

要旨

本アプリケーションノートでは、RZ/T1 の 12 ビット A/D コンバータ (S12ADCa Unit0) 以後、ADC 機能を使用し A/D 変換するサンプルプログラムについて説明します。

ADC サンプルプログラムの特長を以下に示します。

- ポテンションメータの入力電圧を A/D 変換します。
- 変換結果を 4 段階に識別し LED0、LED1、LED2、LED3 に表示します。

対象デバイス

RZ/T1

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.	仕様	4
2.	動作環境	5
3.	関連アプリケーションノート	6
4.	周辺機能説明	7
5.	ハードウェア説明	8
5.1	ハードウェア構成例	8
5.2	使用端子一覧	8
6.	ソフトウェア説明	9
6.1	動作概要	9
6.1.1	プロジェクト設定	9
6.1.2	使用準備	9
6.2	メモリマップ	10
6.2.1	サンプルプログラムのセクション配置	10
6.2.2	MPU の設定	10
6.2.3	例外処理ベクタテーブル	10
6.3	使用割り込み一覧	10
6.4	固定幅整数一覧	10
6.5	定数／エラーコード一覧	11
6.6	構造体／共用体／列挙型一覧	12
6.7	大域変数一覧	16
6.8	関数一覧	17
6.9	関数仕様	18
6.9.1	main	18
6.9.2	adc_sample_led_init	18
6.9.3	adc_sample_led_off	18
6.9.4	adc_sample_adtrg_init	19
6.9.5	adc_sample_callback	19
6.9.6	R_ADC_Open	20
6.9.7	R_ADC_Control	21
6.9.8	R_ADC_Read	21
6.9.9	R_ADC_ReadAll	22
6.9.10	R_ADC_Close	22
6.9.11	R_ADC_GetVersion	23
6.9.12	adc_s12adi0_isr	23
6.9.13	adc_gbadi_isr	23
6.10	フローチャート	24
6.10.1	メイン処理	24
6.10.2	adc_sample_led_init	26
6.10.3	adc_sample_adtrg_init	26
6.10.4	adc_sample_callback	26

6.10.5	adc_s12adi0_isr	27
6.11	R_ADC_Control コマンド一覧	28
6.11.1	ADC_CMD_ENABLE_CHANS	29
6.11.2	ADC_CMD_ENABLE_TEMP_SENSOR	29
6.11.3	ADC_CMD_SET_SAMPLE_STATE_CNT	29
6.11.4	ADC_CMD_ENABLE_TRIG	30
6.11.5	ADC_CMD_DISABLE_TRIG	30
6.11.6	ADC_CMD_SCAN_NOW	30
6.11.7	ADC_CMD_ENABLE_INT	31
6.11.8	ADC_CMD_DISABLE_INT	31
6.11.9	ADC_CMD_ENABLE_INT_GROUPB	31
6.11.10	ADC_CMD_DISABLE_INT_GROUPB	32
6.11.11	ADC_CMD_CHECK_SCAN_DONE	32
6.11.12	ADC_CMD_CHECK_SCAN_DONE_GROUPA	32
6.11.13	ADC_CMD_CHECK_SCAN_DONE_GROUPB	33
7.	サンプルコード	34
8.	参考ドキュメント	35

1. 仕様

表 1.1 に使用する周辺機能と用途を、図 1.1 にサンプルコード実行時の動作環境を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
低消費電力低減機能	ADCモジュールへのクロック供給/停止
割り込みコントローラ (ICUA) (割り込みID : 35)	A/D変換終了割り込み要求を受け付け、Cortex-R4 への割り込み
マルチファンクションピンコントローラ (MPC) (割り当てポート : P44)	A/D変換開始のための外部トリガ端子ADTRG0#に設定
ADC (AN007)	ポテンションメータの入力電圧のA/D変換
ポテンションメータ	ADCへ可変した電圧を入力

