

RL78/I1E

R01AN2819JJ0100

アナログ特性評価用サンプルコード仕様書

Rev.1.00

2015.11.09

要旨

本ドキュメントは、RL78/I1E (R5F11CCC)用サンプルコードの仕様を説明します。

動作確認デバイス

RL78/I1E (R5F11CCC)

目次

1. 仕様.....	2
2. ハードウェア説明	3
3. API 関数一覧.....	6
4. API 定義.....	8
5. アナログ・フロントエンド関連 API.....	26
6. フラッシュ・メモリ関連 API.....	41
7. UART 通信関連 API	43
8. フリーランタイム関連 API.....	46
9. キー判定関連 API.....	47
10. サンプルコード動作仕様.....	53
11. 設定変更手順.....	66
12. API 使用時の注意事項.....	70

1. 仕様

1.1 概要

本ドキュメントで説明するサンプルコードは、主にアナログ・フロントエンド回路 のプログラマブル・ゲイン計装アンプ付き 24 ビット $\Delta \Sigma$ A/D コンバータ(以下 $\Delta \Sigma$ A/D)の制御と PC との通信処理を行います。

$\Delta \Sigma$ A/D の制御としては、A/D 変換、ゲイン誤差測定、オフセット調整、自動ゲイン調整等の処理を行います。

1.2 動作確認条件

サンプルコードは、以下の条件で動作を確認しています。

表 1-1 動作条件一覧

項目	内容
評価ボード	<ul style="list-style-type: none"> ・ RL78/I1E アナログ評価ボード [RTK50011CCC00000BR] ・ RL78/I1E CPU ボード [FB-R5F11CCC-TB]
使用マイコン	R5F11CCC (RL78/I1E) 36 ピン
動作周波数	32MHz
動作電圧	5.0V
統合開発環境 (CS+ for CA,CX)	V3.01.00 [19 Aug 2015]
C コンパイラ (CS+)	CA78K0R V5.00.00.02 [03 Jul 2014]
RL78/I1E コードライブラリ(CS+)	V1.02.00.06 [12 Aug 2015] 注 ¹
データ・フラッシュ・ライブラリ	RL78 ファミリ データ・フラッシュ・ライブラリ Type04 Ver.1.05

注 1 : CS+用のコードライブラリはコード生成プラグインに内包されています。本ドキュメントでは「CS+ Code_Generator for RL78_78K V2.05.00」で動作確認を行っております。

2. ハードウェア説明

2.1 ハードウェアリソース一覧

表 2-1、表 2-2 に本サンプルコードで使用する周辺ハードウェアの設定を示します。

表 2-1 ハードウェアリソース一覧 (1/2)

周辺機能	設定		
クロック発生回路 └クロック	動作モード	高速メイン・モード 2.7(V) ≤VDD≤5.5(V)	
	メインシステムクロック (fMAIN)	高速オンチップオシレータクロック (fHOCO)	
	24 ビット ΔΣ A/D コンバータの動作クロック (fDSAD) ソースの設定	高速オンチップオシレータクロック (fHOCO)	
	RTC の動作クロック (fRTC) ソースの設定	高速オンチップオシレータクロック (fHOCO)	
	高速オンチップ・オシレータ・クロック (fHOCO) 設定	32(MHz)	
	高速システム・クロック (fMX) 設定	なし	
	24 ビット ΔΣ A/D コンバータの動作クロック (fDSAD) の設定	fHOCO 32(MHz)	
	RTC, インターバル・タイマ/タイマ RJ 動作クロック	fIL 15(kHz)	
	CPU と周辺クロック (fCLK)	fHOCO 32000(kHz)	
ポート機能	Port13	P13.7 (INTP)	
タイマ・アレイ・ユニット └ユニット 1	チャンネル 0	インターバル・タイマ	
	インターバル時間	10(ms)	
	割り込み	タイマ・チャンネル 0 のカウント完了で割り込み発生 レベル 3(低優先順位)	
タイマ・アレイ・ユニット └ユニット 1	チャンネル 1	インターバル・タイマ	
	動作モード設定	16 ビット	
	インターバル・タイマ設定	100(us)	
	割り込み	タイマ・チャンネル 1 のカウント完了で割り込み発生 レベル 3(低優先順位)	
ウォッチドッグ・タイマ	ウォッチドッグ・タイマ動作設定	使用しない	
PGA+ΔΣ A/D コンバータ	使用マルチプレクサ設定	マルチプレクサ 0 差動入力モード その他マルチプレクサ使用しない	
	SBIAS 出力電圧	2.0V	
	断線検知	使用しない	
	ΔΣ A/D コンバータ動作モード設定	通常動作	
	ΔΣ A/D コンバータ開始トリガ設定	ソフトウェアトリガ	
	オートスキャンモード設定	シングルスキャン	
	割り込み設定	ΔΣ A/D の変換割り込み許可 (INTDSAD)	レベル 3(低優先順位)
		ΔΣ A/D のスキャン割り込み許可 (INTDSADS)	レベル 3(低優先順位)
		レベル 3(低優先順位)	

表 2-2 ハードウェアリソース一覧 (2/2)

周辺機能		設定
PGA+ Δ Σ A/D コンバータ └マルチプレクサ 0	ゲイン設定 G _{SET1}	1 倍
	ゲイン設定 G _{SET2}	8 倍
	オフセット調整電圧設定	16 (0mV)
	オーバ・サンプリング比	256
	A/D 変換回数	PGA0CTL2 レジスタの設定値で 1~8092 回を指定 1 (1 回)
	平均化処理	平均化処理を行わない
シリアル └SAU0 └チャンネル	チャンネル 1	UART0 送信/受信機能
	データ・ビット長	8 ビット
	データ転送方向	LSB
シリアル └SAU0 └UART1 └受信	パリティ	パリティなし
	ストップ・ビット長	1 ビット
	受信データ・レベル	標準
	転送レート	1000000(bps)
	割り込み	受信完了割り込み設定(INTSR1) レベル 3(低優先順位)
	コールバック機能	受信完了
	転送モード	連続転送モード
シリアル └SAU0 └UART1 └送信	データ・ビット長	8 ビット
	データ転送方向	LSB
	パリティ	パリティなし
	ストップ・ビット長	1 ビット
	送信データ・レベル	標準
	転送レート	1000000(bps)
	割り込み	送信完了割り込み設定(INTST1) レベル 3(低優先順位)
	コールバック機能	送信完了
	転送モード	連続転送モード
データ・トランスファ・コントロール	DTC ベースアドレス	0xffd00
	コントロールデータ 0(DTCD0)	UART1 受信
	コントロールデータ 1(DTCD1)	UART1 送信
データ・トランスファ・コントロール └DTCD0	転送モード設定	ノーマル・モード
	転送データ・サイズ設定	8 ビット
	転送元アドレス	0xFF46(固定)
	転送先アドレス	0xF900(加算)
	転送回数	1
	ブロックサイズ	1
データ・トランスファ・コントロール └DTCD1	転送モード設定	ノーマル・モード
	転送データ・サイズ設定	8 ビット
	転送元アドレス	0xF910(加算)
	転送先アドレス	0xFF44(固定)
	転送回数	1
	ブロックサイズ	1
割り込み └外部割り込み	INT10	INTP0 使用 立下りエッジ検出 レベル 3(低優先順位)

2.2 メモリアドレス空間

本サンプルコードで使用しているメモリアドレス空間を 図 2.1 に示します。

データ・フラッシュ・ライブラリおよび DTC テーブルでは、RAM 上の固定番地の領域を確保する必要があります。設定は、リンク・ディレクティブ・ファイル (*.dr) を作成し CS+への登録を行います。リンク・ディレクティブ・ファイルの詳細は「4.7 リンク・ディレクティブ・ファイル」を参照してください。

