

RX ファミリ

SSIE を使用する音声入出力サンプルプログラム

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 モジュールと RX671 SSIE 機能を使用して音声入出力プログラム動作例を説明します。

動作確認デバイス

RX671

評価基板 EK-RX671 (RTK5EK6710S00001BE)

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。

目次

1. 概要	3
1.1 システム機能	4
1.1.1 正弦波出力モード	4
1.1.2 オンボードマイク音声入力出力モード	5
1.1.3 Audio Player モード	6
2. 動作条件	7
2.1 動作モード設定	7
2.2 設定パラメータ	7
2.2.1 RX671 SSIE マスタスレーブ	8
2.2.2 サンプリングレート	8
2.2.3 量子化ビット数	8
2.2.4 正弦波周波数	8
2.2.5 音量設定	9
3. プログラム説明	10
3.1 処理フロー	10
3.1.1 全体動作フロー	10
3.1.2 正弦波出力モード処理フロー	11
3.1.3 音声入出力モード処理フロー	11
3.1.4 Audio Player モード処理フロー	12
3.2 ファイル構成	13
3.3 関数	14
3.3.1 関数構成	14
3.3.2 アプリケーション関数	15
3.3.3 Audio Codec ドライバ関数	16
3.3.4 GPIO ドライバ関数	18
3.3.5 Audio Player ライブラリ関数	19
4. FIT モジュール	20
5. プロジェクトをインポートする方法	21
5.1 e ² studio での手順	21
5.2 CS+での手順	22
6. サンプルコード	23
7. 参考ドキュメント	23

1. 概要

本アプリケーションノートは、Firmware Integration Technology (FIT) に準拠した SSI モジュールと RX671 SSIE 機能を使用して音声入出力プログラム動作例を説明します。

本プログラムの動作確認を行った評価基板 EK-RX671 ボードの回路構成を図 1.1 に示します。

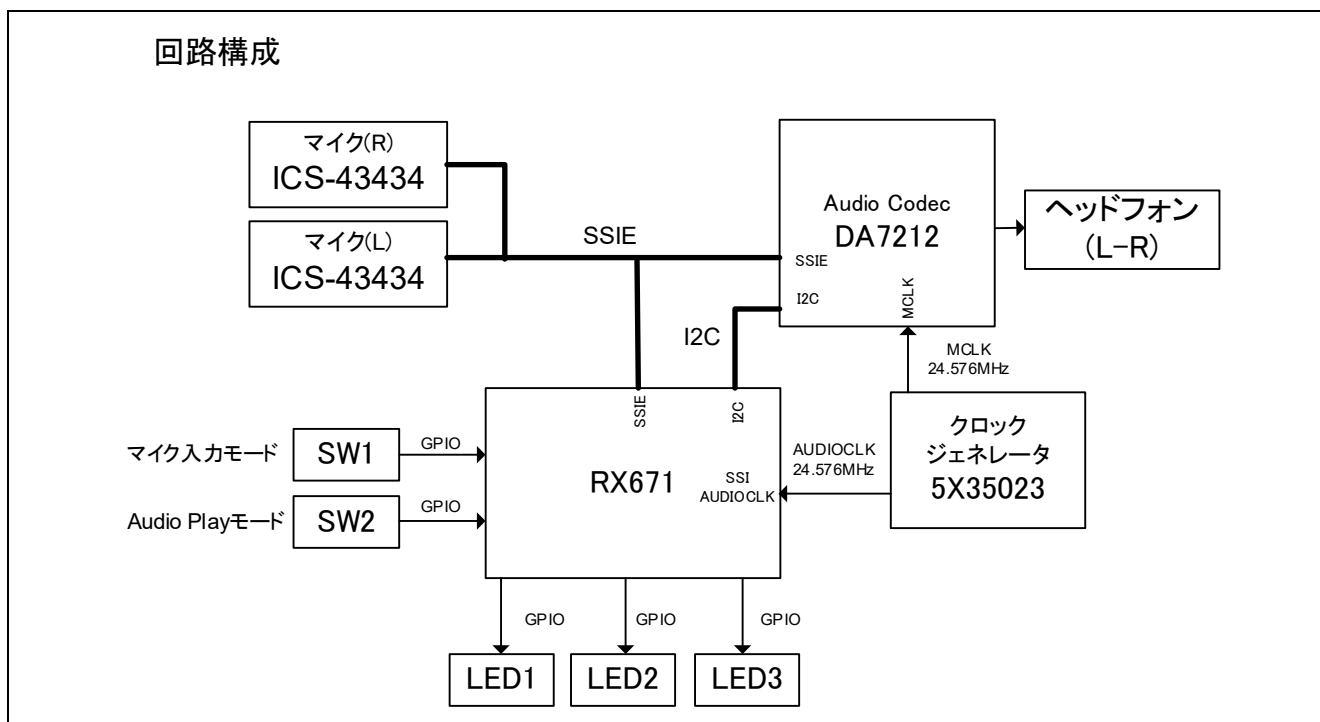


図 1.1 ブロック図

この構成図は、本プログラム動作に関する回路部分を記述しています。

評価基板 EK-RX671 全体の構成は、評価基板 EK-RX671 のマニュアルを参照してください。

EK-RX671 と本プログラムを用いることで RAM やフラッシュメモリに格納した音声データ、オンボードの MEMS Mic から取得した音声データを、AudioCodec を介してヘッドフォンへ出力することができます。

1.1 システム機能

本プログラムは、SSIE 機能を使用した下記の 3 つのモードでの動作が可能です。

- 正弦波出力モード
- オンボードマイク音声入出力モード
- Audio Player モード

1.1.1 正弦波出力モード

設定された周波数の正弦波をソフト生成し、内蔵 RAM に記録した波形データを、DMAC を用いた SSIE 通信で Audio Codec デバイスへ出力します。

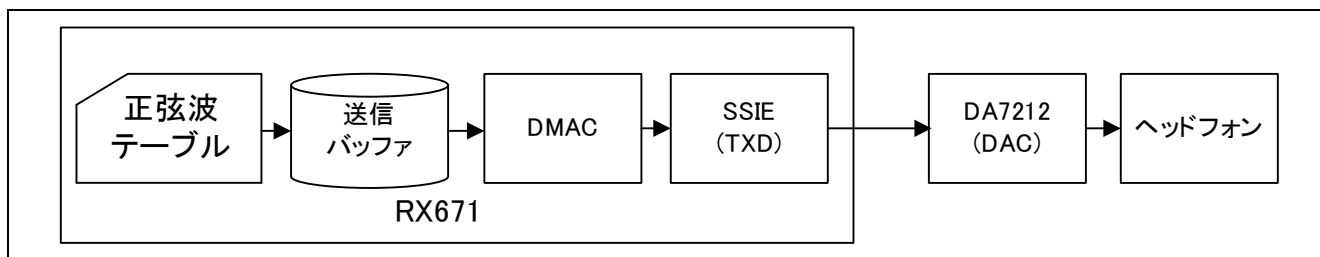


図 1.2 正弦波発生ブロック図

PCM データ転送フォーマット

- PCM データ転送フォーマット : I2S
- PCM データ : 8/16/18/20/22/24/32bit ステレオ
- サンプリングレート : RX671 マスタ 96/48/32/24/16/12/8kHz
RX671 スレーブ 96/48/44.1/32/24/22.05/16/12/11.025/8kHz

