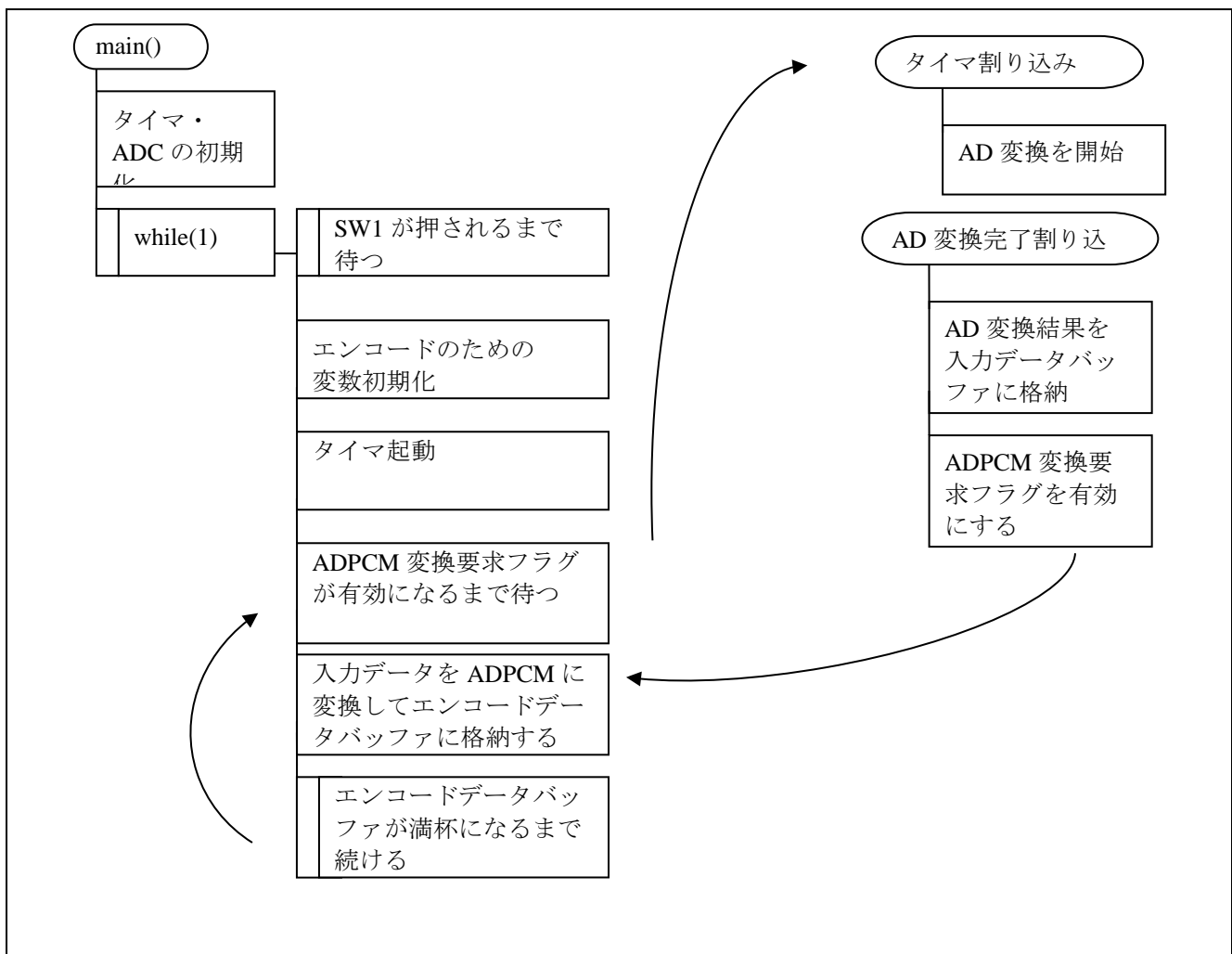


図 3 フロー

### 3.5.3 関数リスト

表 14 関数リスト

No.	関数名	概要
1	main	メイン関数です。
1.1	R_StartSelect	スイッチの入力を判定します。
1.3	R_encode_main	入力音声データを ADPCM データにエンコードします。 ADPCM データは RAM に蓄積されます。
1.3.1	R_adpcm_initEnc	エンコードするための作業領域を初期化します。 - ライブラリ関数です。
1.3.2	encode_ad_converter_init	AD コンバータの初期設定を行います。
1.3.3	encode_interval_timer_init	音声録音用のタイマの初期設定を行います。
1.3.4	encode_interval_timer_start	音声録音用のタイマを開始します。
1.3.5	R_convert_to_ADPCM	ライブラリ関数を呼び出して PCM データを ADPCM データに変換します。
1.3.5.1	R_adpcm_refreshEnc	入力 PCM データバッファアドレスと出力 ADPCM データバッファアドレスを更新します。 - ライブラリ関数です。
1.3.5.2	R_adpcm_encode	16 ビット PCM データを 4 ビット ADPCM データにエンコードします。 - ライブラリ関数です。
1.3.6	encode_interval_timer_stop	音声録音用のタイマを停止します。
1.3.7	encode_ad_converter_stop	AD 変換動作を停止します。
1.3.8	encode_interval_timer_sleep	音声録音用のタイマをモジュールストップ状態にします。
1.3.9	encode_ad_converter_sleep	AD コンバータをスリープ状態にします。
3	R_interrupt_encode_timer(※)	タイマ割り込みです。サンプリング周波数のサイクルごとに音声信号をサンプリングします。
3.1	encode_ad_converter_start	AD 変換を開始します。
4	R_interrupt_encode_ad_convert	AD 変換完了割り込みです。音声信号をサンプリングします。