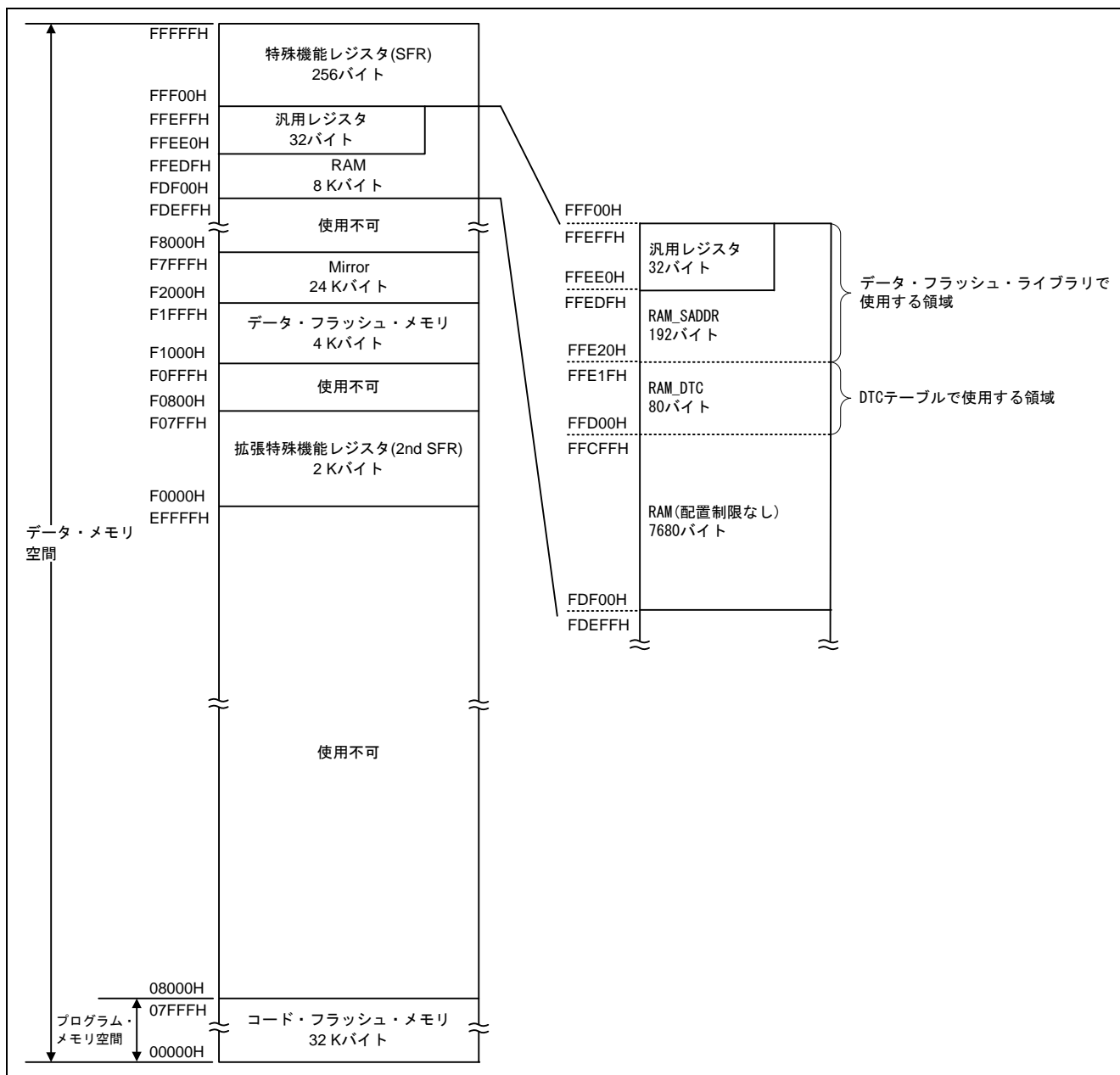


図 2.1 本サンプルコードで使用しているメモリアドレス空間

3. API 関数一覧

3.1 アナログ・フロントエンド関連

表 3-1 にアナログ・フロントエンド関連の API 関数一覧を示します。

ソースファイルは、`r_rl78_i1e_common.c` を参照してください。

表 3-1 アナログ・フロントエンド関連 API 関数一覧

関数名	概要
<code>R_I1E_Variable_Initialize</code>	サンプルコード使用変数初期化処理
<code>R_I1E_RingBuffer_Initialize</code>	DSAD リングバッファ初期化関数
<code>R_I1E_AFE_Calibration</code>	AFE モジュール校正実行関数
<code>R_I1E_PGA_DSAD_GetResult_1Shot</code>	24 ビット $\Delta \Sigma$ A/D コンバータ・指定 ch 1 ショット差動入力 A/D 変換実行関数
<code>R_I1E_PGA_DSAD_GainRegSet</code>	24 ビット $\Delta \Sigma$ A/D コンバータ・ゲイン設定関数
<code>R_I1E_PGA_DSAD_OffsetRegSet</code>	24 ビット $\Delta \Sigma$ A/D コンバータオフセット設定関数
<code>R_I1E_PGA_DSAD_OsrRegSet</code>	24 ビット $\Delta \Sigma$ A/D コンバータ・OSR 設定関数
<code>R_I1E_PGA_DSAD_SettingRegGet</code>	オートスキャンモード、各入力マルチプレクサ設定保持変数格納関数
<code>R_I1E_PGA_DSAD_SettingRegSet</code>	DSAD 設定に構造体変数情報を設定
<code>R_I1E_PGA_DSAD_CorrectValue</code>	PGA オフセット誤差補正適用値取得関数
<code>R_I1E_PGA_DSAD_GetValue</code>	24 ビット $\Delta \Sigma$ A/D コンバータ・変換値取得関数
<code>R_I1E_PGA_DSAD_AutoGainInit</code>	PGA 自動ゲイン用変数初期化
<code>R_I1E_PGA_DSAD_AutoGainBufCheck</code>	24 ビット $\Delta \Sigma$ A/D コンバータ変換完了後 PGA 自動ゲイン調整検査処理関数
<code>R_I1E_PGA_DSAD_AutoGainExecute</code>	24 ビット $\Delta \Sigma$ A/D コンバータ・PGA 自動ゲイン調整関数
<code>R_I1E_PGA_DSAD_OffsetAdjustment</code>	PGA オフセット調整実行関数
<code>R_I1E_PGA_DSAD_DisconnCheck</code>	断線検知チェック関数
<code>R_I1E_AFE_Stop</code>	RL78/I1E AFE 停止処理関数
<code>R_I1E_AFE_ReStart</code>	RL78/I1E AFE 再始動処理関数
<code>R_I1E_CAMP_Calibration</code>	コンフィギュラブル・アンプ トリミング実行関数

3.2 フラッシュ・メモリ関連

表 3-2 にフラッシュ・メモリ関連の API 関数一覧を示します。

ソースファイルは、`r_rl78_i1e_common.c` を参照してください。

表 3-2 フラッシュ・メモリ関連 API 関数一覧

関数名	概要
<code>R_I1E_FlashCheck</code>	データフラッシュ・メモリ格納データチェック関数
<code>R_I1E_FlashRewrite</code>	データフラッシュ・メモリ格納データ上書き関数

【注】 `R_FlashXXXX` の名前の関数については、別途「データフラッシュライブラリ Type04」の関連ドキュメントをご参照ください。

3.3 UART 通信関連

表 3-3 に UART 通信関連の API 関数一覧を示します。

ソースファイルは、`r_rl78_ile_common.c` を参照してください。

表 3-3 UART 通信関連 API 関数一覧

関数名	概要
R_I1E_UartSend	PC 送信処理設定関数(ASCII)。[1 回の送信で 256byte までの制限付き]
R_I1E_UartSendBinary	PC 送信処理設定関数(Binary)。[1 回の送信で 256byte までの制限付き]
R_I1E_UartReceive	PC からの受信処理(DTC0 使用)。[1 回の受信で 256byte までの制限付き]

3.4 フリーランタイム関連

表 3-4 にフリーランタイム関連の API 関数一覧を示します。

ソースファイルは、`r_rl78_common_util.c` を参照してください。

表 3-4 フリーランタイム関連 API 関数一覧

関数名	概要
R_TAU_FreeRunTimerInit	フリーランタイム・設定初期化関数
R_TAU_FreeRunTimerStop	フリーランタイム・停止関数
R_GetTickCount	フリーランタイム・現在カウント値取得関数
R_CmpTickCount	フリーランタイム・カウント値比較関数

3.5 キー判定関連

表 3-5 にキー判定関連の API 関数一覧を示します。

ソースファイルは、`r_keyscan.c` を参照してください。

表 3-5 キー判定関連 API 関数一覧

関数名	概要
R_KEY_Scan	キースキャン処理
R_KEY_Initialize	キー情報初期化
R_KEY_Get	キー情報取得 (指定キーのみ)
R_KEY_GetAll	キー情報取得 (全キー)
R_KEY_WaitOneClick	指定キー押下待ち

4. API 定義

本章では各 API 関数で使用する定義について示します。

4.1 コード生成機能の定義

本サンプルコードでは CS+ のコード生成機能で生成されるヘッダファイル[r_cg_macrodriver.h]をインクルードし、以下の型を用いています。

表 4-1 型定義一覧

定義	値
int8_t	signed char
uint8_t	unsigned char
int16_t	signed short
uint16_t	unsigned short
int32_t	signed long
uint32_t	unsigned long
MD_STATUS	unsigned short