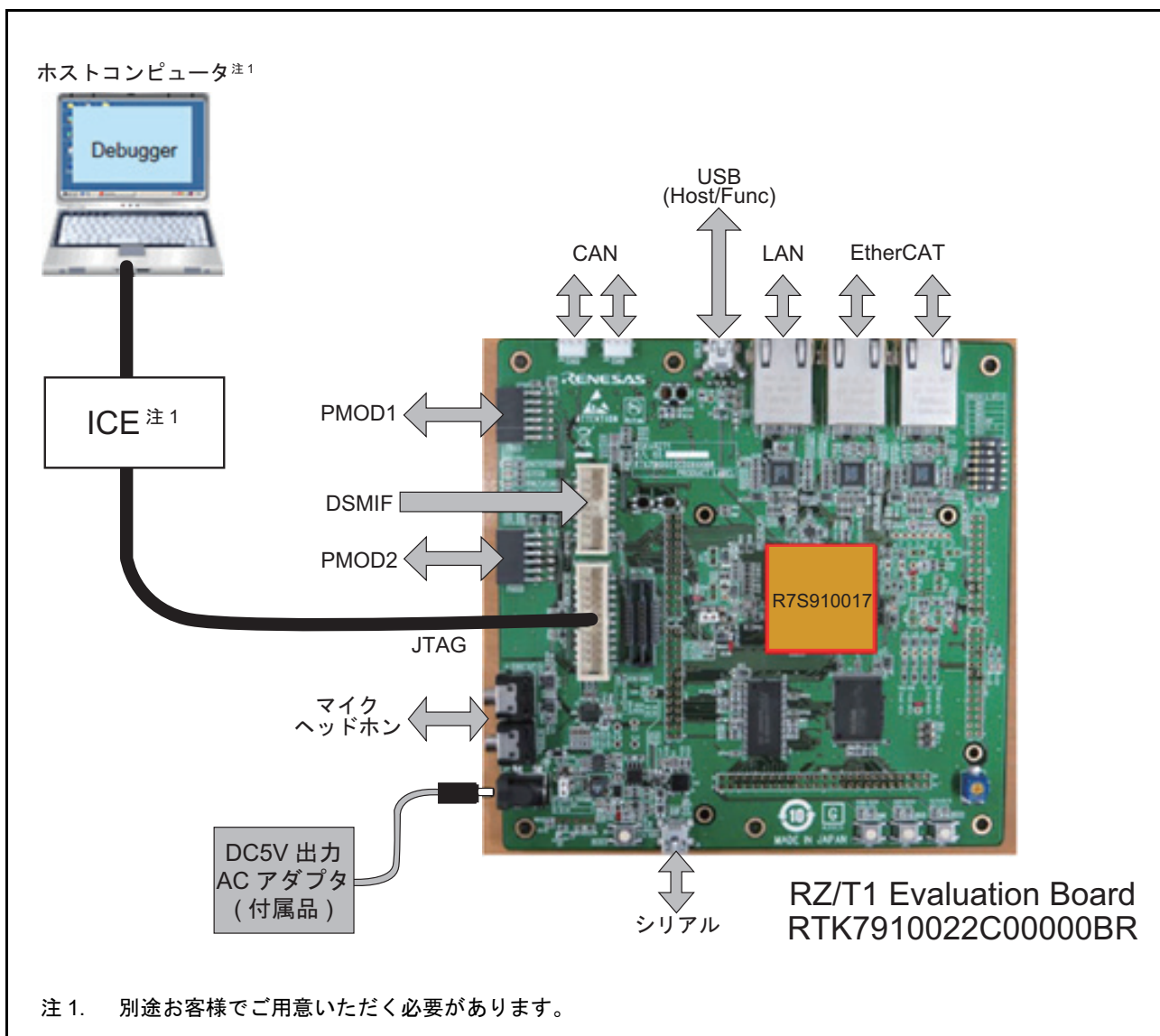


図 1.1 動作環境

2. 動作環境

本アプリケーションノートのサンプルコードは、下記の環境を想定しています。

表2.1 動作環境

項目	内容
使用マイコン	RZ/T1グループ
動作周波数	CPUCLK = 450MHz
動作電圧	3.3V
統合開発環境	IARシステムズ製 Embedded Workbench® for Arm Version 8.20.2 Arm製 DS-5™ 5.26.2 RENESAS製 e2studio 6.1.0
動作モード	SPIブートモード 16ビットバスブートモード
使用ボード	RZ/T1 Evaluation Board (RTK7910022C00000BR)
使用デバイス (ボード上で使用する機能)	<ul style="list-style-type: none">• NORフラッシュメモリ (CS0、CS1空間に接続) メーカー名 : Macronix International Co., 型名 : MX29GL512FLT2I-10Q• SDRAM (CS2、CS3空間に接続) メーカー名 : Integrated Silicon Solution Inc、型名 : IS42S16320D-7TL• シリアルフラッシュメモリ メーカー名 : Macronix International Co., 型名 : MX25L51245G• ポテンショメータ (AN007)

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- RZ/T1グループ 初期設定 アプリケーションノート (R01AN2554JJ)

注. 本アプリケーションノートで記載しないレジスタに関しては、RZ/T1グループ 初期設定 アプリケーションノートで設定した値のまま使用します。

4. 周辺機能説明

動作モード、消費電力低減機能、割り込みコントローラ (ICUA)、汎用入出力ポート、12ビット A/D コンバータ (S12ADCa) についての基本的な内容は、RZ/T1 グループ・ユーザーズマニュアルハードウェア編を参照してください。

5. ハードウェア説明

5.1 ハードウェア構成例

図 5.1 にハードウェア構成例を示します。

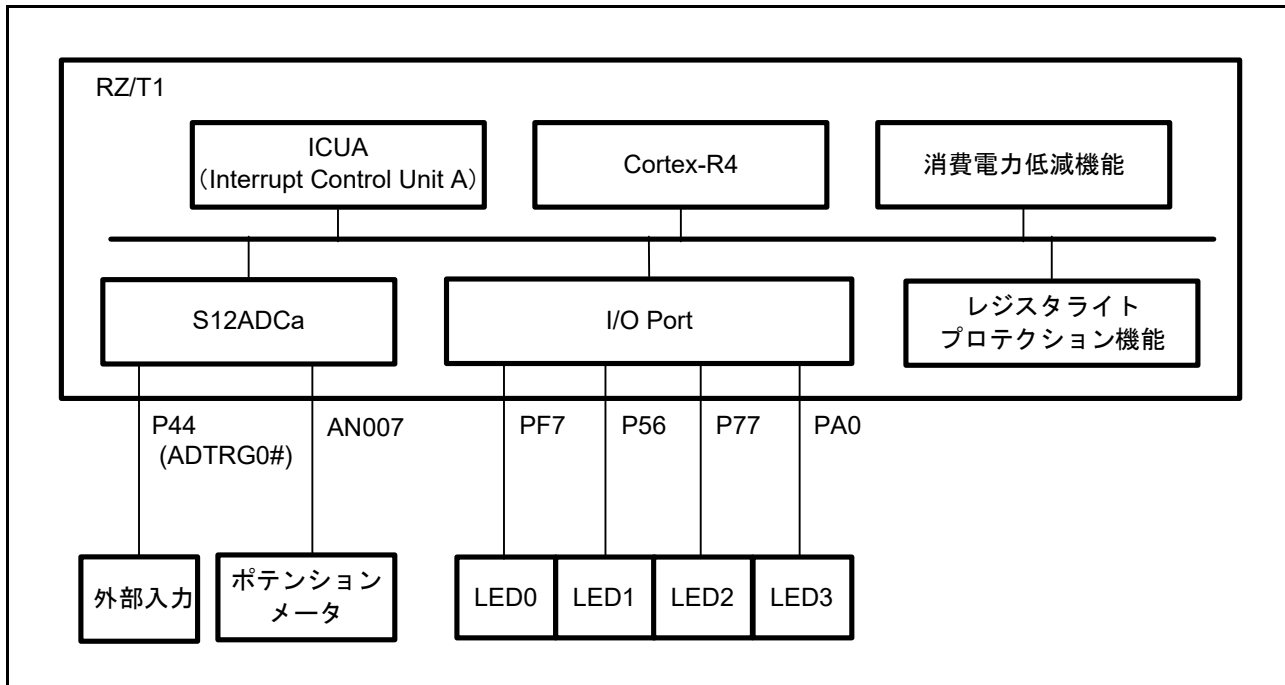


図 5.1 ハードウェア構成例

5.2 使用端子一覧

表 5.1 に使用端子と機能を示します。

表 5.1 使用端子と機能

端子名	入出力	内容
AN007	入力	ポテンションメータ
P44/ADTRG0#	入力	A/D変換開始のための外部トリガ端子

6. ソフトウェア説明

6.1 動作概要

ADC サンプルプログラムの機能概要を表 6.1 動作概要に示します。また、図 6.1 にシステムブロック図を示します。

表 6.1 動作概要

機能	概要
入力チャンネル	ポテンションメータが接続されている AN007 に設定
動作モード	シングルスキャンモードに設定し AN007 のみ変換
A/D 変換開始方法	ビルド時に下記の A/D 変換方法を選択 <ul style="list-style-type: none"> ソフトウェア起動 外部トリガ起動
ADC 変換結果の取得	A/D 変換結果を 4 段階に識別子、それぞれの段階に割り当てた評価ボード上の LED ^{注1} に表示 注1. LED0, LED1, LED2, LED3

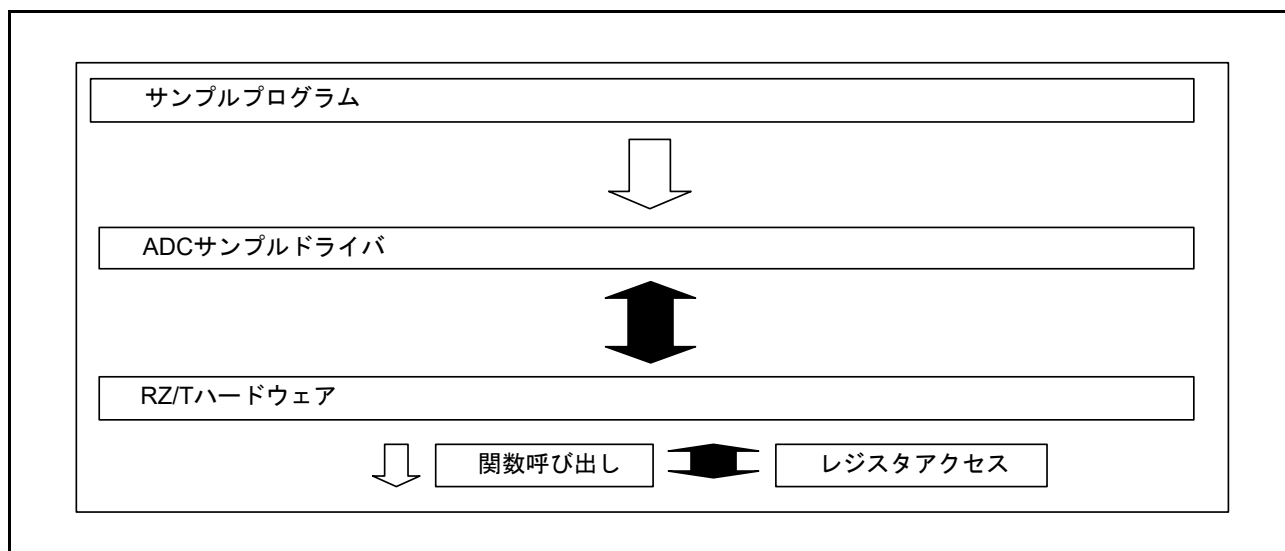


図 6.1 システムブロック図

6.1.1 プロジェクト設定

開発環境となる EWARM 上で使用されるプロジェクト設定については、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.1.2 使用準備