注 : 正弦波テーブル生成時の制約により、本サンプルプログラムでは 25Hz の倍数のみ対応しています。

応用アプリケーション例 : 各種警報器、機器の操作音発生、調整用オーディオ信号発生器

1.1.2 オンボードマイク音声入力出力モード

評価基板上のマイクの音声データを SSIE 通信で取り込み、DMAC で受信バッファに転送。ソフトで送信バッファにコピーし、DMAC を用いた SSIE 通信で Audio Codec デバイスへ出力します。

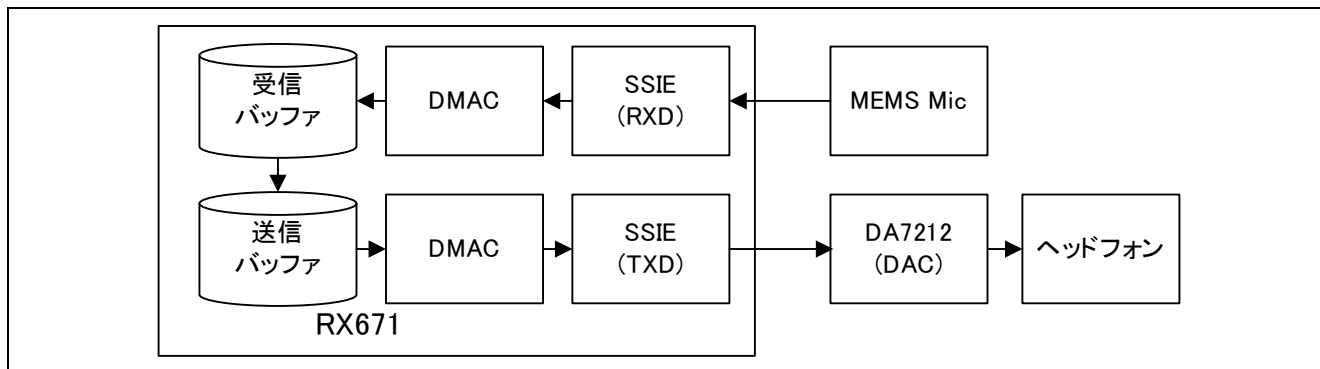


図 1.3 マイク音声入出力ブロック図

PCM データ転送フォーマット

- PCM データ転送フォーマット : I2S
- PCM データ : 8/16/18/20/22/24/32bit ステレオ
- サンプリングレート : RX671 マスタ 96/48/32/24/16/12/8kHz
RX671 スレーブ 96/48/44.1/32/24/22.05/16/12/11.025/8kHz

注 : EK ボード上の MEMS Mic(ICS-43434)の制約により、本システムでは PCM データ 24bit、サンプリングレート 48kHz が上限となります。

応用アプリケーション例 : ボイスレコーダー、音声認識システム

1.1.3 Audio Player モード

内蔵 Flash メモリに記録されている PCM または IMA ADPCM の WAV ファイルを読み出して、SSIE 通信で Audio Codec デバイスへ出力します。

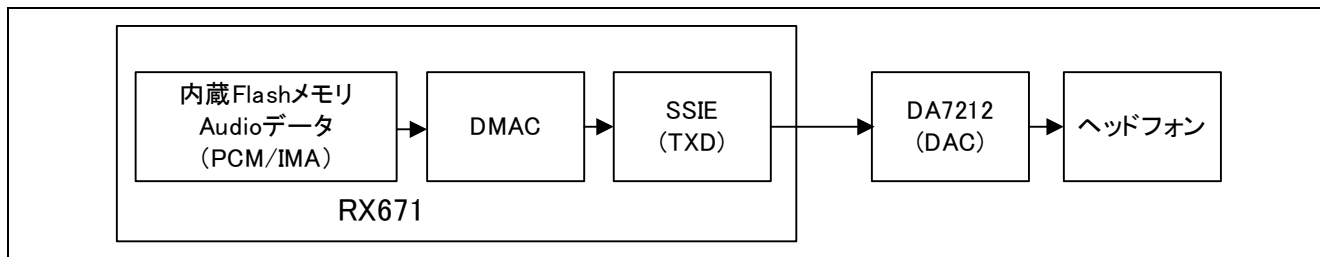


図 1.4 Audio Player ブロック図

対応ファイル

— PCM 8/16bit または IMA ADPCM *1 の WAV ファイル 最大サイズは 2,023,424byte
データは、内蔵 Flash メモリへ収納 アドレス 0xFFE10000 サイズ(最大) 0x1EE000

対応ファイル

- WAV ファイル PCM 8bit モノラル
- WAV ファイル PCM 8bit ステレオ
- WAV ファイル PCM 16bit モノラル
- WAV ファイル PCM 16bit ステレオ
- WAV ファイル IMA ADPCM モノラル *1
- WAV ファイル IMA ADPCM ステレオ *1

*1 : IMA ADPCM は動作サンプルであり、動作を保証するものではありません。

PCM データ転送フォーマット

— PCM データ転送フォーマット : I2S

— サンプリングレート : RX671 マスタ 96/48/32/24/16/12/8kHz

RX671 スレーブ 96/48/44.1/32/24/22.05/16/12/11.025/8kHz

応用アプリケーション例 : Audio Player、音声合成システム

2. 動作条件

2.1 動作モード設定

本プログラムは、起動時の SW 選択により、以下の 3 つのモードを選択出来ます。

表 2.1 動作モードの選択

項目	動作モード	起動時の操作	LED
1	正弦波出力モード	通常起動 (SW 操作なし)	LED1 が点灯
2	マイク音声入力-出力モード	SW1 を押しながら起動	LED2 が点灯
3	Audio Player モード	SW2 を押しながら起動	LED3 が点灯

2.2 設定パラメータ

表 2.2 設定パラメーター一覧

設定項目	設定項目ラベル	設定方法
RX671 SSIE マスタ/スレーブ	SSIE_CH0_CLK_MODE	SSI モジュールコード生成 ・ Ch0 Clock Supply Mode
サンプリングレート *1	SSIE_MCLK	SSI モジュールコード生成 ・ Master Clock
量子化ビット数	SSIE_CH0_DATA_WIDTH	SSI モジュールコード生成 ・ Ch0 PCM data width
正弦波周波数 *2	左チャンネル	WAVE_FREQ_HZ
	右チャンネル	WAVE_FREQ2_HZ
音量設定	ヘッドフォン	DAC_L_GAIN_OUTSIN_DB
		DAC_R_GAIN_OUTSIN_DB
		DAC_L_GAIN_THROUGH_DB
		DAC_R_GAIN_THROUGH_DB
		DAC_L_GAIN_AUDIO_PLAYER_DB
		DAC_R_GAIN_AUDIO_PLAYER_DB
		HP_L_GAIN_OUTSIN_DB
		HP_R_GAIN_OUTSIN_DB
		HP_L_GAIN_THROUGH_DB
		HP_R_GAIN_THROUGH_DB
		HP_L_GAIN_AUDIO_PLAYER_DB
		HP_R_GAIN_AUDIO_PLAYER_DB