本サンプルコードに含まれる API (Application Program Interface ; 以下 API) 関数は、一部を除き、共通した型でステータス値を返します。ユーザーアプリケーションでは、この返り値を判断し、正常であれば処理の続行、エラーであれば訂正処理を行う必要があります。

表 4-2 本サンプルコードで使用する MD_STATUS 型の戻り値一覧

型名	マクロ名	定数値	内容
MD_STATUS	MD_OK	00H	正常終了
	MD_ERROR	80H	エラー
	MD_ARGERROR	81H	パラメータ・エラー

4.2 ユーザー環境依存設定用マクロ宣言

本 API ではユーザー環境、使用条件に依存する部分をマクロ宣言にて定義しています。ご使用の環境に応じて各定義を変更してください。

(a) r_cg_main.c

表 4-3 ユーザー環境依存設定用マクロ宣言一覧

マクロ宣言	デフォルト設定値	入力範囲	内容
D_BULK_NUM	488U	1 以上	BULKSTART コマンドから BULKEND コマンドまでの転送データの総数を指定
D_BULK_COMMAND_NUM	20U ^{注1}	1 以上	1 回の BULK コマンド毎の転送データ数を指定
D_STREAMHEADER	(ヘッダ文字列) ^{注2}		STREAMHEADER 送信文字列定義

注 1 送信バッファサイズ (D_DSAD_VALUE_BUFFER_SIZE) に収まる範囲で設定してください

注 2 STREAM 転送のヘッダ送信の項を参照してください

(b) r_rl78_i1e_common.h

表 4-4 ユーザー環境依存設定用マクロ宣言一覧

マクロ宣言	デフォルト設定値	入力範囲	内容
D_MCU_CLOCK_MHZ	32U	uint8_t ^{注1}	CPU と周辺クロック周波数の定義。 ^{注1} ^{注2} API 内部で使用するソフトウェアウエイト の大きな時間算出に使用します。 (単位:MHz)
D_FLASH_MEMORY_DATA_USE	1U	0U,1U	フラッシュ・メモリデータ使用設定 0:使用しない 1:使用する
D_FLASH_FORCE_WRITING	0U	0U,1U	フラッシュデータ強制書き込み設定 0:強制書き換えしない。 (フラッシュ値無効時のみ書き換える) 1:強制書き換えする
D_DSAD_CORRECT_USE	1U	0U,1U	PGA 誤差測定有効化設定 0:PGA 誤差測定無効 1:PGA 誤差測定有効
D_DSAD_CORRECT_MPXn	D_PGA_DSAD_MP X0	MPX0- MPX4 ^{注3}	PGA 誤差測定で使用する入力マルチプレ クサ番号 ^{注3}
D_DSAD_VALUE_BUFFER_SIZE	256U	uint16_t	DSAD 変換値格納バッファサイズ
D_DSAD_AUTO_GAIN_USE	1U	0U,1U	PGA 自動ゲイン調整有効化設定 0:PGA 自動ゲイン調整無効 1:PGA 自動ゲイン調整有効
D_DSAD_AUTO_GAIN_TRIGGER_SEC	5U ^{注4}	uint8_t	PGA 自動ゲイン調整タイミング (単位:秒)
D_GAIN_ERROR_REFERENCE_mV	10.0F	float	PGA ゲイン誤差測定基準電圧 (単位:mV)
D_DISCONNECTION_CHECK_COUNT	3U ^{注1}	uint8_t	断線チェック回数(1 回以上)
D_DISCONNECTION_THRESHOLD_mV	10.0F	float	断線判定電圧(単位:mV)
D_UART_SEND_USE	1U	0U,1U	UART 送信有効化設定 0:UART 送信無効 1:UART 送信有効
D_UART_SEND_BUFFER_SIZE	256U	Max.256	PC 送信用送信バッファサイズ

【注 1】 0 より大きな値を指定してください

【注 2】 使用するマイコンの CPU クロックの設定値を指定してください

【注 3】 入力マルチプレクサ番号の define 宣言値を指定してください

D_PGA_DSAD_MPX0 … 入力マルチプレクサ 0

D_PGA_DSAD_MPX1 … 入力マルチプレクサ 1

D_PGA_DSAD_MPX2 … 入力マルチプレクサ 2

D_PGA_DSAD_MPX3 … 入力マルチプレクサ 3

【注 4】 実際の設定は CPU クロック周波数を基にしたカウンタ値を求める式が付加されます。

(5U*D_MCU_CLOCK_MHZ*1000000)

(c) r_keyscan.h

表 4-5 ユーザー環境依存設定用マクロ宣言一覧

マクロ宣言	デフォルト設定値	入力範囲	内容
DEF_KEY_ACTIVE	DEF_KEY_ACTIVE_LOW	DEF_KEY_ACTIVE_LOW, DEF_KEY_ACTIVE_HIGH	SW 有効レベル
KEY_SCAN_NORM	10U	uint8_t	単押し判定時間 (単位:10ms)
KEY_SCAN_LONG	100U	uint8_t	長押し判定時間 (単位:10ms)
KEY_SCAN_DEAD	5U	uint8_t	キー状態変更後の不感帯期間 (単位:10ms)
KEY_SCAN_NOT	5U	uint8_t	非アクティブ期間 (単位:10ms)
KEY_SCAN_DOUBLE	50U	uint8_t	ダブルクリック判定有効期間 (単位:10ms)

(d) r_cg_userdefine.h

表 4-6 ユーザー環境依存設定用マクロ宣言一覧

マクロ宣言	デフォルト設定値	入力範囲	内容
D_DEBUG_LED_USE	0U	0U,1U	デバッグ用 LED 使用設定 0U: 使用しない 1U: 使用する
D_DEBUG_LED_PORT	P1.5	(出力ポート) 注 1	デバッグ用 LED ポート設定

【注 1】 プルアップされた LED と接続しているデジタル出力ポートを指定してください

4.3 マクロ宣言

本節では API で定義されているマクロ宣言について示します。

(a) r_cg_main.c

表 4-7 マクロ宣言一覧

マクロ宣言	値	内容
D_BULK_SPLIT_NUM	D_BULK_NUM / _DSAD_VALUE_BUF_SIZE	BULK 分割数
D_BULK_SPLIT_MOD	D_BULK_NUM %D_DSAD_VALUE_BUF_SIZE	BULK 分割余り
D_NO_ERROR	0x00U	エラー無し
D_OVER_FLOW_ERROR	0x02U	オーバフローエラー
D_DISCONNECTION_ERROR	0x80U	断線エラー
D_ON_OFF_COMMAND	@0¥r¥n	ON/OFF コマンド
D_COMMAND_SIZE	4U	PC 通信用受信コマンドサイズ

(b) r_rl78_i1e_common.h

表 4-8 マクロ宣言一覧

マクロ宣言	値	内容
D_MCU_VOLTAGE_MODE	0U	フラッシュ・メモリ電圧モード (フルスピード・モード)[固定]
D_PGA_DSAD_MPX0	0x00U	入力マルチプレクサ 0
D_PGA_DSAD_MPX1	0x01U	入力マルチプレクサ 1
D_PGA_DSAD_MPX2	0x02U	入力マルチプレクサ 2
D_PGA_DSAD_MPX3	0x03U	入力マルチプレクサ 3
D_DSAD_AUTO_GAIN_TRIGGER_COUNT	D_DSAD_AUTO_GAIN_TRIGGER_SEC * D_MCU_CLOCK_MHZ * 1000000	PGA 自動ゲイン調整タイミング カウンタ値
D_GAIN_ERROR_REFERENCE_LSB	(int32_t)((0x800000L * D_GAIN_ERROR_REFERENCE_mV / 0.8 / 1000L)	PGA ゲイン誤差測定基準電圧 (単位:LSB)
D_DISCONNECTION_THRESHOLD_LSB	(int32_t)(0x800000L * D_DISCONNECTION_THRESHOLD_mV / 0.8 / 1000L)	断線チェック判定電圧(単位:LSB)
D_UART_SEND_BUFFER_NUMBER	2U	PC 通信用送信バッファ本数
D_FLASH_SUCCESS	0U	フラッシュ・メモリ処理成功
D_FLASH_MATCH	0U	フラッシュ・メモリと RAM に差異なし
D_FLASH_MISMATCH	1U	フラッシュ・メモリと RAM に差異あり
D_FLASH_INVALID	2U	フラッシュ・メモリデータ無効
D_FLASH_FAILURE	6U	フラッシュ・メモリ処理エラー
D_FLASH_DATA_STRUCT_SIZE	sizeof(str_flash_data_t) / sizeof(uint8_t)	フラッシュ格納変数用構造体全体サイ ズ (単位:byte)
D_FLASH_DATA_CHECKSUM_SIZE	2U	チェックサムサイズ (単位:byte)
D_CAMP_CH_NUMBER	3U	コンフィギュラブル・アンプのチャン ネル数