本サンプルプログラムの実行準備は必要ありません。

6.2 メモリマップ

RZ/T1 グループのアドレス空間と RZ/T1 評価ボードのメモリマッピングについては、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.2.1 サンプルプログラムのセクション配置

サンプルプログラムで使用するセクションおよびサンプルプログラムの初期状態のセクション配置（ロードビュー）、スキップロード機能を使用後のセクション配置（実行ビュー）は、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.2.2 MPU の設定

MPU の設定は、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.2.3 例外処理ベクタテーブル

例外処理ベクタテーブルについては、RZ/T1 グループ 初期設定 アプリケーションノートに記載しています。

6.3 使用割り込み一覧

表 6.2 にサンプルコードで使用する割り込みを示します。

表6.2 サンプルコードで使用する割り込み

割り込み（要因ID）	優先度	処理概要
A/D変換終了割り込み	7	コールバック関数をコール

6.4 固定幅整数一覧

表 6.3 にサンプルコードで使用する固定幅整数を示します。

表6.3 サンプルコードで使用する固定幅整数

シンボル	内容
int8_t	8ビット整数、符号あり（標準ライブラリにて定義）
int16_t	16ビット整数、符号あり（標準ライブラリにて定義）
int32_t	32ビット整数、符号あり（標準ライブラリにて定義）
int64_t	64ビット整数、符号あり（標準ライブラリにて定義）
uint8_t	8ビット整数、符号なし（標準ライブラリにて定義）
uint16_t	16ビット整数、符号なし（標準ライブラリにて定義）
uint32_t	32ビット整数、符号なし（標準ライブラリにて定義）
uint64_t	64ビット整数、符号なし（標準ライブラリにて定義）

6.5 定数／エラーコード一覧

表 6.4 にサンプルコードで使用する定数を示します。

表6.4 サンプルコードで使用する定数

定数名	設定値	内容
ADC_PORT_PDR_OUT	3u	I/Oポート出力
ADC_PORT_PMR_IO_SET	0u	I/Oポートを汎用入出力ポート
ADC_LED_OFF	0u	LED0、LED1、LED2、LED3を消灯
ADC_LED_ON	1u	LED0、LED1、LED2、LED3を点灯
ADC_LED_COUNT	9999u	LED点灯デューティ比生成カウンタ定数
ADC_MPC_ADTRG0	0x09u	兼用機能をADTRG#に割り当て
ADC_ADI_PRI	7u	スキャン割り込みの優先度
ADC_LEVEL0	820u	LED0が点灯するA/D変換値の閾値
ADC_LEVEL1	1639u	LED1が点灯するA/D変換値の閾値
ADC_LEVEL2	2458u	LED2が点灯するA/D変換値の閾値
ADC_LEVEL3	3277u	LED3が点灯するA/D変換値の閾値
ADC_SAMPLE_TRIG	0	A/D変換の起動方法を切り替えるコンパイルスイッチ 0：ソフトウェアトリガによる起動 1：外部トリガによる起動