*1 : RX671 がマスタの時に有効

*2 : 正弦波周波数設定は、正弦波出力モード時のみ有効

*3 : SSI モジュールコード生成 :

rx671_example_ssi_dma_rx_tx.scfg→コンポーネント→r_ssi_api_rx→プロパティ

2.2.1 RX671 SSIE マスタスレーブ

RX671 の SSIE 通信動作モードのマスタ/スレーブを切り替えます。

表 2.3 マスタスレーブ設定

SSIE_CH0_CLK_MODE	動作モード	状態
0	マスタ	RX671 は MCLK 出力 サンプリングレート設定(SSIE_MCLK)が有効
1 (初期値)	スレーブ	RX671 は MCLK 入力 サンプリングレート設定(SSIE_MCLK)が無効

2.2.2 サンプリングレート

RX671 がマスタの場合、サンプリングレートは SSIE_MCLK の値によって設定されます。

表 2.4 サンプリングレートの設定

SSIE_MCLK *1	サンプリングレート
256u	96kHz
512u (初期値)	48kHz
768u	32kHz
1024u	24kHz
1536u	16kHz
2048u	12kHz
3072u	8kHz

*1: RX671 がスレーブの場合、SSIE_MCLK の値は無視されます。

サンプリングレートの計算式

$$\text{サンプリングレート} = \text{AUDIO_CLK} / \text{SSIE_MCLK}$$

*本評価基板の場合: AUDIO_CLK=24576000 [Hz]

2.2.3 量子化ビット数

オーディオデータの量子化ビット数を設定します。

表 2.5 量子化ビット数

変数ラベル	設定値
SSIE_CH0_DATA_WIDTH	32bits / 24bits / 22bits / 20bits / 18bits / 16bits (初期値) / 8bits

2.2.4 正弦波周波数

正弦波出力モードでの正弦波周波数を設定します。

表 2.6 正弦波周波数

設定チャンネル	定義ラベル	設定値[Hz]
モノラル / L-ch	WAVEFREQ_HZ *1	1000(初期値)
R-ch	WAVEFREQ2_HZ *1	(500) *2

*1: 設定値は 25Hz の倍数を設定 (正弦波生成ソフトの仕様)

*2: モノラル時は WAVEFREQ2 を未定義にする

2.2.5 音量設定

L-R 別々に DAC、ヘッドフォンアンプのゲインを以下の定義にて設定できます。

表 2.7 音量設定

ゲイン設定	設定範囲 [dB]	説明
DAC_L_GAIN_OUTSIN_DB	-77.25 ~ 12 (0.75step)	正弦波出力モード時 DAC L-ch digital gain
DAC_R_GAIN_OUTSIN_DB	-77.25 ~ 12 (0.75step)	正弦波出力モード時 DAC R-ch digital gain
DAC_L_GAIN_THROUGH_DB	-77.25 ~ 12 (0.75step)	オンボードマイク入力出力モード時 DAC L-ch digital gain
DAC_R_GAIN_THROUGH_DB	-77.25 ~ 12 (0.75step)	オンボードマイク入力出力モード時 DAC R-ch digital gain
DAC_L_GAIN_AUDIO_PLAYER_DB	-77.25 ~ 12 (0.75step)	Audio Player モード時 DAC L-ch digital gain
DAC_R_GAIN_AUDIO_PLAYER_DB	-77.25 ~ 12 (0.75step)	Audio Player モード時 DAC R-ch digital gain
HP_L_GAIN_OUTSIN_DB	-57 ~ 6 (1step)	正弦波出力モード時 HP L-ch amplifier gain
HP_R_GAIN_OUTSIN_DB	-57 ~ 6 (1step)	正弦波出力モード時 HP R-ch amplifier gain
HP_L_GAIN_THROUGH_DB	-57 ~ 6 (1step)	オンボードマイク入力出力モード時 HP L-ch amplifier gain
HP_R_GAIN_THROUGH_DB	-57 ~ 6 (1step)	オンボードマイク入力出力モード時 HP R-ch amplifier gain
HP_L_GAIN_AUDIO_PLAYER_DB	-57 ~ 6 (1step)	Audio Player モード時 HP L-ch amplifier gain
HP_R_GAIN_AUDIO_PLAYER_DB	-57 ~ 6 (1step)	Audio Player モード時 HP R-ch amplifier gain

本アプリケーションでは各動作モードによって表 2.8 のように音量設定しています。

表 2.8 動作モード毎の音量設定

動作モード	DAC L/R digital gain	HP L/R amplifier gain
正弦波出力	0dB	-20dB
オンボードマイク入出力	0dB	+6dB
Audio Player	0dB	-20dB