(c) r_rl78_i1e_common.c

表 4-9 マクロ宣言一覧

マクロ宣言	値	内容
D_DSAD_VALEU_MAX	(1L<<23) - 1	$\Delta \Sigma$ A/D コンバータへの入力電圧 最大値 : +800 mV/(G _{TOTAL})
D_DSAD_VALEU_MIN	-1*(1L<<23)	$\Delta \Sigma$ A/D コンバータへの入力電圧 最小値 : -800 mV/(G _{TOTAL})
D_DSAD_AUTO_GAIN_MAX_POS	0U	PGA 自動ゲイン調整用配列添え字 最大値添え字
D_DSAD_AUTO_GAIN_MIN_POS	1U	PGA 自動ゲイン調整用配列添え字 最小値添え字

(d) r_rl78_common_util.h

表 4-10 マクロ宣言一覧

マクロ宣言	値	内容
D_MCU_FREQUENCY_HZ	(D_MCU_CLOCK_MHZ * 1000000U)	フリーランタイム用定義 MCU 動作クロック周波数 (単位:Hz)
D_WAIT10S	(D_MCU_FREQUENCY_HZ * 10U - 1U)	フリーランタイム用カウンタ値:10s
D_WAIT3S	(D_MCU_FREQUENCY_HZ * 3U - 1U)	フリーランタイム用カウンタ値:3s
D_WAIT1S	(D_MCU_FREQUENCY_HZ * 1U - 1U)	フリーランタイム用カウンタ値:1s
D_WAIT500MS	(D_MCU_FREQUENCY_HZ / 2U - 1U)	フリーランタイム用カウンタ値:500ms
D_WAIT100MS	(D_MCU_FREQUENCY_HZ / 10U - 1U)	フリーランタイム用カウンタ値:100ms
D_WAIT10MS	(D_MCU_FREQUENCY_HZ / 100U - 1U)	フリーランタイム用カウンタ値:10ms
D_WAIT5MS	(D_MCU_FREQUENCY_HZ / 200U - 1U)	フリーランタイム用カウンタ値:5ms
D_WAIT1MS	(D_MCU_FREQUENCY_HZ / 1000U - 1U)	フリーランタイム用カウンタ値:1ms
D_WAIT100US	(D_MCU_FREQUENCY_HZ / 10000U - 1U)	フリーランタイム用カウンタ値:100us
D_WAIT10US	(D_MCU_FREQUENCY_HZ / 100000U - 1U)	フリーランタイム用カウンタ値:10us

(e) r_keyscan.h

表 4-11 マクロ宣言一覧

マクロ宣言	値	内容
DEF_KEY_ACTIVE_LOW	0U	ポート有効レベル=LOW
DEF_KEY_ACTIVE_HIGH	1U	ポート有効レベル=HIGH

4.4 型宣言

本項では API で定義されている独自型について示します。

4.4.1 ユーザー環境依存設定用列挙体

表 4-12 に示す列挙体は、「4.5 ユーザー環境依存設定グローバル定数」で記載されている定数と共に用いられ、その設定に従って変更する必要があります。詳細は「4.5 ユーザー環境依存設定グローバル定数」を参照ください。

(a) r_keyscan.h

表 4-12 キー設定情報格納グローバル変数指定用列挙体

型名	デフォルト定義	内容
e_key_t	SW_MODE_CHANGE	入力ポート番号などのキー設定情報が格納されているグローバル変数構造体配列の添え字を指し示す列挙体です。キー情報取得関数の呼び出し時に使用します。 ユーザーがキーを追加する場合は、ユニークな ENUM 値を追加してください。 (例) SW_MODE_CHANGE = 0x00U, SW_GAIN_UP, // 以降、追加キー SW_GAIN_DOWN, SW_OFFSET_UP, SW_OFFSET_DOWN,
	SW_TYPE_MAX	本定義によりキー設定情報が格納されているグローバル変数構造体配列のサイズを決定していますので、名称の変更・削除・列挙体の最後尾からの移動はしないでください。

注：変更時は必ず上記列挙体の設定とグローバル変数構造体配列の要素を一致させてください。

4.4.2 列挙体

本節では API で定義されている列挙体宣言について示します。

(a) r_cg_main.c

表 4-13 BULK フォーマットデータ生成制御用列挙体

型名	マクロ名	内容
e_bulk_control_t	E_BULKDATA_BULKSTART	BULKSTART フォーマットデータ生成
	E_BULKDATA_BULK	BULK フォーマットデータ生成
	E_BULKDATA_BULKEND	BULKEND フォーマットデータ生成

表 4-14 通信コマンド生成制御用列挙体

型名	マクロ名	内容
e_communication_data_t	E_STREAMHEADER	STREAMHEADER フォーマットデータ生成
	E_STREAM	STREAM フォーマットデータ生成
	E_BULK	BULK フォーマットデータ生成
	E_BINARY	BINARY フォーマットデータ生成

表 4-15 通信コマンド判定用列举体

型名	マクロ名	内容
e_communication_data_t	E_ON_OFF_COMMAND	ON/OFF コマンド
	E_COMMAND_NONE	コマンド無し判定用

表 4-16 測定制御用列举体

型名	マクロ名	内容
e_measurement_control_t	E_TRANSMISSION_START	シリアル送信開始
	E_DATA_GENERATION	測定値送信データ生成
	E_ERROR_CHECK	ERROR チェック
	E_WAIT	動作無し

表 4-17 MCU 状態判定用列举体

型名	マクロ名	内容
e_mcu_state_t	E_MCU_HALT	HALT モード
	E_MCU_RUN,	動作中

(b) r_rl78_i1e_common.h

表 4-18 入力マルチプレクサ指定用列举体

型名	マクロ名	値	内容
e_pga_dsad_mpx_t	E_PGA_DSAD_MPX0	0x00U	入力マルチプレクサ 0
	E_PGA_DSAD_MPX1	0x01U	入力マルチプレクサ 1
	E_PGA_DSAD_MPX2	0x02U	入力マルチプレクサ 2
	E_PGA_DSAD_MPX3	0x03U	入力マルチプレクサ 3
	E_PGA_DSAD_MPX4	0x04U	入力マルチプレクサ 4 -> 内蔵温度センサ固定
	E_PGA_DSAD_MPX_MAX	0x05U	入力マルチプレクサ最大判定

表 4-19 入力モード指定用列举体

型名	マクロ名	値	内容
e_pga_dsad_input_mode_t	E_PGA_DSAD_DIFF_INPUT	0x00U	差動入力モード
	E_PGA_DSAD_SINGLE_INPUT	0x01U	シングル・エンド入力モード

表 4-20 $\Delta \Sigma$ A/D 変換許可/停止指定用列举体

型名	マクロ名	値	内容
e_pga_dsad_onoff_t	E_PGA_DSAD_OFF	0x01U	A/D 変換を停止
	E_PGA_DSAD_ON	0x00U	A/D 変換を許可