6.6 構造体／共用体／列挙型一覧

図 6.2 にサンプルコードで使用する構造体／共用体／列挙型を示します。

```
typedef enum e_adc_mode
{
    ADC_MODE_SS_TEMPERATURE,      // single scan temperature sensor
    ADC_MODE_SS_ONE_CH,          // single scan one channel
    ADC_MODE_SS_ONE_CH_DBLTRIG,  // on even triggers save to ADDBLDR & interrupt
    ADC_MODE_SS_MULTI_CH,        // 1 trigger source, scan multiple channels
    ADC_MODE_SS_MULTI_CH_GROUPED, // 2 trigger sources, scan multiple channels
    ADC_MODE_SS_MULTI_CH_GROUPED_DBLTRIG_A,
    ADC_MODE_CONT_ONE_CH,        // continuous scan one channel
    ADC_MODE_CONT_MULTI_CH,      // continuous scan multiple channels
    ADC_MODE_MAX
} adc_mode_t;

typedef enum e_adc_trig          // trigger sources
{
    ADC_TRIG_ADTRG0 = 0,
    ADC_TRIG_TRGA0N = 1,
    ADC_TRIG_TRGA1N = 2,
    ADC_TRIG_TRGA2N = 3,
    ADC_TRIG_TRGA3N = 4,
    ADC_TRIG_TRGA4N = 5,
    ADC_TRIG_TRGA6N = 6,
    ADC_TRIG_TRGA7N = 7,
    ADC_TRIG_TRG0N = 8,
    ADC_TRIG_TRG4AN = 9,
    ADC_TRIG_TRG4BN = 10,
    ADC_TRIG_TRG4AN_OR_TRG4BN = 11,
    ADC_TRIG_TRG4ABN = 12,
    ADC_TRIG_TRG7AN = 13,
    ADC_TRIG_TRG7BN = 14,
    ADC_TRIG_TRG7AN_OR_TRG7BN = 15,
    ADC_TRIG_TRG7ABN = 16,
    ADC_TRIG_GTADTRA0N = 17,
    ADC_TRIG_GTADTRB0N = 18,
    ADC_TRIG_GTADTRA1N = 19,
    ADC_TRIG_GTADTRB1N = 20,
    ADC_TRIG_GTADTRA2N = 21,
    ADC_TRIG_GTADTRB2N = 22,
    ADC_TRIG_GTADTRA3N = 23,
    ADC_TRIG_GTADTRB3N = 24,
    ADC_TRIG_GTADTRA0N_OR_GTADTRB0N = 25,
    ADC_TRIG_GTADTRA1N_OR_GTADTRB1N = 26,
```

```
ADC_TRIG_GTADTRA2N_OR_GTADTRB2N = 27,
ADC_TRIG_GTADTRA3N_OR_GTADTRB3N = 28,
ADC_TRIG_TPTRGAN_0 = 31,
ADC_TRIG_TPTRG0AN_0 = 32,
ADC_TRIG_TPTRGAN_1 = 33,
ADC_TRIG_TPTRG6AN_1 = 34,
ADC_TRIG_ELCTRG0 = 48,
ADC_TRIG_SOFTWARE = 63
} adc_trig_t;

typedef enum e_adc_add
{
    ADC_ADD_OFF = 0,                // addition is turned off for chans/sensors
    ADC_ADD_TWO_SAMPLES = 1,
    ADC_ADD_THREE_SAMPLES = 2,
    ADC_ADD_FOUR_SAMPLES = 3,
    ADC_ADD_MAX
} adc_add_t;

typedef enum e_adc_align
{
    ADC_ALIGN_RIGHT = 0x0000,
    ADC_ALIGN_LEFT = 0x8000
} adc_align_t;

typedef enum e_adc_clear
{
    ADC_CLEAR_AFTER_READ_OFF = 0x0000,
    ADC_CLEAR_AFTER_READ_ON = 0x0020
} adc_clear_t;

typedef struct st_adc_cfg
{
    adc_add_t    add_cnt;
    adc_align_t  alignment;        // ignored if addition used
    adc_clear_t  clearing;
    adc_trig_t   trigger;         // default and Group A trigger source
    adc_trig_t   trigger_groupb;  // valid only for group modes
    uint8_t      priority;        // S12ADIO interrupt priority; 0-15
    uint8_t      priority_groupb; // GBADI interrupt priority; 0-15
} adc_cfg_t;
```

```
typedef enum e_adc_err    // ADC API error codes
{
    ADC_SUCCESS = 0,
    ADC_ERR_AD_NOT_CLOSED, // peripheral still running in another mode
    ADC_ERR_MISSING_PTR,   // missing required pointer argument
    ADC_ERR_INVALID_ARG,   // argument is not valid for parameter
    ADC_ERR_ILLEGAL_ARG,   // argument is illegal for mode
    ADC_ERR_SCAN_NOT_DONE  // default, Group A, or Group B scan not done
} adc_err_t;

typedef enum e_adc_cb_evt    // callback function events
{
    ADC_EVT_SCAN_COMPLETE, // normal/Group A scan complete
    ADC_EVT_SCAN_COMPLETE_GROUPB // Group B scan complete
} adc_cb_evt_t;

typedef struct st_adc_cb_args // callback arguments
{
    adc_cb_evt_t event;
} adc_cb_args_t;

typedef enum e_adc_cmd
{
    ADC_CMD_ENABLE_CHANS, // enables chans and INT(s) if priority != 0
    ADC_CMD_ENABLE_TEMP_SENSOR, // enables sensor and INT if priority != 0
    ADC_CMD_SET_SAMPLE_STATE_CNT,
    ADC_CMD_ENABLE_TRIG, // allows an async/sync trigger to start scan
    ADC_CMD_DISABLE_TRIG, // prevents an async/sync trigger to start scan
    ADC_CMD_SCAN_NOW, // issue software trigger
    ADC_CMD_DISABLE_INT, // interrupt disable; ADCSR.ADIE=0
    ADC_CMD_ENABLE_INT, // interrupt enable; ADCSR.ADIE=1
    ADC_CMD_DISABLE_INT_GROUPB, // interrupt disable; ADCSR.GBADIE=0
    ADC_CMD_ENABLE_INT_GROUPB, // interrupt enable; ADCSR.GBADIE=1
    ADC_CMD_CHECK_SCAN_DONE, // for Normal, GroupA or GroupB scan
    ADC_CMD_CHECK_SCAN_DONE_GROUPA,
    ADC_CMD_CHECK_SCAN_DONE_GROUPB,
    ADC_CMD_MAX
} adc_cmd_t;
```

```
typedef struct st_adc_ch_cfg      // bit 0 is ch0; bit 7 is ch7
{
    uint32_t    chan_mask;        // channels/bits 0-7
    uint32_t    chan_mask_groupb; // valid for group modes
    uint32_t    add_mask;         // valid if add enabled in Open()
} adc_ch_cfg_t;

typedef enum e_adc_sst_reg      // sample state registers
{
    ADC_SST_CH0 = 0,
    ADC_SST_CH1,
    ADC_SST_CH2,
    ADC_SST_CH3,
    ADC_SST_CH4,
    ADC_SST_CH5,
    ADC_SST_CH6,
    ADC_SST_CH7,
    ADC_SST_TEMPERATURE,
    ADC_SST_NUM_REGS
} adc_sst_reg_t;

typedef struct st_adc_time
{
    adc_sst_reg_t reg_id;
    uint8_t    num_states;    // default=11
} adc_time_t;

typedef enum e_adc_reg
{
    ADC_REG_CH0 = 0,
    ADC_REG_CH1 = 1,
    ADC_REG_CH2 = 2,
    ADC_REG_CH3 = 3,
    ADC_REG_CH4 = 4,
    ADC_REG_CH5 = 5,
    ADC_REG_CH6 = 6,
    ADC_REG_CH7 = 7,
    ADC_REG_TEMP = 8,
    ADC_REG_DBLTRIG = 9,
    ADC_REG_MAX
} adc_reg_t;
```

```
typedef struct st_adc_data
{
    uint16_t  chan[8];
    uint16_t  dbltrig;
} adc_data_t;
```

図 6.2 サンプルコードで使用する構造体 / 共用体 / 列挙型

6.7 大域変数一覧

表 6.5 に大域変数一覧を示します。

表 6.5 大域変数一覧

型	変数名	内容	使用関数
volatile static bool	adc_end_flg	A/D変換終了割り込み通知フラグ	main.c mian() adc_sample_callback()

6.8 関数一覧

表 6.6 に関数を示します。

表 6.6 関数一覧

関数名	ページ番号
main	18
adc_sample_led_init	18
adc_sample_adtrg_init	19
adc_sample_callback	19
R_ADC_Open	20
R_ADC_Control	21
R_ADC_Read	21
R_ADC_ReadAll	22
R_ADC_Close	22
R_ADC_GetVersion	23
adc_s12adi0_isr	23
adc_gbadi_isr	23

6.9 関数仕様

サンプルコードの関数仕様を示します。

6.9.1 main

main	
概要	ポテンションメーター入力電圧の A/D 変換
ヘッダ	—
宣言	int_t main(void);
説明	本関数は以下の処理を行います。 評価ボード上で接続されているポテンションメータの入力電圧の A/D 変換結果を 4 段階に識別し、それぞれの段階に割り当てた評価ボード上の LED 注1 に表示します。 注1. LED0、LED1、LED2、LED3
引数	なし
リターン値	なし
補足	A/D 変換開始条件は表 6.4 の ADC_SAMPLE_TRIG の値を変更することで、ソフトウェアトリガ、外部トリガのどちらか選択することができます。 デフォルトはソフトウェアトリガに設定してあります。

6.9.2 adc_sample_led_init

adc_sample_led_init	
概要	LED 接続端子の初期設定
ヘッダ	—
宣言	static void adc_sample_led_init(void);
説明	下記の端子を汎用出力ポートに設定します。 PF7 : LED0 が接続 P56 : LED1 が接続 P77 : LED2 が接続 PA0 : LED3 が接続
引数	なし
リターン値	なし

6.9.3 adc_sample_led_off

adc_sample_led_off	
概要	LED の消灯
ヘッダ	—
宣言	static void adc_sample_led_off (void);
説明	LED0、LED1、LED2、LED3 を消灯します。
引数	なし
リターン値	なし

6.9.4 adc_sample_adtrg_init

adc_sample_adtrg_init

概要	外部トリガ ADTRG#0 端子の初期設定
ヘッダ	—
宣言	static void adc_sample_adtrg_init(void);
説明	P44 端子に ADTRG#0 機能を割り当てます。
引数	なし
リターン値	なし
補足	A/D 変換開始条件が外部トリガの場合（ADC_SAMPLE_TRIG の設定値が 1）の本関数が実行されます。

6.9.5 adc_sample_callback

adc_sample_callback

概要	ADC サンプルプログラムコールバック関数
ヘッダ	—
宣言	static void adc_sample_callback(void *p_args_adc);
説明	adc_s12adi0_isr 関数からコールされる関数です。ポテンションメータの入力電圧の表示で点灯させた LED を消灯させます。
引数	なし
リターン値	なし
補足	R_ADC_Open 関数で本関数を登録してください。 A/D 変換終了割り込みを有効にしてください。