3. プログラム説明

3.1 処理フロー

3.1.1 全体動作フロー

図 3.1 にメイン処理フロー、図 3.2 にオーディオ・アプリケーション処理フローを示します。

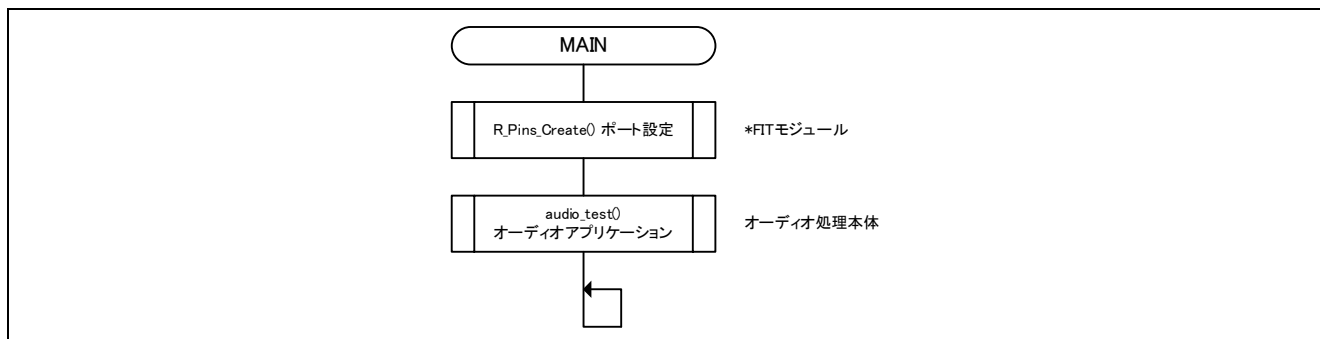


図 3.1 メイン処理フロー

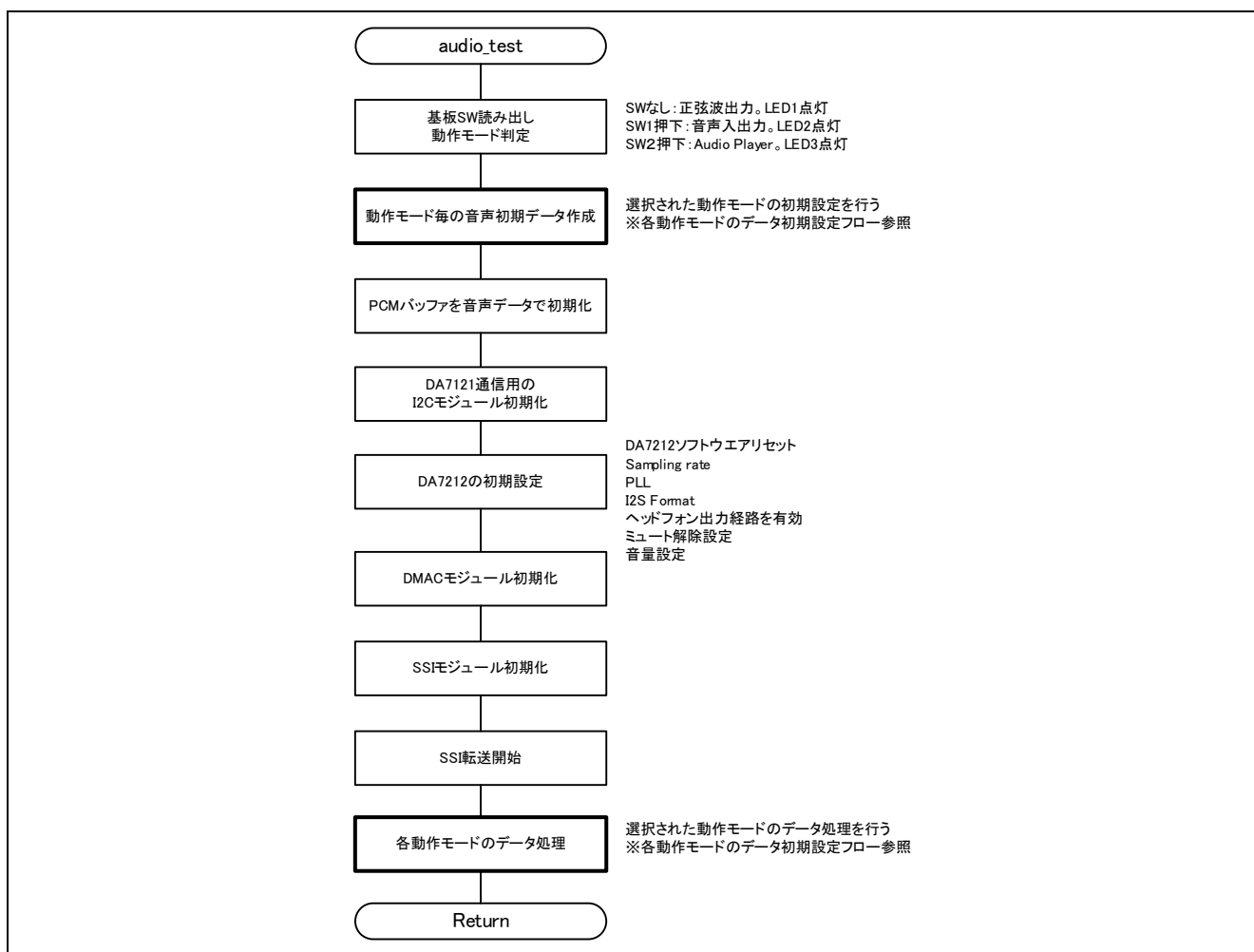


図 3.2 オーディオ・アプリケーション処理フロー

3.1.2 正弦波出力モード処理フロー

正弦波出力モードでは、パラメータ(WAVE_FREQ_HZ)で設定された周波数の正弦波の波形データをソフトウェア生成し、DMAC を用いて SSIE 通信で Audio Codec デバイスへ送信します。

マイクからの音声データは SSIE 通信で受信処理しますが、使用しません。

正弦波出力モードのデータ初期設定フローを図 3.3 に示します。

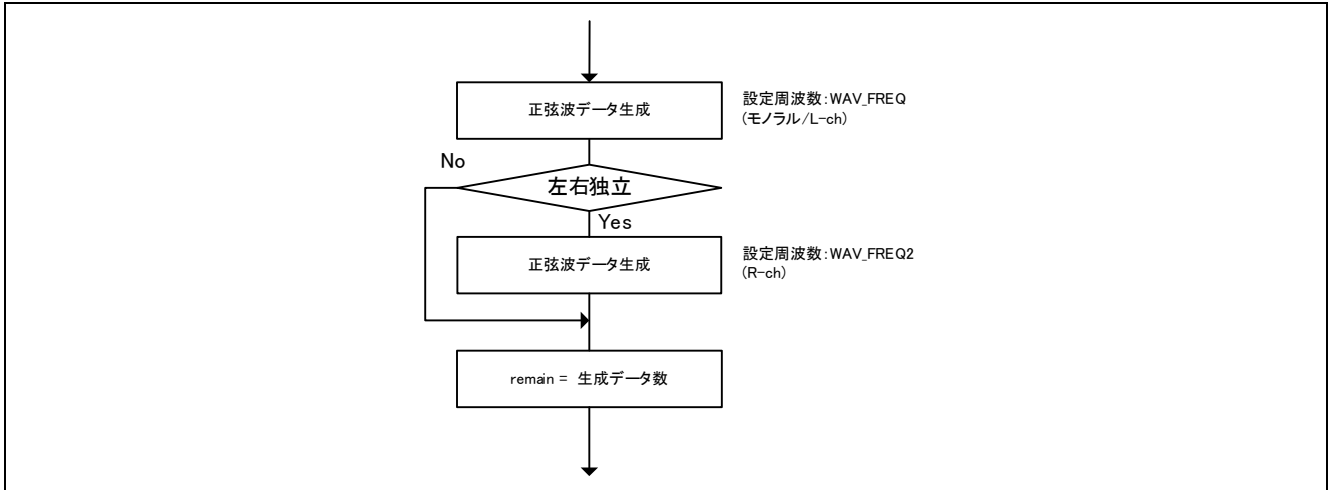


図 3.3 正弦波出力モードのデータ初期設定フロー

正弦波出力モードのデータ処理は DMAC によって行います。

3.1.3 音声入出力モード処理フロー

音声入出力モードでは、評価基板上のマイク(ICS-43434)からの音声データを SSIE 通信で受信し、DMAC で受信バッファへ転送。プログラムで送信バッファへコピーし、DMAC を用いて SSIE 通信で Audio Codec デバイスへ送信します。