表 4-21 PGA ゲイン設定指定用列挙体

型名	マクロ名	値	内容
e_pga_dsad_gain_t	E_PGA_DSAD_GAIN_1_1_1	0x00U	$G_{SET1} = x1, G_{SET2} = x1, G_{TOTAL} = x1$
	E_PGA_DSAD_GAIN_2_1_2	0x04U	$G_{SET1} = x2, G_{SET2} = x1, G_{TOTAL} = x2$
	E_PGA_DSAD_GAIN_3_1_3	0x08U	$G_{SET1} = x3, G_{SET2} = x1, G_{TOTAL} = x3$
	E_PGA_DSAD_GAIN_4_1_4	0x0CU	$G_{SET1} = x4, G_{SET2} = x1, G_{TOTAL} = x4$
	E_PGA_DSAD_GAIN_8_1_8	0x10U	$G_{SET1} = x8, G_{SET2} = x1, G_{TOTAL} = x8$
	E_PGA_DSAD_GAIN_1_2_2	0x01U	$G_{SET1} = x1, G_{SET2} = x2, G_{TOTAL} = x2$
	E_PGA_DSAD_GAIN_2_2_4	0x05U	$G_{SET1} = x2, G_{SET2} = x2, G_{TOTAL} = x4$
	E_PGA_DSAD_GAIN_3_2_6	0x09U	$G_{SET1} = x3, G_{SET2} = x2, G_{TOTAL} = x6$
	E_PGA_DSAD_GAIN_4_2_8	0x0DU	$G_{SET1} = x4, G_{SET2} = x2, G_{TOTAL} = x8$
	E_PGA_DSAD_GAIN_8_2_16	0x11U	$G_{SET1} = x8, G_{SET2} = x2, G_{TOTAL} = x16$
	E_PGA_DSAD_GAIN_1_4_4	0x02U	$G_{SET1} = x1, G_{SET2} = x4, G_{TOTAL} = x4$
	E_PGA_DSAD_GAIN_2_4_8	0x06U	$G_{SET1} = x2, G_{SET2} = x4, G_{TOTAL} = x8$
	E_PGA_DSAD_GAIN_3_4_12	0x0AU	$G_{SET1} = x3, G_{SET2} = x4, G_{TOTAL} = x12$
	E_PGA_DSAD_GAIN_4_4_16	0x0EU	$G_{SET1} = x4, G_{SET2} = x4, G_{TOTAL} = x16$
	E_PGA_DSAD_GAIN_8_4_32	0x12U	$G_{SET1} = x8, G_{SET2} = x4, G_{TOTAL} = x32$
	E_PGA_DSAD_GAIN_1_8_8	0x03U	$G_{SET1} = x1, G_{SET2} = x8, G_{TOTAL} = x8$
	E_PGA_DSAD_GAIN_2_8_16	0x07U	$G_{SET1} = x2, G_{SET2} = x8, G_{TOTAL} = x16$
	E_PGA_DSAD_GAIN_3_8_24	0x0BU	$G_{SET1} = x3, G_{SET2} = x8, G_{TOTAL} = x24$
	E_PGA_DSAD_GAIN_4_8_32	0x0FU	$G_{SET1} = x4, G_{SET2} = x8, G_{TOTAL} = x32$
	E_PGA_DSAD_GAIN_8_8_64	0x13U	$G_{SET1} = x8, G_{SET2} = x8, G_{TOTAL} = x64$

表 4-22 オフセット電圧設定指定用列举体

型名	マクロ名	値	内容
e_pga_dsad_offset_t	E_PGA_DSAD_OFFSET_164p06	0x1FU	164.06/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_153p13	0x1EU	153.13/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_142p19	0x1DU	142.19/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_131p25	0x1CU	131.25/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_120p31	0x1BU	120.31/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_109p38	0x1AU	109.38/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_98p44	0x19U	98.44/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_87p50	0x18U	87.50/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_76p56	0x17U	76.56/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_65p63	0x16U	65.63/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_54p69	0x15U	54.69/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_43p75	0x14U	43.75/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_32p81	0x13U	32.81/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_21p88	0x12U	21.88/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_10p94	0x11U	10.94/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_0p00	0x10U	0.00/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M10p94	0x0FU	-10.94/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M21p88	0x0EU	-21.88/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M32p81	0x0DU	-32.81/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M43p75	0x0CU	-43.75/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M54p69	0x0BU	-54.69/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M65p63	0x0AU	-65.63/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M76p56	0x09U	-76.56/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M87p50	0x08U	-87.50/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M98p44	0x07U	-98.44/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M109p38	0x06U	-109.38/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M120p31	0x05U	-120.31/G _{SET1} [mV]
	E_PGA_DSAD_OFFSET_M131p25	0x04U	-131.25/G _{SET1} [mV]
E_PGA_DSAD_OFFSET_M142p19	0x03U	-142.19/G _{SET1} [mV]	
E_PGA_DSAD_OFFSET_M153p13	0x02U	-153.13/G _{SET1} [mV]	
E_PGA_DSAD_OFFSET_M164p06	0x01U	-164.06/G _{SET1} [mV]	

表 4-23 OSR 設定指定用列举体

型名	マクロ名	値	内容
e_pga_dsad_osr_t	E_PGA_DSAD_OSR_64	0x00U	15625.000 [sps]
	E_PGA_DSAD_OSR_128	0x01U	7812.500 [sps]
	E_PGA_DSAD_OSR_256	0x02U	3906.250 [sps]
	E_PGA_DSAD_OSR_512	0x03U	1953.125 [sps]
	E_PGA_DSAD_OSR_1024	0x04U	976.563 [sps]
	E_PGA_DSAD_OSR_2048	0x05U	488.281 [sps]

表 4-24 $\Delta \Sigma$ A/D 変換回数算出方法の指定用列举体

型名	マクロ名	値	内容
e_pga_dsad_count_mode_t	E_PGA_DSAD_COUNT_CALCULATION	0x00U	PGAxCTL2 レジスタの設定値で 1~8032 回を指定
	E_PGA_DSAD_COUNT_LINEAR	0x01U	PGAxCTL2 レジスタの設定値で 1~255 回をリニアに指定

表 4-25 平均化処理の動作選択指定用列挙体

型名	マクロ名	値	内容
e_pga_dsad_average_operation_t	E_PGA_DSAD_DO_NOT_AVERAGE_1	0x00U	平均化処理を行わない
	E_PGA_DSAD_DO_NOT_AVERAGE_2	0x01U	平均化処理を行わない
	E_PGA_DSAD_AVERAGE_INT_AN_ADC	0x02U	平均化処理を行い、INTDSAD を 1 回の A/D 変換ごとに発生する
	E_PGA_DSAD_AVERAGE_INT_UPDATE	0x03U	平均化処理を行い、INTDSAD を平均値更新ごとに発生する

表 4-26 平均するデータ数の選択指定用列挙体

型名	マクロ名	値	内容
e_pga_dsad_average_number_t	E_PGA_DSAD_AVERAGE_8	0x00U	平均するデータ数 8
	E_PGA_DSAD_AVERAGE_16	0x01U	平均するデータ数 16
	E_PGA_DSAD_AVERAGE_32	0x02U	平均するデータ数 32
	E_PGA_DSAD_AVERAGE_64	0x03U	平均するデータ数 64

表 4-27 $\Delta \Sigma$ A/D 変換(AUTOSCAN)の開始/停止指定用列挙体

型名	マクロ名	値	内容
e_pga_dsad_start_stop_t	E_PGA_DSAD_STOP	0x00U	停止
	E_PGA_DSAD_START	0x01U	開始

表 4-28 オートスキャンのモード指定用列挙体

型名	マクロ名	値	内容
e_pga_dsad_autoscan_mode_t	E_PGA_DSAD_CONTINUOUS	0x00U	連続スキャンモード
	E_PGA_DSAD_SINGLE	0x01U	シングル・スキャンモード

表 4-29 PGA 自動ゲイン調整処理戻り値指定用列挙体

型名	マクロ名	値	内容
e_auto_gain_return_t	E_AUTO_GAIN_NO_ADJUSTMENT	0x00U	PGA 自動ゲイン調整なし
	E_AUTO_GAIN_ADJUSTMENT	0x01U	PGA 自動ゲイン調整実施

表 4-30 断線検知処理戻り値指定用列挙体

型名	マクロ名	値	内容
e_disconnection_return_t	E_DISCONNECTION_NO_DETECT	0x00U	非断線状態
	E_DISCONNECTION_DETECT	0x01U	断線状態

(c) r_keyscan.h

表 4-31 キー状態指定用列挙体

型名	マクロ名	内容
e_key_status_t	E_KEY_OFF	キーオフ状態
	E_KEY_OFF_TO_ON	キーオフからキー短押し状態に遷移
	E_KEY_ON_NORM	キー短押し状態
	E_KEY_NORM_TO_LONG	キー短押し状態からキー長押し状態に遷移
	E_KEY_ON_LONG	キー長押し状態
	E_KEY_LONG_TO_LONG	キー長押し状態からキー長押し状態に遷移
	E_KEY_DOUBLE_CLICK	キーダブルクリック状態

4.4.3 構造体

本項では API で定義されている構造体宣言について示します。

(a) r_cg_main.c

表 4-32 測定データ格納変数用構造体

構造体型名	measurement_data_t		
概要	測定データ格納変数用の定義。		
メンバ変数	型	名称	内容
	uint32_t	count	$\Delta \Sigma$ A/D 変換回数カウンタ
	str_pga_dsad_value_t	dsad_value	$\Delta \Sigma$ A/D 変換値構造体変数
	e_mcu_state_t	mcu_state	MCU の動作状態
uint8_t	error_state	エラー状態	

表 4-33 BULK 転送データ格納変数用構造体

構造体型名	bulk_data_t		
概要	BULK 転送データ格納変数用の定義。		
メンバ変数	型	名称	内容
	uint16_t	send_count	BULK 転送送信回数カウンタ
	e_bulk_control_t	control	BULK 転送制御
	int32_t	buf[D_BULK_BUFFER_SIZE]	BULK 転送データ用バッファ
	uint16_t	str_count	BULK 転送データ用バッファカウンタ
uint16_t	split_count	BULK 転送分割送信回数カウンタ	