6.9.6 R_ADC_Open

R_ADC_Open

概 要	ADC ドライバの初期化関数	
ヘッダ	r_adc_rzt1_if.h	
宣 言	<pre>adc_err_t R_ADC_Open(adc_mode_t const mode, adc_cfg_t * const p_cfg, void (* const p_callback)(void *p_args));</pre>	
説 明	<p>本関数は以下の処理を行います。</p> <ul style="list-style-type: none"> • 引数の確認 • 消費電力低減機能の設定 <ul style="list-style-type: none"> - ADC ヘクロックを供給（モジュールストップ機能の S12ADCa の解除） - 温度センサヘクロックを供給 • ADC の初期設定 <ul style="list-style-type: none"> - 動作モードの設定 - A/D 変換開始条件の設定 - A/D 変換値加算モードの設定 - A/D データレジスタのフォーマットの設定 • ICUA の初期設定 <ul style="list-style-type: none"> - A/D 変換終了割り込みの優先度を設定 <p>グループ B AD 変換終了割り込みの優先度を設定</p>	
引 数	adc_mode_t const mode	ADC ドライバ初期化パラメータ ADC ドライバの動作モードを設定する。
	adc_cfg_t * const p_cfg	ADC の初期設定
	void (* const p_callback) (void *p_args)	A/D 変換終了割り込みハンドラから呼ばれるコール バック関数のアドレス
リターン値	ADC_SUCCESS	: ADC の初期化が正常に完了
	ADC_ERR_AD_NOT_CLOSED	: 二重初期化
	ADC_ERR_INVALID_ARG	: 引数が不正
	ADC_ERR_ILLEGAL_ARG	: モード設定引数が不正
	ADC_ERR_MISSING_PTR	: ポンタ引数が不正
補足	<p>すべての ADC ドライバの API 関数を実行する前に本関数を実行してください。 本関数実行後に 1us 待機してから A/D 変換を開始してください。</p>	

6.9.7 R_ADC_Control

R_ADC_Control

概要	ADCの機能設定関数	
ヘッダ	r_adc_rzt1_if.h	
宣言	adc_err_t R_ADC_Control(adc_cmd_t const cmd, void * const p_args);	
説明	S12ADCaが有する機能の設定を行います。 「adc_cmd_t 列挙型」参照	
引数	adc_cmd_t const cmd	ADCの使用する機能を設定します。
	void * const p_args	A/D変換チャンネルの設定 サンプルリングステートの設定
リターン値	ADC_SUCCESS	: 正常に機能設定が完了
	ADC_ERR_MISSING_PTR	: ポンタ引数がNULL
	ADC_ERR_INVALID_ARG	: 引数の値が不正
	ADC_ERR_ILLEGAL_ARG	: cmdが不正
	ADC_ERR_SCAN_NOT_DONE	: A/D変換未終了
補足	R_ADC_Open関数の後に実行してください。	

6.9.8 R_ADC_Read

R_ADC_Read

概要	指定した1チャンネルのA/D変換値の読み込み関数	
ヘッダ	r_adc_rzt1_if.h	
宣言	adc_err_t R_ADC_Read(adc_reg_t const reg_id, uint16_t * const p_data);	
説明	A/Dデータレジスタ、A/Dデータ2重化レジスタ、A/D温度センサデータレジスタから変換結果を読み出します。	
引数	adc_reg_t const reg_id	A/D変換値の読み込み指定チャンネル
	uint16_t * const p_data	A/D変換値を格納する変数のポインタ
リターン値	ADC_SUCCESS	: 正常終了
	ADC_ERR_INVALID_ARG	: reg_idが不正
	ADC_ERR_MISSING_PTR	: p_dataがNULL
補足	ADCが停止しているときに本関数を実行してください。	

6.9.9 R_ADC_ReadAll

R_ADC_ReadAll

概要	全チャンネルの A/D 変換値の読み込み関数	
ヘッダ	r_adc_rzt1_if.h	
宣言	adc_err_t R_ADC_ReadAll(adc_data_t * const p_all_data);	
説明	A/D データレジスタ全チャンネル、A/D データ 2 重化レジスタから変換結果を読み出します。	
引数	adc_data_t * const p_all_data	A/D 変換格納よう配列の先頭アドレス
リターン値	ADC_SUCCESS ADC_ERR_MISSING_PTR	: 正常終了 : p_all_data が NULL
補足	ADC が停止しているときに本関数を実行してください。	

6.9.10 R_ADC_Close

R_ADC_Close

概要	ADC ドライバの終了処理関数	
ヘッダ	r_adc_rzt1_if.h	
宣言	void R_ADC_Close(void);	
説明	<p>本関数は以下の処理を行います。</p> <ul style="list-style-type: none"> ICUA の終了処理 <ul style="list-style-type: none"> - S12ADI の割り込みを禁止に設定 - S12GBADI の割り込みを禁止に設定 ADC の終了処理 <ul style="list-style-type: none"> - A/D コントロールレジスタをリセット後の値に設定 温度センサの終了処理 <ul style="list-style-type: none"> - 温度センサコントロールレジスタをリセット後の値に設定 消費電力低減機能の設定 <ul style="list-style-type: none"> - 温度センサへクロック供給を停止 <p>ADC ヘクロック供給を停止（モジュールストップ機能の S12ADCa の禁止）</p>	
引数	なし	
リターン値	なし	
補足	<p>R_ADC_Open 関数の後に実行してください。</p> <p>A/D 変換開始条件に非同期トリガ、または同期トリガを選択している場合は、トリガを停止してから本関数を実行してください。</p> <p>ELC、EMU2 を使用している場合は、下記の設定をしてから本関数を実行してください。</p> <p>ELC で ADC からのイベント入力を禁止</p> <p>EMU2 で ADC からのデータ転送を禁止</p>	

6.9.11 R_ADC_GetVersion

R_ADC_GetVersion

概要	ADC ドライババージョン情報取得関数
ヘッダ	r_adc_rzt1_if.h
宣言	uint32_t R_ADC_GetVersion(void);
説明	ADC ドライバのバージョンを戻り値として返します。
引数	なし
リターン値	ADC ドライバのバージョン

6.9.12 adc_s12adi0_isr

adc_s12adi0_isr

概要	A/D 変換終了割り込みハンドラ
ヘッダ	—
宣言	void adc_s12adi0_isr (void);
説明	R_ADC_Open 関数で登録した、コールバック関数を呼びだします。
引数	なし
リターン値	なし

6.9.13 adc_gbadi_isr

adc_gbadi_isr

概要	グループ B A/D 変換終了割り込みハンドラ
ヘッダ	—
宣言	static void adc_gbadi_isr(void);
説明	R_ADC_Open 関数で登録した、コールバック関数を呼びだします。
引数	なし
リターン値	なし