(※) 本サンプルでは AD 変換はタイマ機能から自動で開始するように設定しています。タイマの割り込み設定はしていませんので本関数は実行されません。

3.5.4 関数チャート

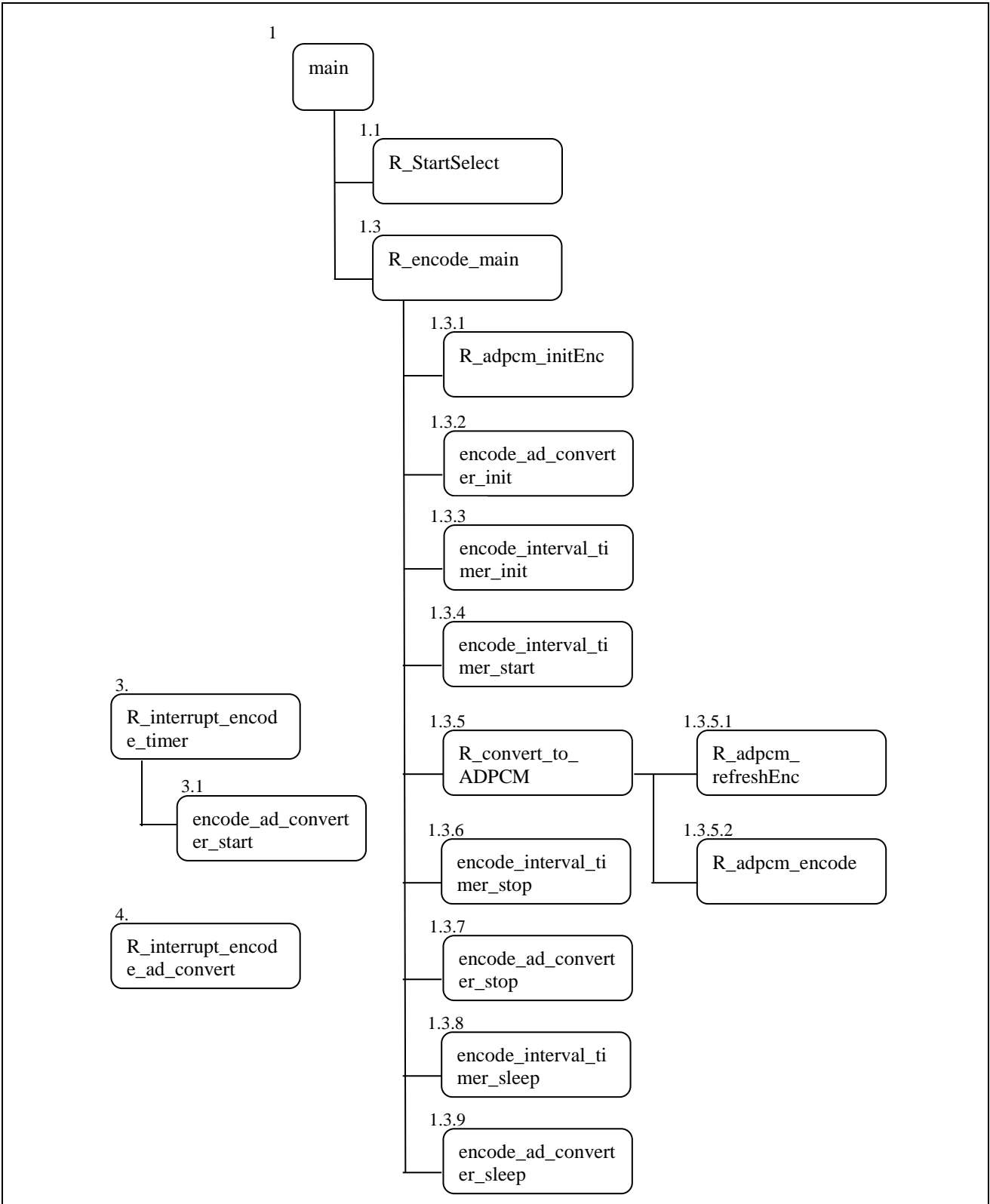


図 4 関数チャート

### 3.5.5 ADPCM データ取得手順

このセクションでは、サンプルプログラムの ADPCM データ取得手順について説明します。サンプルプログラムによってエンコードされた ADPCM データは以下の手順により取得することができます。

#### (1) ADPCM データのアップロード

エンコードデータアレイ変数 `g_EncodedData` の内容は、統合開発環境のメモリウィンドウに表示することができます。このエンコードデータアレイ全体をアップロードし、バイナリファイルとして保存します。アップロード手順については、統合開発環境のマニュアルを参照してください。

#### (2) wav ファイルへの変換

"ADPCM\_TOOL(ADPCM.exe)"を使用して ADPCM データ(バイナリ形式)から Wave ファイル(.wav)を作成します。使用方法については、ツール"ADPCM\_TOOL"のマニュアルを参照してください。

### 3.5.6 サンプルプログラムの変更

#### ・MAX\_DATA\_LENGTH

マクロ"MAX\_DATA\_LENGTH"の値は、サンプルプログラムの `r_s2_driver.h` ファイルに定義しています。この値は、エンコードされた ADPCM データを格納するために使用可能な RAM のサイズに従って設定しなければなりません。ADPCM データを格納するために使用可能なサイズが大きくなるほど、エンコード可能な音声サンプルの長さが長くなります。

[たとえば、ADPCM データを格納するために使用可能なサイズが 1500 バイトである場合は、設定は次のようになります。]

```
#define MAX_DATA_LENGTH (1500u) /* Max. size of ADPCM encoded data */
```

### 3.5.7 ADPCM データの検証

ADPCM\_TOOL を使用して、アップロードした ADPCM ファイルから生成した Wave ファイルが入力ファイルと性質が類似することを検証することができます。

## 4. ソフトウェア更新履歴

表 15 更新履歴

本書 version	S2 ライブラリ version	変更点
1.01	3.04	S2 ライブラリを更新 RX65N RSK サンプルプログラム追加
1.00	3.01	FIT モジュール化対応 RX64M RSK、RX111 RSK サンプルプログラム追加

## 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
1.01	2016.10.01	—	サンプルプログラム更新
1.00	2014.07.01	—	初版発行

## 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、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、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、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用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とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



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/>