(b) r_rl78_i1e_common.h

表 4-34 $\Delta \Sigma$ A/D 変換情報格納変数用構造体

構造体型名	str_pga_dsad_setting_t		
概要	PGA、 $\Delta \Sigma$ A/D の各制御レジスタ設定情報格納変数用の定義。		
メンバ変数	型	名称	内容
	e_pga_dsad_onoff_t	dsad_onoff	A/D 変換 有効/無効フラグ
	e_pga_dsad_input_mode_t	dsad_input_mode	入力モード
	e_pga_dsad_offset_t	dsad_offset	オフセットレジスタ設定値
	e_pga_dsad_osr_t	dsad_osr	オーバー・サンプリング・レート
	e_pga_dsad_gain_t	dsad_gain	ゲイン
	uint8_t	dsad_count	オートスキャン回数カウンタ
	e_pga_dsad_count_mode_t	dsad_count_mode	A/D 変換回数算出方法指定
	e_pga_dsad_average_operation_t	dsad_average_operation	平均化処理の動作選択
e_pga_dsad_average_number_t	dsad_average_number	平均するデータ数の選択	

表 4-35 $\Delta \Sigma$ A/D 変換値格納バッファ用構造体

構造体型名	str_pga_dsad_value_t		
概要	$\Delta \Sigma$ A/D 変換値と、ゲイン、オフセット等、各種情報格納変数用の定義。		
メンバ変数	型	名称	内容
	uint32_t	count	測定番号カウンタ値
	uni_long_t	adc_value	A/D 変換値(右詰め 24bit)
	uint8_t	ch	Ch 番号
	uint8_t	overflow	オーバーフローフラグ
	uint8_t	gain_set_1	前段ゲイン倍率
	uint8_t	gain_set_2	後段ゲイン倍率
	uint8_t	gain_total	トータルゲイン倍率
	e_pga_dsad_offset_t	offset_reg	オフセットレジスタ設定値
int32_t	adc_correct	A/D 変換補正值(右詰め 24bit)	

表 4-36 フラッシュ格納変数用構造体

構造体型名	str_flash_data_t		
概要	フラッシュ格納変数用の定義。		
メンバ変数	型	名称	内容
	int32_t	gain_real[5U]	実ゲイン設定値 $G_{SET1} \times \text{固定 } G_{SET2} = G_{SET1} [x1x2x3x4x8] \times G_{SET2} \times 100(\text{固定小数})$
	int32_t	offset_factor[5U][32U]	PGA オフセット誤差補正值 オフセット 31 段階 \times 前段ゲイン 5 段階 = 155 個保持 [m][n] m はゲイン設定、n はオフセット設定値
	uint16_t	checksum	チェックサム値(16bit)

(c) r_keyscan.h

表 4-37 キー情報設定変数用構造体

構造体型名	key_setting_t		
概要	キー情報設定変数用の定義。		
メンバ変数	型	名称	内容
	uint8_t *	p_port_addr	SW 端子に接続しているポートレジスタのアドレス
	uint8_t	port_bit_num	SW 端子に接続しているポートレジスタのビット番号
	uint8_t	multiple_group	同時押し許可判別用グループ番号/同番号の同時押しを認めない

4.4.4 共用体

本項では API で定義されている共用体宣言について示します。

(a) r_rl78_i1e_common.h

表 4-38 signed long/unsigned long 読み出し用共用体

構造体型名	uni_long_t			
概要	signed/unsigned 読み出し用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	int32_t	32	LONG	signed long
	uint32_t	32	uLONG	unsigned long

表 4-39 $\Delta \Sigma$ A/D コンバータ変換結果レジスタ C 読み出し用共用体

構造体型名	uni_pga_dsad_conversion_result_t			
概要	$\Delta \Sigma$ A/D コンバータ変換結果レジスタ C 読み出し用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	BIT	4	-	未使用
		1	overflow	Overflow Flag
		3	ch	Ch number

表 4-40 $\Delta \Sigma$ A/D コンバータ変換結果レジスタ読み出し用共用体

構造体型名	uni_pga_dsad_value_t				
概要	$\Delta \Sigma$ A/D コンバータ変換結果レジスタ読み出し用共用体定義。				
メンバ変数	型	ビット数	名称	内容	
	uint32_t	32	LONG	32bit access	
	uint16_t	32	WORD[2]	16bit access	
	BYTE	uni_pga_dsad_conversion_result_t	8	dsad_c	$\Delta \Sigma$ A/D コンバータ変換結果レジスタ C
		uint8_t	8	dsad_l	$\Delta \Sigma$ A/D コンバータ変換結果レジスタ L
		uint8_t	8	dsad_m	$\Delta \Sigma$ A/D コンバータ変換結果レジスタ M
		uint8_t	8	dsad_h	$\Delta \Sigma$ A/D コンバータ変換結果レジスタ H

表 4-41 PGAxCTL0 レジスタ、ゲイン設定用共用体

構造体型名	uni_pga_dsad_gain_t			
概要	PGAxCTL0 レジスタ、ゲイン設定読み出し用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	GAIN_BIT	2	gain_set_2	G _{SET2}
		3	gain_set_1	G _{SET1}
		3	-	未使用

表 4-42 PGAxCTL0 レジスタ制御用共用体

構造体名	uni_reg_pgaxctl0_t			
概要	PGAxCTL0 レジスタ制御用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	PGAxCTL0_BIT	5	gain	GAIN
		3	osr	OSR

表 4-43 PGAxCTL1 レジスタ制御用共用体

構造体名	uni_reg_pgaxctl1_t			
概要	PGAxCTL1 レジスタ制御用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	PGAxCTL1_BIT	5	offset	offset
		1	-	未使用
		1	pga3tsel	コンフィギュラブル・アンプ自己補正
1		input_mode	AINSEL	

表 4-44 PGAxCTL2 レジスタ制御用共用体

構造体名	uni_reg_pgaxctl2_t			
概要	PGAxCTL2 レジスタ制御用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	PGAxCTL2_BIT	5	low	offset
		3	high	未使用

表 4-45 PGAxCTL3 レジスタ制御用共用体

構造体名	uni_reg_pgaxctl3_t			
概要	PGAxCTL3 レジスタ制御用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	PGAxCTL3_BIT	2	pgaxave0_1	PGAXAV[1:0]平均化するデータの選択
		2	pgaxave3_2	PGAXAV[3:2]平均化処理の動作選択
		3	-	未使用
1		pgaxctm	A/D 変換回数指定モード選択	

表 4-46 DSADCTL レジスタ制御用共用体

構造体名	uni_reg_pgaxctl3_t			
概要	DSADCTL レジスタ制御用共用体定義。			
メンバ変数	型	ビット数	名称	内容
	uint8_t	8	BYTE	8bit access
	DSADCTL_BIT	5	dsadbmp4_0	入力マルチプレクサ n からの信号(n=4-0)
		1	dsadscm	オートスキャンモードの選択
		1	-	未使用
1		dsadtst	A/D 変換(AUTOSCAN)の制御	

4.5 ユーザー環境依存設定グローバル定数

4.5.1 キー設定情報格納グローバル変数

本項で示す `r_keyscan.h` ファイルにある定数を変更することにより、ユーザーが使用するハードウェア構成 (I/O ポート) に対応させることができます。

(a) `r_keyscan.h`

表 4-47 キー設定情報格納グローバル変数用構造体

構造体型名	key_setting_t		
概要	キー設定情報格納グローバル変数用構造体		
メンバ変数	型	名称	内容
	uint8_t*	p_port_addr	SW に接続しているポートレジスタのアドレス
	uint8_t	port_bit_num	SW に接続しているポートレジスタのビット番号
	uint8_t	multiple_group	同時押し許可判別用グループ番号 (同番号の同時押しは無効)

(b) 記述例

- ケース 1

RL78/I1E の P13.7 を「SW_MODE_CHANGE」としてキー登録を行う。(本サンプルコードの既定値)

```
typedef enum
{
    SW_MODE_CHANGE = 0x00U,

    SW_TYPE_MAX
} e_key_t;

const key_setting_t g_con_key_setting[SW_TYPE_MAX] =
{
    /*          ポート ビット番号 グループ */
    /* SW_MODE_CHANGE */ {&P13, 7U, 0U},
};
```