6.10 フローチャート

6.10.1 メイン処理

図 6.3 にメイン処理のフローチャートを示します。

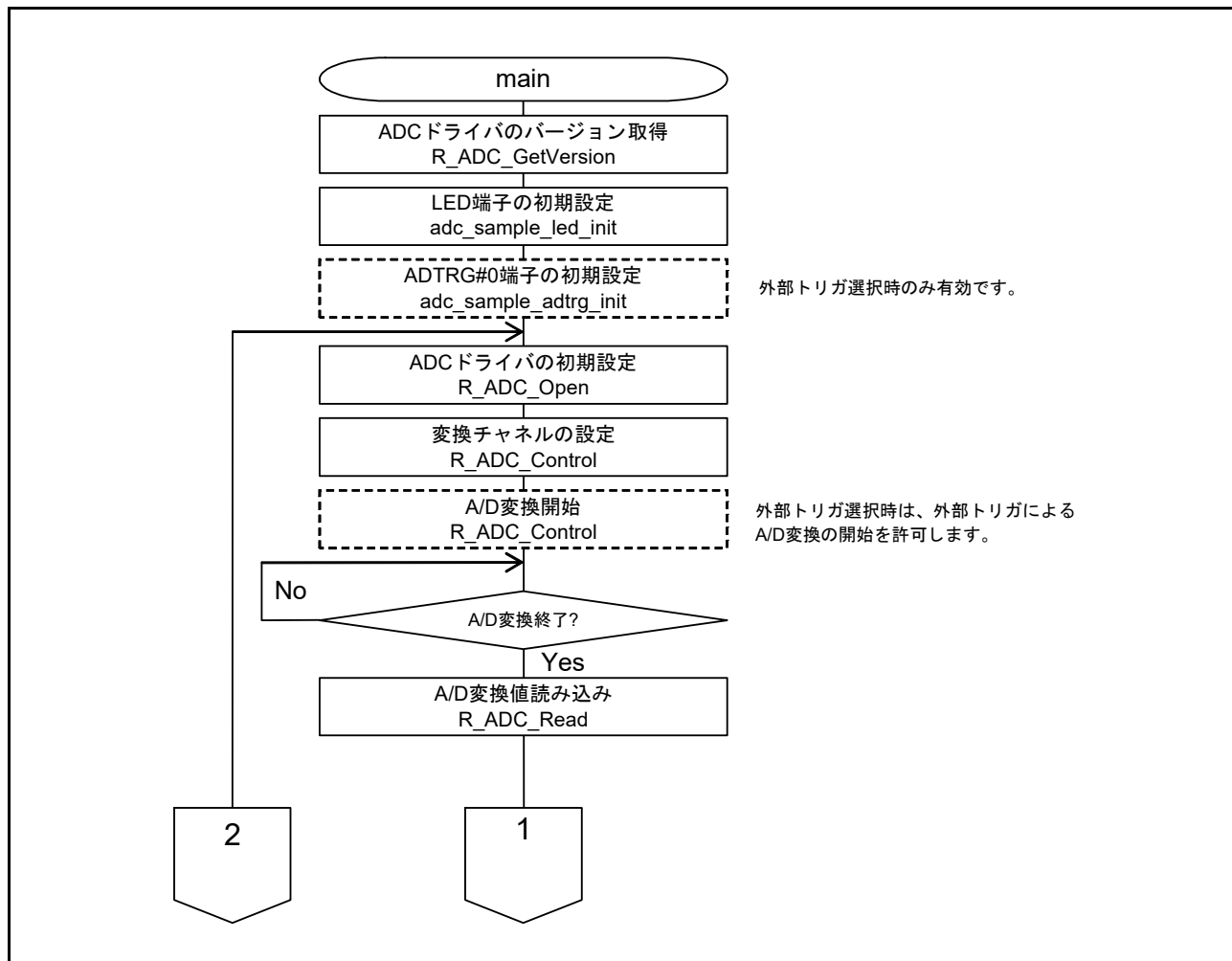


図 6.3 メイン処理 (1/2)

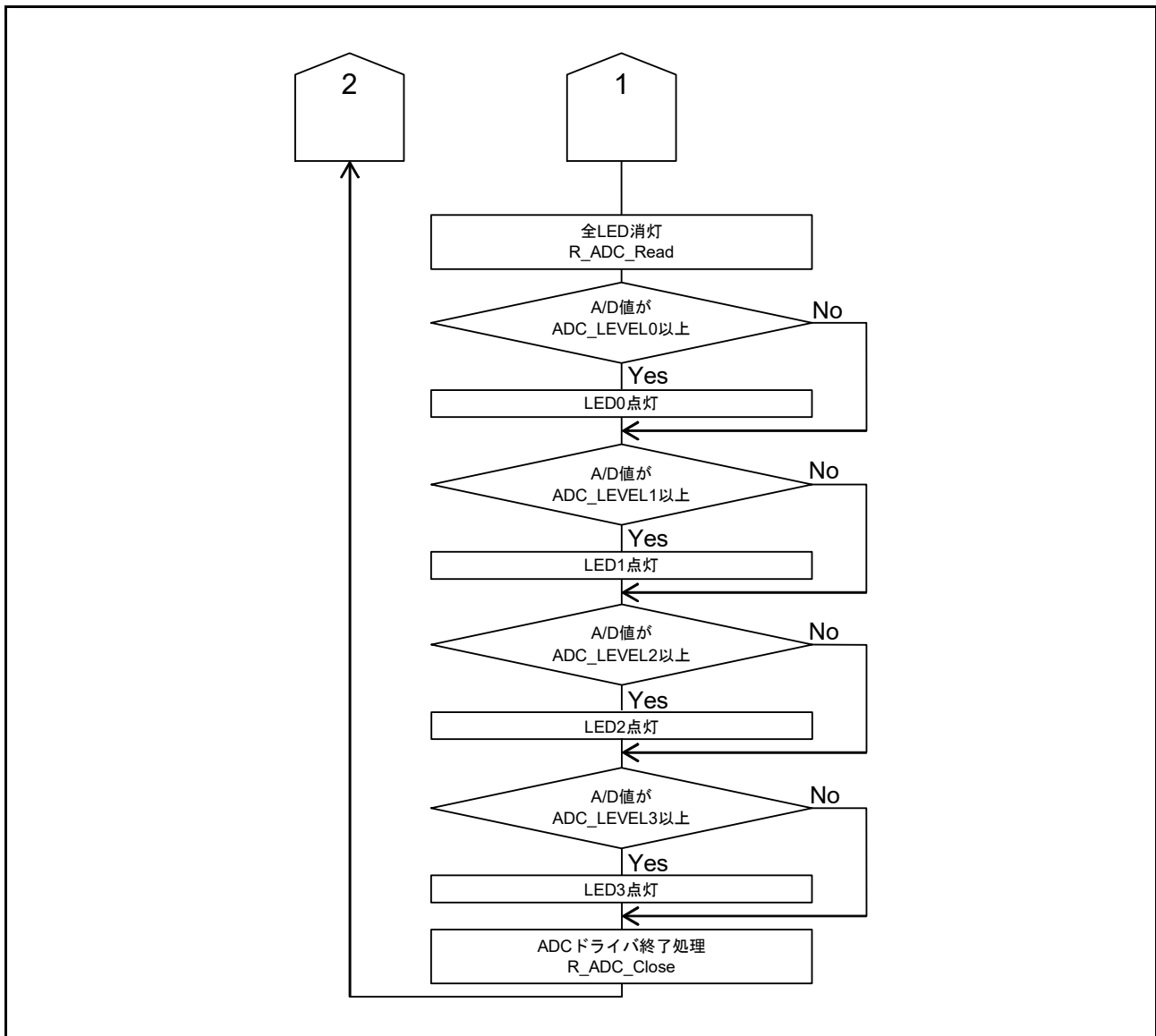


図 6.3 メイン処理 (2/2)