Audio Player モードのデータ初期設定フローを図 3.4 に示します。

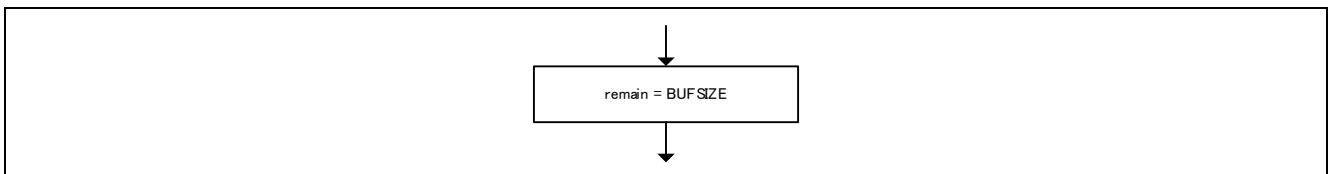


図 3.4 音声入出力モードのデータ初期設定フロー

注：音声入出力モードでは SSIE で受信した音声データを使用する為、データの初期設定処理は不要です。

音声入出力モードのデータ処理フローを図 3.5 に示します。

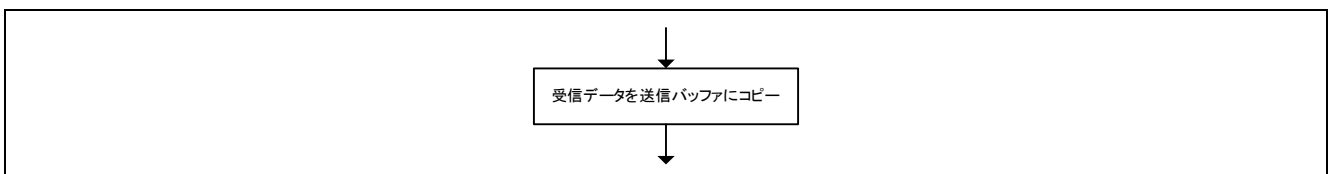


図 3.5 音声入出力モードのデータ処理フロー

SSIE 受信したマイクからの音声データを送信バッファにコピーします。送信バッファから SSIE へは DMAC で転送します。

3.1.4 Audio Player モード処理フロー

Audio Player モードでは、Audio データの記録された内蔵 Flash メモリからデータを読み出し、DMAC を用いて SSIE 通信で Audio Codec デバイスへ送信します。