- ケース 2

RL78/I1E の P12.1 を「SW_GAIN_UP」、P12.0 を「SW_GAIN_DOWN」としてキー登録を行い、グループ 1 として同一グループ設定（同時押し無効）とする。また、P1.2 を「SW_OFFSET_UP」、P1.5 を「SW_OFFSET_DOWN」としてキー登録を行い、グループ 2 として同一グループ設定（同時押し無効）とする。

```
typedef enum
{
    SW_GAIN_UP = 0x00U,
    SW_GAIN_DOWN,
    SW_OFFSET_UP,
    SW_OFFSET_DOWN,

    SW_TYPE_MAX
} e_key_t;

const key_setting_t g_con_key_setting[SW_TYPE_MAX] =
{
    /*          ポート ビット番号 グループ */
    /* SW_GAIN_UP      */ /* {&P12, 1U,      1U},
    /* SW_GAIN_DOWN    */ /* {&P12, 0U,      1U},
    /* SW_OFFSET_UP    */ /* {&P1,  2U,      2U},
    /* SW_OFFSET_DOWN  */ /* {&P1,  5U,      2U},
};
```

4.6 グローバル変数

本節では API で定義されているグローバル変数について示します。

(a) r_cg_main.c

表 4-48 r_cg_main.c グローバル変数一覧

型名	値	内容
bulk_data_t	g_bulk_data	BULK 転送データ格納変数

(b) r_rl78_i1e_common.c

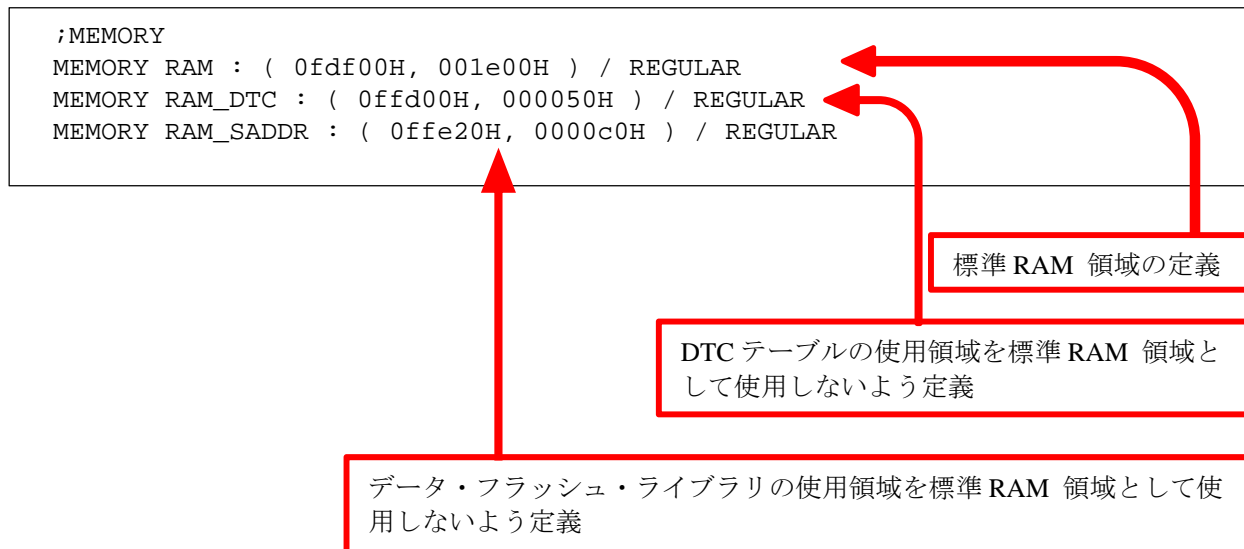
表 4-49 r_rl78_i1e_common.c グローバル変数一覧

変数名	型名	内容
g_flash_value	str_flash_data_t	データ・フラッシュ格納データ構造体
g_dsad_autoscan_mode	e_pga_dsad_autoscan_mode_t	オートスキャンモード設定
g_dsad_setting	str_pga_dsad_setting_t	各入力マルチプレクサ設定保持変数
g_camp_trimming	uint8_t	コンフィギュラブル・アンプトリミング設定値格納変数
g_dsad_value	str_pga_dsad_value_t	DSAD 変換値格納バッファ
g_dsad_value_write_pos	uint16_t	DSAD 変換値格納バッファ書き込み位置
g_dsad_value_read_pos	uint16_t	DSAD 変換値格納バッファ読み出し位置
g_dsad_value_user_read_pos	uint16_t	DSAD 変換値格納バッファユーザ読み出し位置
g_dsad_measurement_count	uint32_t	DSAD 変換測定番号カウンタ
g_i1e_uart_send_buffer	int8_t	PC 送信用の送信バッファ(UART への転送は DTC 使用)
g_buffer_number	uint8_t	PC 送信用の送信バッファの格納先番号

4.7 リンク・ディレクティブ・ファイル

リンク・ディレクティブ・ファイルによって、データ・フラッシュ・ライブラリおよび DTC テーブルで使用する RAM 領域を使用しないように設定を行います。

本サンプルコードで使用するリンク・ディレクティブ・ファイル「SampleCode.dr」の概要を以下に記載します。

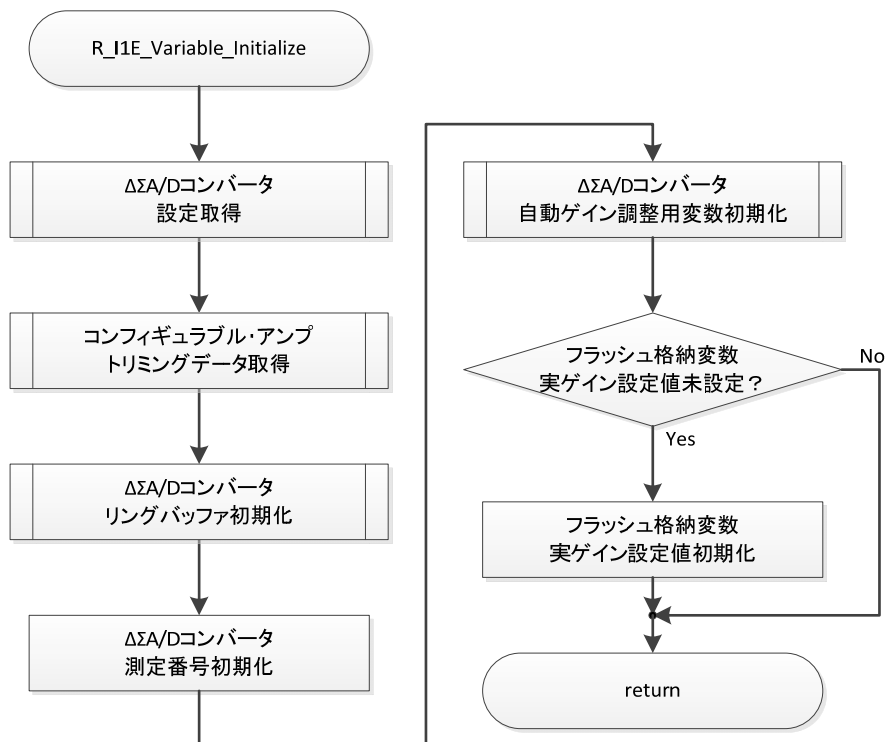


5. アナログ・フロントエンド関連 API

5.1 サンプルコード使用変数初期化処理

void R_I1E_Variable_Initialize(void)

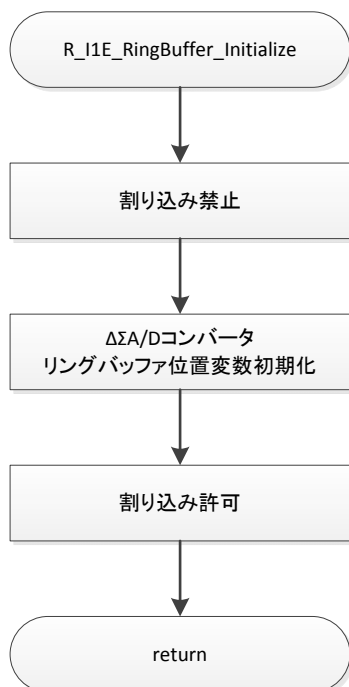
説明	サンプルコード使用変数初期化処理
引数	なし
グローバル変数	g_dsad_autoscan_mode: オートスキャンモード設定 g_dsad_setting[]: 各入力マルチプレクサ設定保持変数 g_camp_trimming: コンフィギュラブル・アンプ n トリミング設定値 g_flash_value: データ・フラッシュ格納データ構造体 gs_con_gain_set_1_size: 前段ゲインの倍率要素数 gs_con_gain_set_1[]: 前段ゲインの倍率
SFR	なし
戻り値	なし
処理内容	<p>ΔΣA/D コンバータの各種設定を取得する。</p> <p>コンフィギュラブル・アンプのトリミングデータを取得する。</p> <p>ΔΣA/D コンバータ変換値リングバッファを初期化する。</p> <p>ΔΣA/D コンバータ変換測定番号カウンタを初期化する。(0 クリア)</p> <p>PGA 自動ゲイン調整用変数を初期化する。</p> <p>フラッシュ格納用構造体の実ゲイン設定値配列が設定されていない場合、初期化する。</p>