6.10.2 adc_sample_led_init

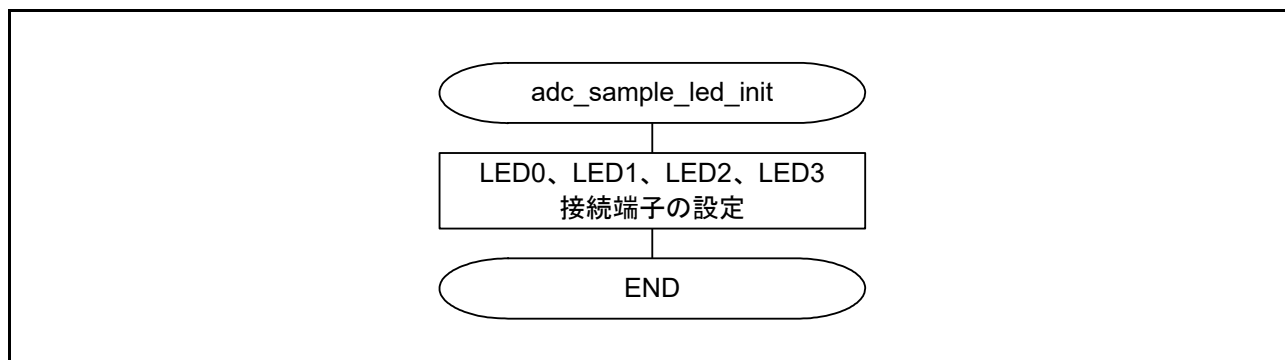


図 6.4 adc_sample_led_init

6.10.3 adc_sample_adtrg_init

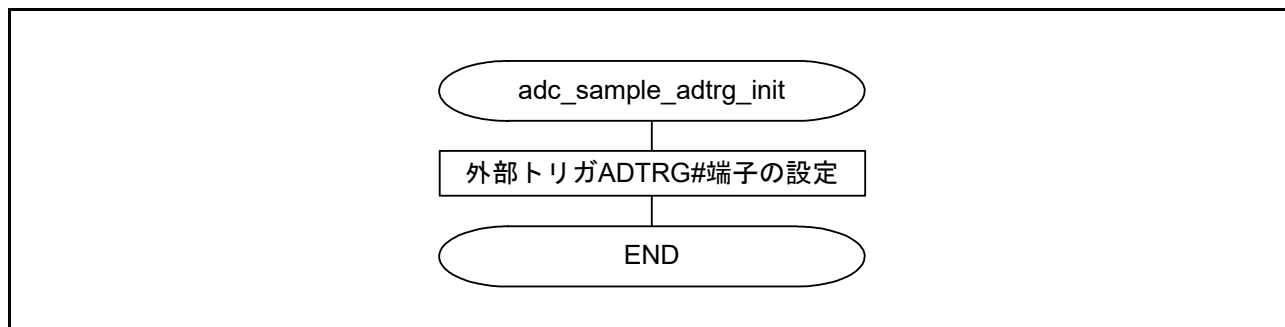


図 6.5 adc_sample_adtrg_init

6.10.4 adc_sample_callback

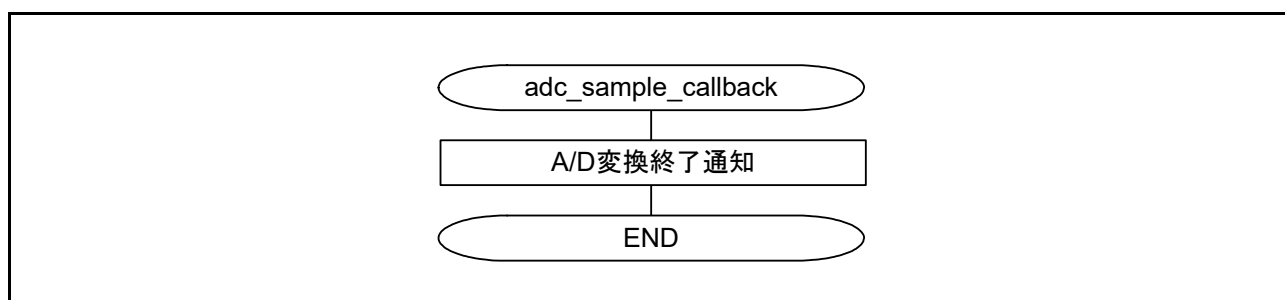


図 6.6 adc_sample_callback

6.10.5 adc_s12adi0_isr

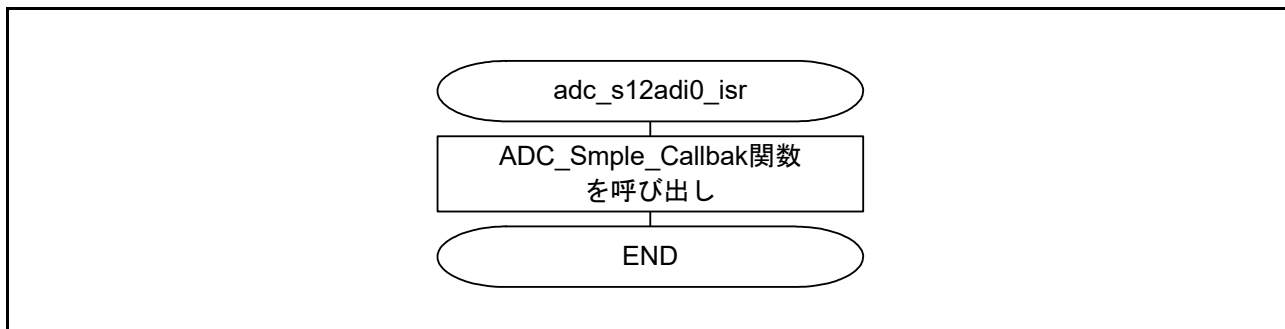


図 6.7 adc_s12adi0_isr

6.11 R_ADC_Control コマンド一覧

R_ADC_Control 関数で使用するコマンド一覧を以下に示します。

表6.7 コマンド一覧

コマンド	概要
ADC_CMD_ENABLE_CHANS	A/D変換チャンネルの指定
ADC_CMD_ENABLE_TEMP_SENSOR	温度センサの初期設定
ADC_CMD_SET_SAMPLE_STATE_CNT	アナログ入力のサンプリング時間の設定
ADC_CMD_ENABLE_TRIG	同期、非同期トリガによるA/D変換の開始を許可
ADC_CMD_DISABLE_TRIG	同期、非同期トリガによるA/D変換の開始を禁止
ADC_CMD_SCAN_NOW	ソフトウェアトリガによるA/D変換を開始
ADC_CMD_ENABLE_INT	スキャン終了後、S12ADI割り込み発生を許可
ADC_CMD_DISABLE_INT	スキャン終了後、S12ADI割り込み発生を禁止
ADC_CMD_ENABLE_INT_GROUPB	グループBのスキャン終了後にS12GBADI割り込み発生を許可
ADC_CMD_DISABLE_INT_GROUPB	グループBのスキャン終了後にS12GBADI割り込み発生を禁止
ADC_CMD_CHECK_SCAN_DONE	A/D変換の確認
ADC_CMD_CHECK_SCAN_DONE_GROUPA	グループAのスキャンを確認
ADC_CMD_CHECK_SCAN_DONE_GROUPB	グループBのスキャンを確認

6.11.1 ADC_CMD_ENABLE_CHANS

ADC_CMD_ENABLE_CHANS

概要	A/D変換チャンネルの指定		
ヘッダ	r_adc_rzt1_if.h		
説明	A/D変換のチャンネルを指定します。 パラメータはadc_ch_cfg_t型変数の形で受け渡します。		
パラメータ	adc_ch_cfg_t	p_args	A/D変換を行うチャンネルを指定します。
リターン値	ADC_SUCCESS		: チャンネルの指定に成功
	ADC_ERR_MISSING_PTR		: ポンタ引数がNULL
	ADC_ERR_ILLEGAL_ARG		: R_ADC_Open関数のmodeが ADC_MODE_SS_TEMPERATUREの場合
	ADC_ERR_INVALID_ARG		: 引数の値が不正
補足	—		