マイクからの音声データは SSIE 通信で受信処理しますが、使用しません。

Audio Player モードのデータ初期設定フローを図 3.6 に示します。

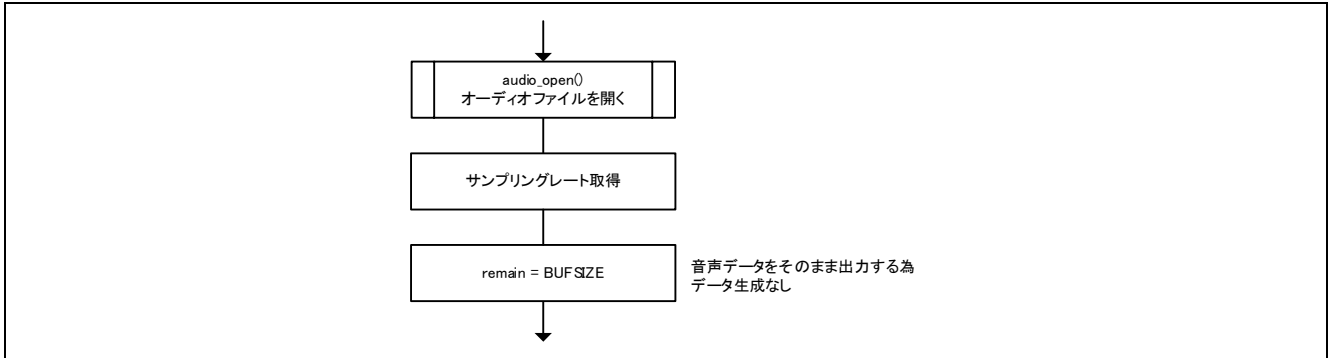


図 3.6 Audio Player モードのデータ初期設定フロー

Audio Player モードでは、記録されているオーディオデータを読み出してオーディオ情報を設定します。

Audio Player モードのデータ処理フローを図 3.7 に示します。

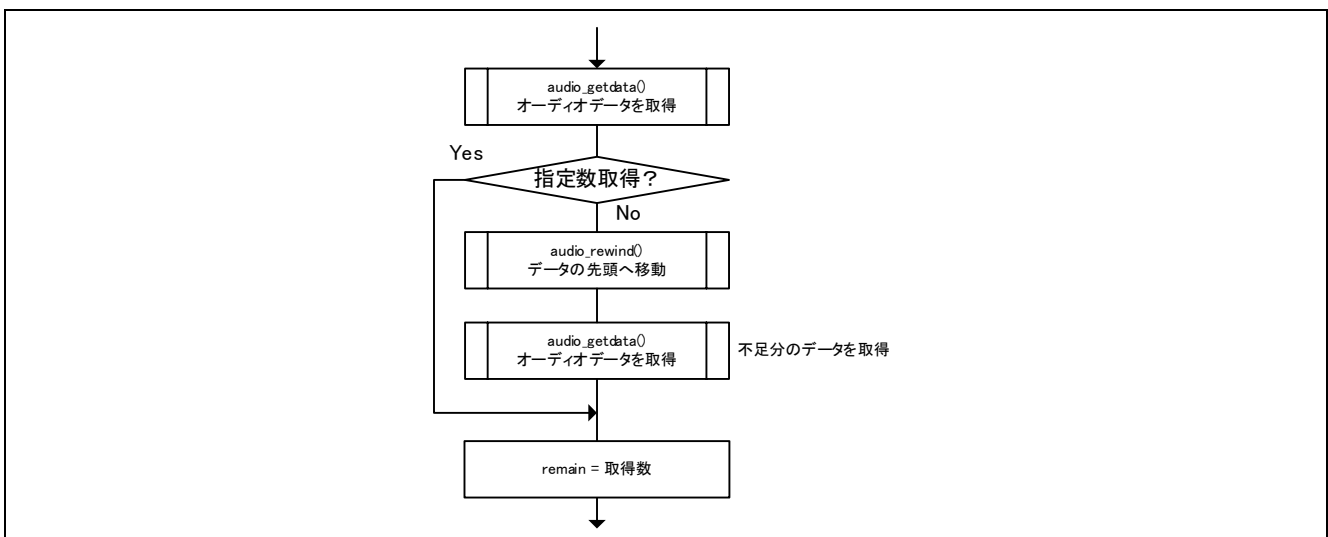


図 3.7 Audio Player モードのデータ処理フロー

3.2 ファイル構成

アプリケーションノートのプロジェクトファイル構成を図 3.8 に示します。

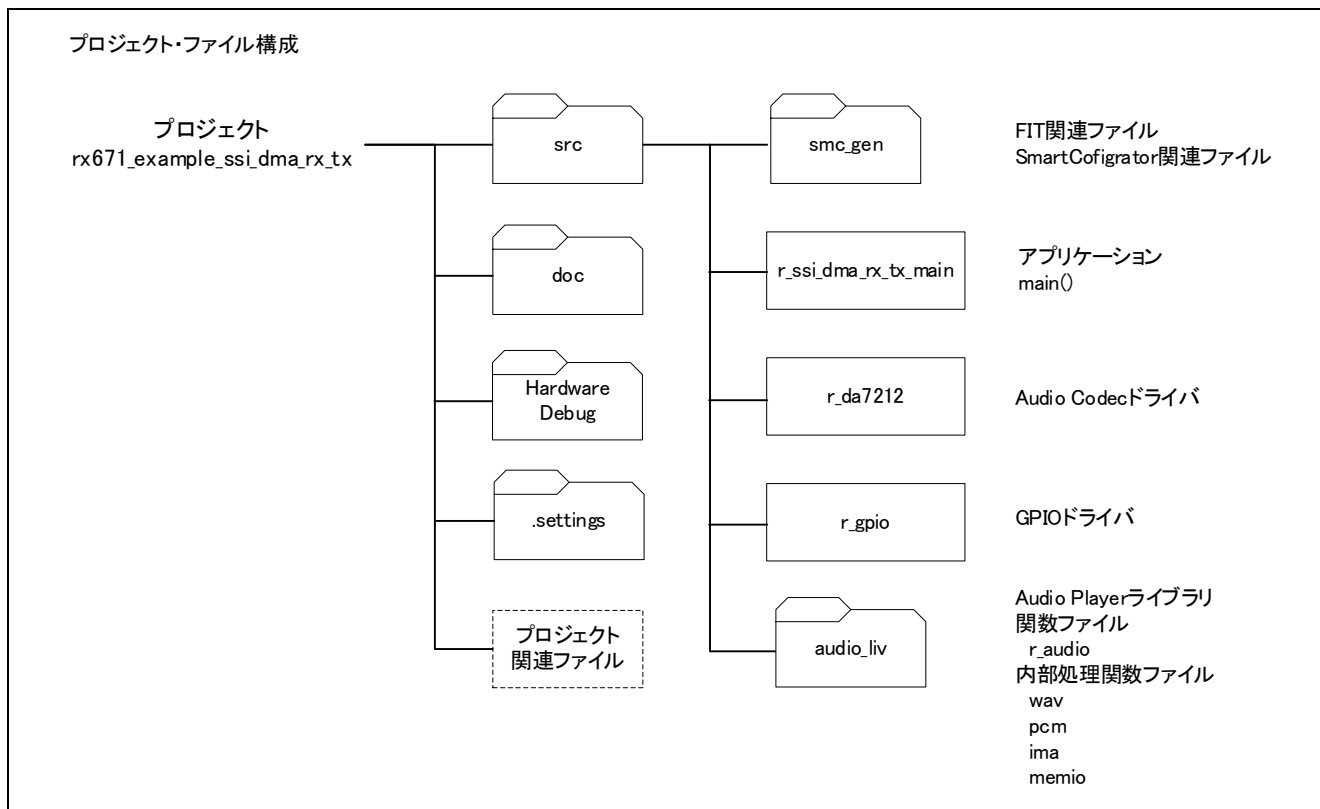


図 3.8 プロジェクトファイル構成

3.3 関数

3.3.1 関数構成

本プログラムで使用する関数構成を図 3.9 に示します。

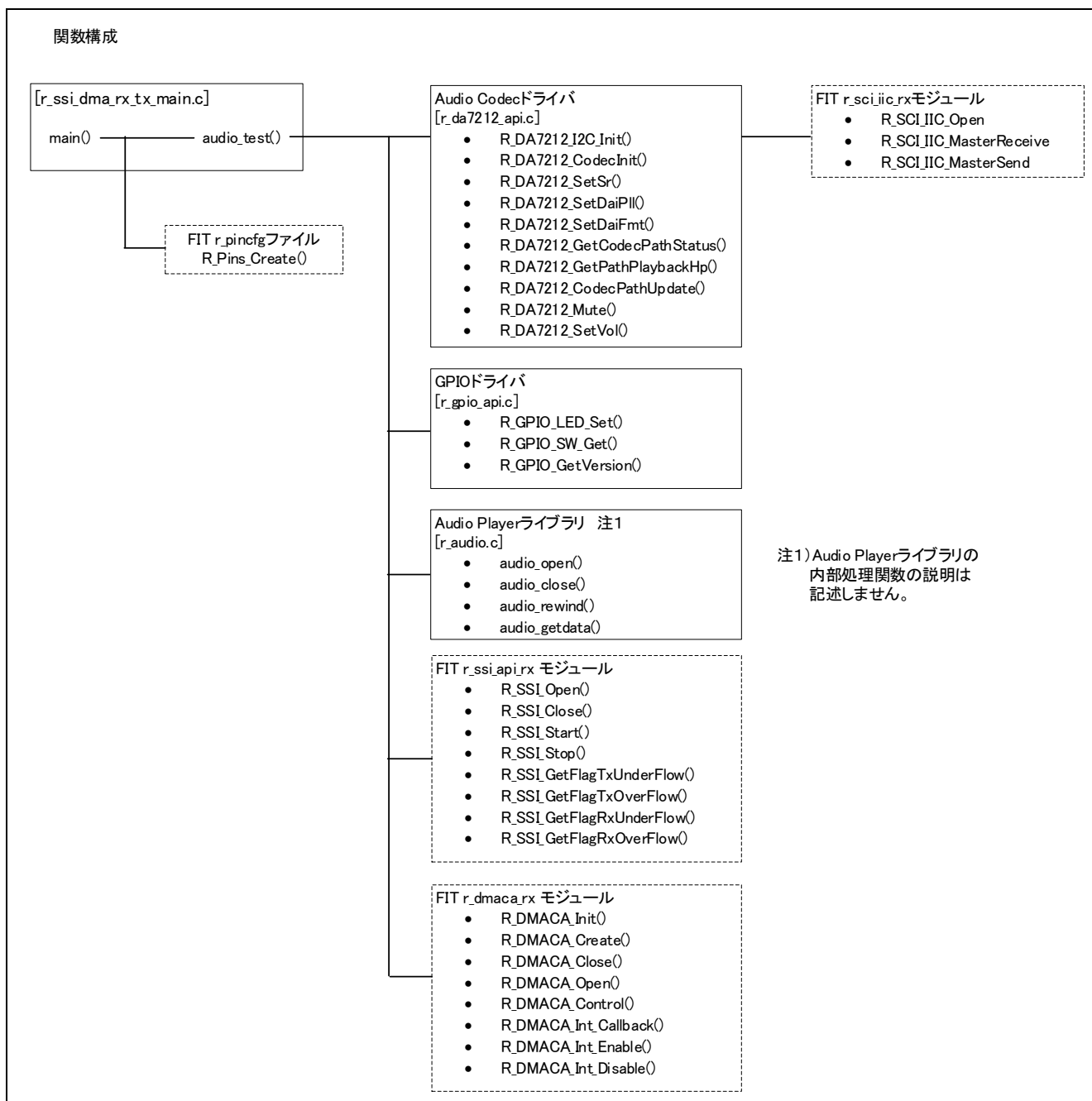


図 3.9 プログラム関数構成

3.3.2 アプリケーション関数

本アプリケーションで使用する、アプリケーション関数の一覧を表 3.1 に示します。

表 3.1 アプリケーション関数一覧

関数名	機能	ファイル
main	本アプリケーションのエントリ関数	r_ssi_dma_rx_tx_main.c
audio_test	オーディオ処理本体	

main

機能	エントリ関数
書式	void main(void)
引数	無し
戻り値	無し
処理	マイコンのピン定義 : FIT モジュール R_Pins_Create() オーディオ処理本体 audio_test()を起動

audio_test

機能	オーディオ処理本体
書式	int32_t audio_test(void)
引数	無し
戻り値	-1 : エラー発生 (正常動作時は内部動作を繰り返す為、関数は終了しない)
処理	処理内容は図 3.2 オーディオ・アプリケーション処理フロー参照

3.3.3 Audio Codec ドライバ関数

本アプリケーションで使用する Audio Codec ドライバ関数の一覧を表 3.2 に示します。

表 3.2 Audio Codec ドライバ関数一覧

関数名	機能	ファイル
R_DA7212_I2C_Init	DA7212 と通信するための I2C の初期化	r_da7212_api.c
R_DA7212_CodecInit	DA7212 の初期化	
R_DA7212_SetSr	Audio Codec のサンプリングレート設定	
R_DA7212_SetDaiPll	Audio Codec の PLL 設定	
R_DA7212_SetDaiFmt	Audio Codec の I2S フォーマット設定	
R_DA7212_GetCodecPathStatus	Audio Codec の現在のコントロールレジスタ設定値を取得	
R_DA7212_GetPathPlaybackHp	Audio Codec のヘッドフォン出力のコントロールレジスタ設定値を取得	
R_DA7212_Mute	Audio Codec の機能のミュート制御	
R_DA7212_SetVol	Audio Codec のボリューム設定	
R_DA7212_CodecPathUpdate	Audio Codec のレジスタ設定関数	

R_DA7212_I2C_Init

機能	DA7212 と通信するための I2C 初期化
書式	int32_t R_DA7212_I2C_Init(void)
引数	無し
戻り値	0:success, -1: error
処理	FIT モジュール R_SCI_IIC_Open()実行し成否を返す

R_DA7212_CodecInit

機能	DA7212 の初期化
書式	int32_t audio_test(void)
引数	無し
戻り値	0:success, -1: error
処理	DA7212 をソフトウェアリセットし、コントロールレジスタ設定を全て OFF に設定

R_DA7212_SetSr

機能	Audio Codec のサンプリングレート設定
書式	int32_t R_DA7212_SetSr(int32_t sampling_rate)
引数	int32_t sampling_rate sampling rate
戻り値	0:success, -1: error
処理	指定された sampling rate に対応する値をレジスタに設定

R_DA7212_SetDaiPll

機能	Audio Codec の PLL 設定
書式	int32_t R_DA7212_SetDaiPll(int32_t sampling_rate, uint32_t sys_mclk)
引数	int32_t sampling_rate サンプルレイト uint32_t sys_mclk SYS_MCLK
戻り値	0:success, -1: error
処理	sampling_rate と sys_mclk から、適切な PLL 設定を算出しレジスタに設定

R_DA7212_SetDaiFmt

機能	Audio Codec の I2S フォーマット設定
書式	int32_t R_DA7212_SetDaiFmt(int32_t data_width)
引数	int32_t data_width Word Length
戻り値	0:success, -1: error
処理	data_width、DA7212_CODEEC_SLAVE をもとに通信フォーマットを算出しレジスタに設定

R_DA7212_GetCodecPathStatus

機能	Audio Codec の現在のコントロールレジスタ設定を取得
書式	void R_DA7212_GetCodecPathStatus(uint8_t * amp_tbl)
引数	uint8_t * amp_tbl コントロールテーブルポインタ
戻り値	無し
処理	コントロールレジスタ設定を取得

R_DA7212_GetPathPlaybackHp

機能	ヘッドフォン出力設定、コントロールレジスタ値を取得
書式	void R_DA7212_GetPathPlaybackHp(uint8_t * amp_tbl)
引数	uint8_t * amp_tbl コントロールテーブルポインタ
戻り値	無し
処理	ヘッドフォン出力に設定し、コントロールレジスタ設定を取得

R_DA7212_Mute

機能	Audio Codec のミュート制御
書式	int32_t R_DA7212_Mute(e_hal_controls_t controls, int32_t mute)
引数	e_hal_controls_t controls control target int32_t mute 0: un-mute, 1: mute
戻り値	0:success, -1: error
処理	指定したターゲットのミュート制御する

R_DA7212_SetVol

機能	Audio Codec のボリューム設定
書式	int32_t R_DA7212_SetVol(e_hal_controls_t amp, double vol)
引数	e_hal_controls_t controls control target double vol volume
戻り値	0:success, -1: error
処理	指定したモジュールのボリュームを制御する

R_DA7212_CodecPathUpdate

機能	Audio Codec のレジスタ設定関数	
書式	int32_t R_DA7212_CodecPathUpdate(uint8_t * amp_tbl)	
引数	uint8_t * amp_tbl	コントロールテーブルポインタ
戻り値	0:success, -1:error	
処理	コントロールテーブル値をレジスタに設定	

3.3.4 GPIO ドライバ関数

本アプリケーションで使用する GPIO ドライバ関数の一覧を表 3.3 に示します。

表 3.3 GPIO ドライバ関数一覧

関数名	機能	ファイル
R_GPIO_LED_Set	LED の点灯制御	r_gpio_api.c
R_GPIO_SW_Get	評価基板のスイッチのポート状態を取得	
R_GPIO_GetVersion	ドライバのバージョンを取得	

R_GPIO_LED_Set

機能	LED の点灯制御	
書式	int32_t R_GPIO_LED_Set(int32_t no, int32_t offon)	
引数	int32_t no	LED Number 1:LED1, 2:LED2, 3:LED3
	int32_t offon	0:LED OFF, 1:LED ON
戻り値	0:success, -1: error	
処理	指定のポートを H/L 制御し LED を ON/OFF 制御する	

R_GPIO_SW_Get

機能	評価基板のスイッチのポート状態を取得	
書式	int32_t R_GPIO_SW_Get(int32_t no)	
引数	int32_t no	SW Number 1:SW1, 2:SW2
戻り値	0/1: pin status, -1: error	
処理	指定の SW のポートの状態を取得する	

R_GPIO_GetVersion

機能	GPIO ドライバのバージョンを取得	
書式	uint32_t R_GPIO_GetVersion(void)	
引数	なし	
戻り値	GPIO ドライバのバージョン (最上位の 2 バイトがメジャーバージョン番号、最下位の 2 バイトがマイナーバージョン番号)	
処理	GPIO ドライバのバージョンを取得する	

3.3.5 Audio Player ライブラリ関数

本アプリケーションの Audio Player モード実行時に使用する専用関数一覧を表 3.4 に示します。

表 3.4 Audio Player モード実行時に使用する関数一覧

関数名	機能	ファイル
audio_open	オーディオファイルを開く	r_audio.c
audio_close	オーディオファイルを閉じる	
audio_rewind	オーディオファイルの再生位置を先頭に戻す	
audio_getdata	オーディオファイルからデータを読み込む	

audio_open

機能	オーディオファイルを開く	
書式	int32_t audio_open(const void *data, uint32_t size, int32_t ch, int32_t bits, audio_t *audio)	
引数	const void *data	オーディオファイルの格納アドレス
	uint32_t size	オーディオ・ファイルサイズ
	int32_t ch	出力データのチャンネル数(1:モノラル 2:ステレオ)
	int32_t bits	出力データの量子化ビット数
	audio_t *audio	Audio ハンドルを格納するポインタ
戻り値	0	正常
	-2	ファイルが開けなかった(AUDIO_E_FILE)
	-4	サポートしていないデータ形式(AUDIO_E_NOSPT)
処理	オーディオファイルを開き、audio 構造体にAudio ハンドルを格納する audio_rewind(), audio_getdata(), audio_close()はこのハンドルを指定して使用する	

audio_close

機能	オーディオファイルを閉じる	
書式	int32_t audio_close(audio_t *audio)	
引数	audio_t *audio	Audio ハンドル
戻り値	0	正常
	-5	不明なエラー(AUDIO_E_UNKNOWN)
処理	指定したオーディオファイルを閉じる	

audio_rewind

機能	オーディオファイルの再生位置を先頭に戻す	
書式	int32_t audio_rewind(audio_t *audio)	
引数	audio_t *audio	Audio ハンドル
戻り値	0	正常
	-5	不明なエラー(AUDIO_E_UNKNOWN)
処理	オーディオファイルの再生位置を先頭に戻す	

audio_getdata		
機能	オーディオファイルからデータを読み込む	
書式	int32_t audio_getdata(audio_t *audio, void *p, uint32_t samples)	
引数	audio_t *audio	Audio ハンドル
	void *p	読み出した音声データの格納位置(ポインタ)
	long samples	読み出すデータの最大サンプル数
戻り値	0 以上の値	読み出したデータのサンプル数(0 の場合はもうデータがない)
	-2	ファイル・オープン失敗(AUDIO_E_FILE)
	-4	サポートしていないデータ形式(AUDIO_E_NOSPT)
	-5	不明なエラー(AUDIO_E_UNKNOWN)
処理	オーディオデータを指定された読み出し位置から指定されたデータ数読み出す	