6.11.2 ADC_CMD_ENABLE_TEMP_SENSOR

ADC_CMD_ENABLE_TEMP_SENSOR

概要	温度センサの初期設定		
ヘッダ	r_adc_rzt1_if.h		
説明	温度センサの初期設定を行います。 パラメータはありません。NULLを指定してください。		
パラメータ	NULL		
リターン値	ADC_SUCCESS		: 温度センサの初期化に成功
	ADC_ERR_ILLEGAL_ARG		: R_ADC_Open関数のmodeが ADC_MODE_SS_TEMPERATURE以外の場合
補足	—		

6.11.3 ADC_CMD_SET_SAMPLE_STATE_CNT

ADC_CMD_SET_SAMPLE_STATE_CNT

概要	アナログ入力のスプリング時間の設定		
ヘッダ	r_adc_rzt1_if.h		
説明	アナログ入力のスプリングの時間を設定します。 パラメータはadc_time_t型変数の形で受け渡します。		
パラメータ	adc_time_t	p_args	スプリング時間を設定するチャンネルと、スプリング時間を指定します。 adc_time_t構造体を参照
リターン値	ADC_SUCCESS		: アナログ入力のスプリング時間の設定に成功
	ADC_ERR_MISSING_PTR		: ポンタ引数がNULL
	ADC_ERR_ILLEGAL_ARG		: 引数の値が不正
補足	—		

6.11.4 ADC_CMD_ENABLE_TRIG

ADC_CMD_ENABLE_TRIG

概要	同期、非同期トリガによる A/D 変換の開始を許可	
ヘッダ	r_adc_rzt1_if.h	
説明	同期、非同期トリガによる A/D 変換の開始を許可します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: 同期、非同期トリガによる A/D 変換の開始の許可に成功
補足	—	

6.11.5 ADC_CMD_DISABLE_TRIG

ADC_CMD_DISABLE_TRIG

概要	同期、非同期トリガによる A/D 変換の開始を禁止	
ヘッダ	r_adc_rzt1_if.h	
説明	同期、非同期トリガによる A/D 変換の開始を禁止します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: 同期、非同期トリガによる A/D 変換の開始の禁止に成功
補足	—	

6.11.6 ADC_CMD_SCAN_NOW

ADC_CMD_SCAN_NOW

概要	ソフトウェアトリガによる A/D 変換を開始	
ヘッダ	r_adc_rzt1_if.h	
説明	ソフトウェアトリガによる A/D 変換を開始します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: A/D 変換の開始に成功
	ADC_ERR_SCAN_NOT_DONE	: A/D 変換中
補足	—	

6.11.7 ADC_CMD_ENABLE_INT

ADC_CMD_ENABLE_INT

概要	スキャン終了後、S12ADI 割り込み発生 の許可	
ヘッダ	r_adc_rzt1_if.h	
説明	スキャン終了後、S12ADI 割り込み発生を許可します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: スキャン終了後、S12ADI 割り込み発生 の許可に成功
	ADC_ERR_ILLEGAL_ARG	: コールバック関数が未登録
補足	—	

6.11.8 ADC_CMD_DISABLE_INT

ADC_CMD_DISABLE_INT

概要	スキャン終了後、S12ADI 割り込み発生 の禁止	
ヘッダ	r_adc_rzt1_if.h	
説明	スキャン終了後、S12ADI 割り込み発生を禁止します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: スキャン終了後、S12ADI 割り込み発生 の禁止に成功
補足	—	

6.11.9 ADC_CMD_ENABLE_INT_GROUPB

ADC_CMD_ENABLE_INT_GROUPB

概要	グループ B のスキャン終了後に S12GBADI 割り込み発生を許可	
ヘッダ	r_adc_rzt1_if.h	
説明	グループ B のスキャン終了後に S12GBADI 割り込み発生を許可します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: グループ B のスキャン終了後の S12GBADI 割り込み発生 の許可に成功
	ADC_ERR_ILLEGAL_ARG	: コールバック関数が未登録
補足	—	

6.11.10 ADC_CMD_DISABLE_INT_GROUPB

ADC_CMD_DISABLE_INT_GROUPB

概要	グループ B のスキャン終了後に S12GBADI 割り込み発生を禁止	
ヘッダ	r_adc_rzt1_if.h	
説明	グループ B のスキャン終了後に S12GBADI 割り込み発生を禁止します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: グループ B のスキャン終了後の S12GBADI 割り込み発生 の禁止に成功
補足	—	

6.11.11 ADC_CMD_CHECK_SCAN_DONE

ADC_CMD_CHECK_SCAN_DONE

概要	A/D 変換の確認	
ヘッダ	r_adc_rzt1_if.h	
説明	A/D 変換中か確認します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: A/D 変換が終了
	ADC_ERR_SCAN_NOT_DONE	: A/D 変換中
補足	—	

6.11.12 ADC_CMD_CHECK_SCAN_DONE_GROUPA

ADC_CMD_CHECK_SCAN_DONE_GROUPA

概要	グループ A のスキャンを確認します。	
ヘッダ	r_adc_rzt1_if.h	
説明	グループ A のスキャンが終了したか確認します。 パラメータはありません。NULL を指定してください。	
パラメータ	NULL	
リターン値	ADC_SUCCESS	: グループ A のスキャンが終了
	ADC_ERR_SCAN_NOT_DONE	: グループ A がスキャン中
補足	—	

6.11.13 ADC_CMD_CHECK_SCAN_DONE_GROUPB

ADC_CMD_CHECK_SCAN_DONE_GROUPB

概要	グループ B のスキャンを確認します。
ヘッダ	r_adc_rzt1_if.h
説明	グループ B のスキャンが終了したか確認します。 パラメータはありません。NULL を指定してください。
パラメータ	NULL
リターン値	ADC_SUCCESS : グループ B のスキャンが終了 ADC_ERR_SCAN_NOT_DONE : グループ B のスキャン中
補足	—

7. 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。

8. 参考ドキュメント

- ユーザーズマニュアル：ハードウェア

RZ/T1 グループ ユーザーズマニュアルハードウェア編

(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。)

RZ/T1 Evaluation Board RTK7910022C00000BR ユーザーズマニュアル

(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／テクニカルニュース

(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。)

- ユーザーズマニュアル：開発環境

IAR 統合開発環境 (IAR Embedded Workbench® for Arm) に関しては、IAR ホームページから入手してください。

(最新版を IAR 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/>

お問合せ先

<http://japan.renesas.com/contact/>

改訂記録

ADC サンプルプログラム アプリケーションノート

Rev.	発行日	改訂内容			
		ページ	ポイント		
0.10	2015.03.06	—	初版発行		
1.00	2015.04.10	—	Web掲載に際しRevのみ変更		
1.10	2015.07.06	2. 動作環境			
		5	表2.1 動作環境 統合開発環境 表記一部修正、追加		
		6. ソフトウェア説明			
		10	6.2.4 説明文 参照を追加		
		10	表6.2 タイトル、サイズを一部修正		
		11	表6.3 追加		
1.20	2015.12.03	2. 動作環境			
		5	表2.1 動作環境 統合開発環境 一部修正		
		1.30	2017.07.13	全体	Cortex-R4F → Cortex-R4に変更
		1		要旨 本文を変更：S12ADCaのユニットを追加、制御するLEDを変更	
2. 動作環境					
5	表2.1 動作環境 統合開発環境の内容変更				
5. ハードウェア説明					
8	図5.1 ハードウェア構成例 I/Oポートと制御するLEDを変更				
1.40	2018.06.07	6. ソフトウェア説明			
		—	6.2.4 必要メモリサイズ 削除		
		2.動作環境			
5	表2.1 動作環境 統合開発環境の内容変更				
8. 参考ドキュメント					
35	IAR 統合開発環境名変更				

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。

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）がありません。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うと、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)



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/>