4. FIT モジュール

本プロジェクトで使用する API 関数の一覧を表 4.1 に示します。

表 4.1 使用する API 関数一覧

FIT モジュールファイル名	FIT バージョン	内容
r_bsp	7.42	ボードサポートパッケージモジュール (AN1685)
r_dmaca_rx	3.20	DMAC モジュール(AN2063)
r_ssi_api_rx	2.03	SSI API モジュール (AN2150)
r_sci_iic_rx	2.71	SCI 簡易 I2C モジュール (AN1691)

各 API 関数及び機能説明は、各 FIT のアプリケーションノートを参照してください。

5. プロジェクトをインポートする方法

サンプルコードは 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。

5.1 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。)

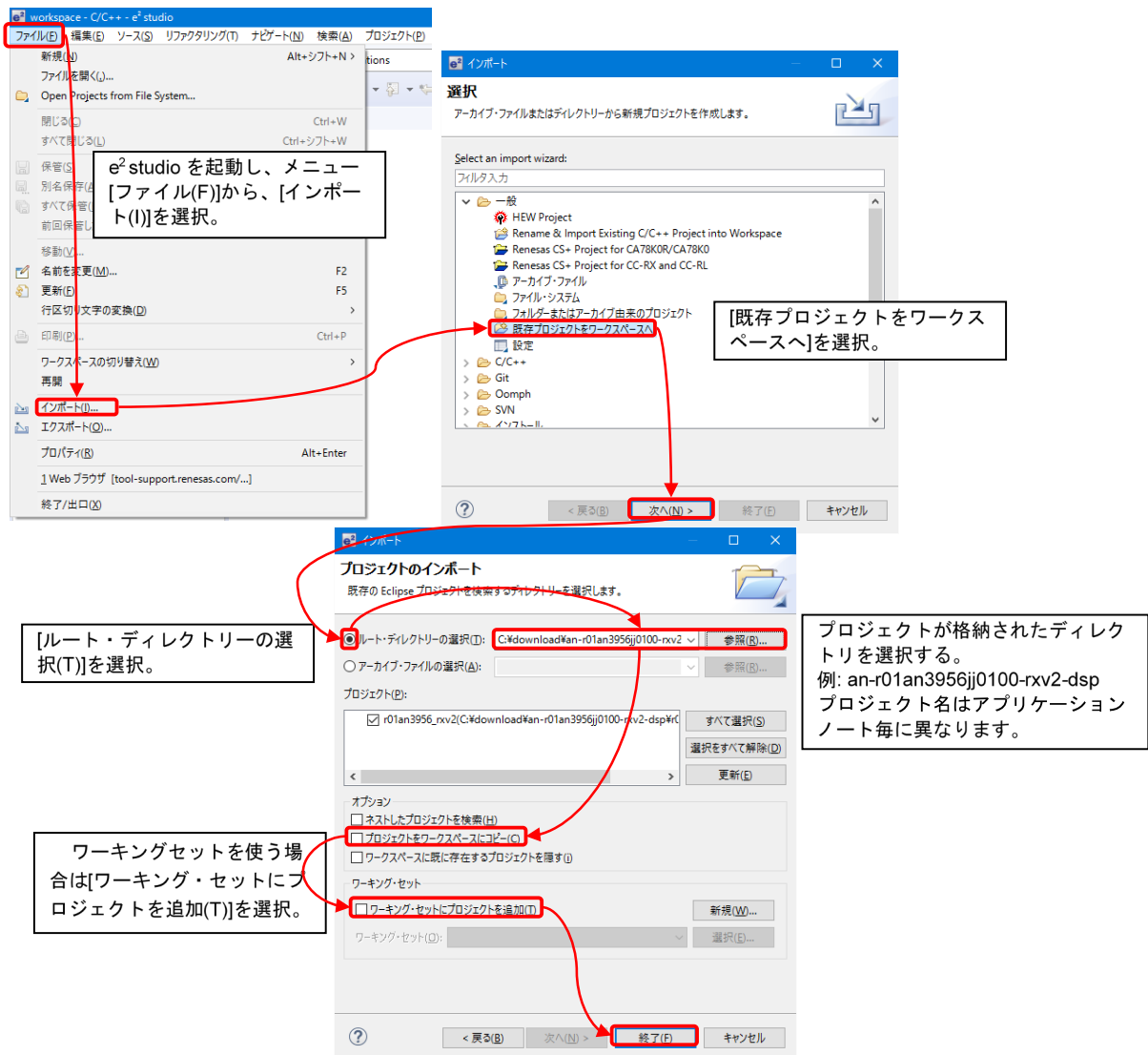


図 5.1 プロジェクトを e² studio にインポートする方法

5.2 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。)

CS+を起動し、スタート画面から、
[e² studio/CubeSuite/High-performance Embedded Workshop/PM+のプロジェクトを開く]を選択。

プロジェクトを選択する。
例: r01an3956_rvx2
プロジェクト名はアプリケーションノート毎に異なります。

拡張子[.rcpc]のファイルを選択して[開く]ボタンを押す。

[e² studio用プロジェクト・ファイル(*.rcpc)]を選択。

使用するマイクロコントローラを選択してください。

プロジェクト種類: 「空のアプリケーション(CC-RX)」を選択し、プロジェクト名と作成場所を指定してください。

図 5.2 プロジェクトをCS+にインポートする方法

6. サンプルコード

サンプルコードは、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。

7. 参考ドキュメント

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

Firmware Integration Technology ユーザーズマニュアル(R01AN1833)

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 簡易 I2C モジュール Firmware Integration Technology (R01AN1691)

RX ファミリ DMAC モジュール Firmware Integration Technology(R01AN2063)

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(R01AN2825)

(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)

マニュアル、データシート

RX671 グループ ユーザーズマニュアル ハードウェア編 (R01UH0899)

EK-RX671 v1 ユーザーズマニュアル(R20UT5234)

(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	2023.01.10	-	初版発行
1.01	2024.02.13	14	図 3-9 を修正。
		17	R_DA7212_SetVol 関数の書式説明を修正。
		18	R_GPIO_GetVersion 関数を追加。
		プログラム	SW2 に対応する I/O ポートの define 定義を修正。 Audio Codec ドライバ及び GPIO ドライバのファイル名を修正。

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しないでください。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えば、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。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た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。