5.2 DSAD リングバッファ初期化関数

```
void R_I1E_RingBuffer_Initialize(void)
```

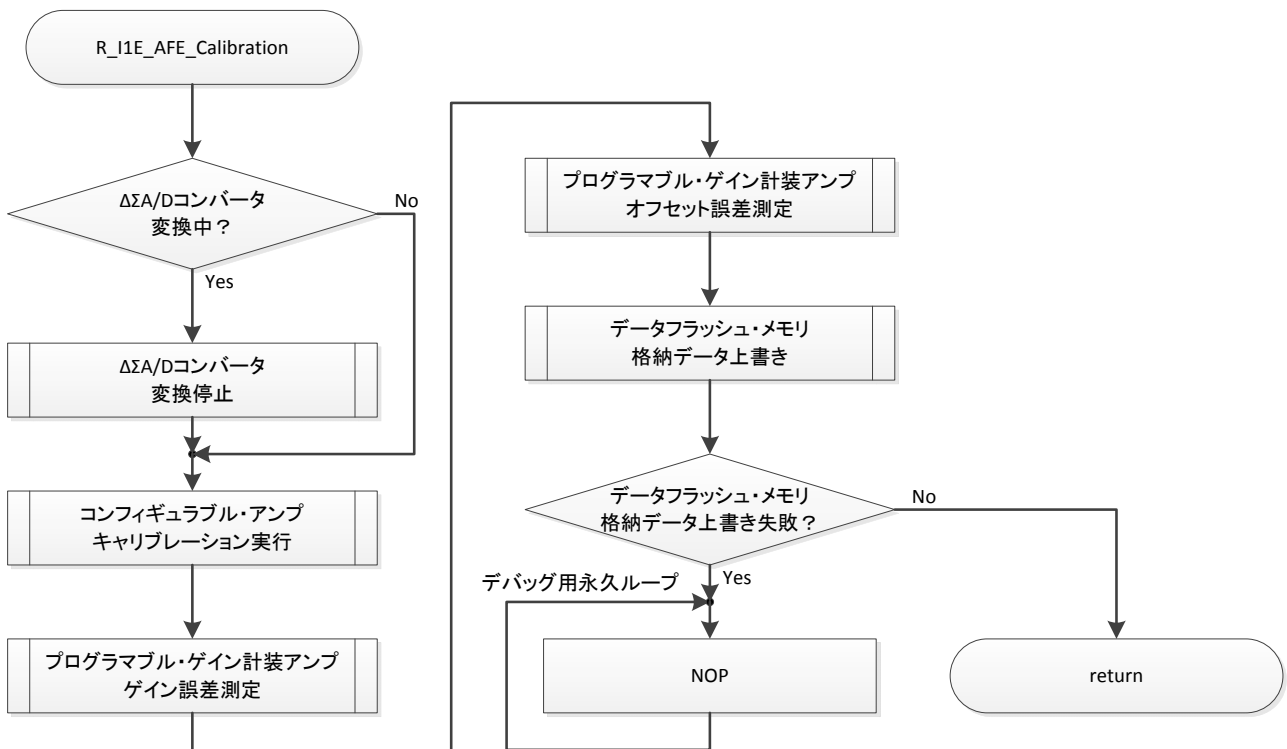
説明	DSAD リングバッファ初期化関数
引数	なし
グローバル変数	g_dsad_value_write_pos: DSAD 変換値格納バッファ書き込み位置 g_dsad_value_read_pos: DSAD 変換値格納バッファ読み出し位置 g_dsad_value_user_read_pos: DSAD 変換値格納バッファユーザ読み出し位置
SFR	なし
戻り値	なし
処理内容	$\Delta\Sigma/A/D$ コンバータ変換値格納リングバッファの書き込み位置/読み出し位置記憶変数を初期化する。



5.3 AFE モジュール校正実行関数

void R_I1E_AFE_Calibration(void)

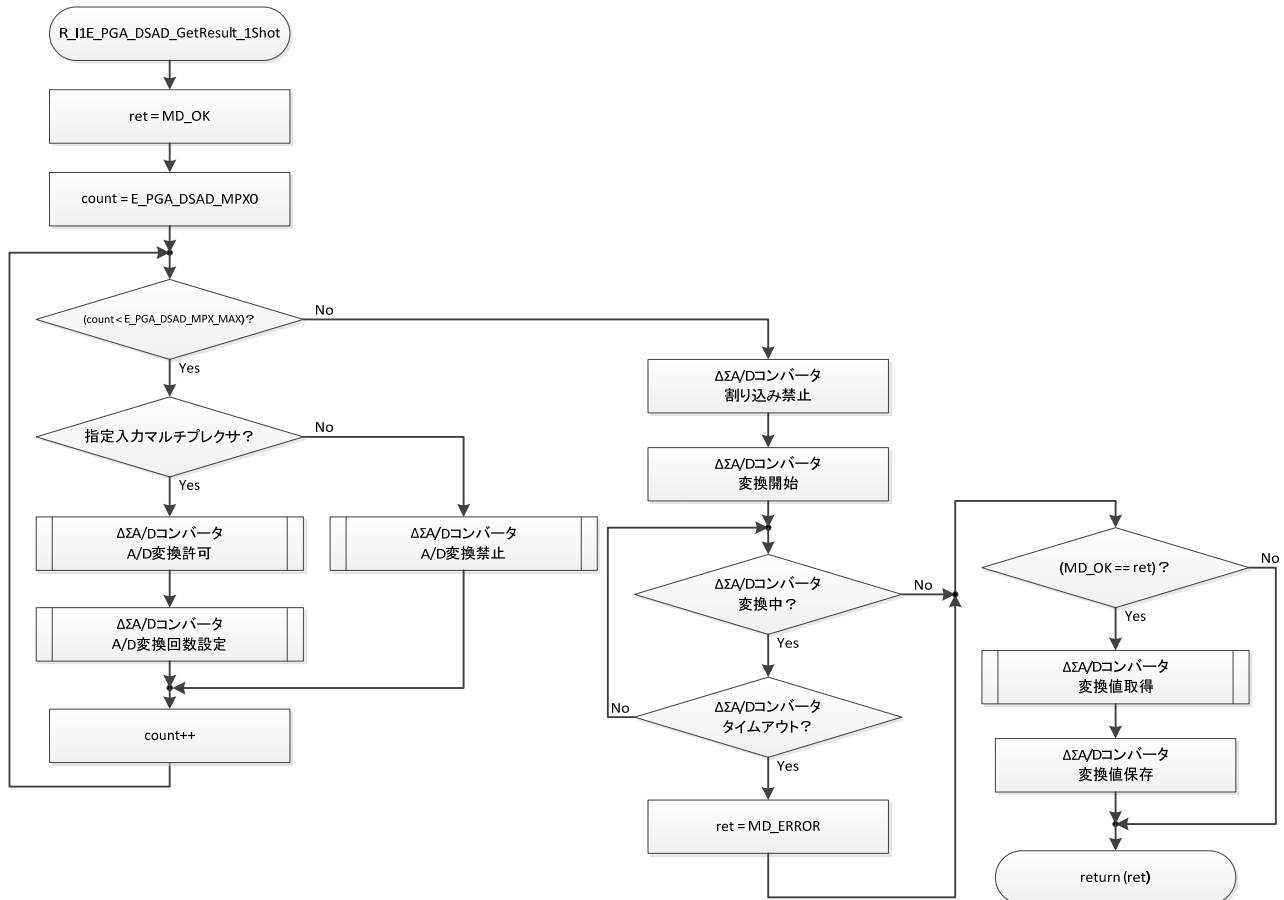
説明	AFE モジュール校正実行関数
引数	なし
グローバル変数	g_camp_trimming: コンフィギュラブル・アンプ n トリミング設定値 g_flash_value: データ・フラッシュ格納データ構造体
SFR	なし
戻り値	なし
処理内容	コンフィギュラブル・アンプのオフセット・キャリブレーションを実行し、結果を g_camp_trimming に格納する。 プログラマブル・ゲイン計装アンプのゲイン誤差を測定し、フラッシュ格納用構造体の実ゲイン設定値配列に格納する。 プログラマブル・ゲイン計装アンプのオフセット誤差を測定し、フラッシュ格納用構造体のオフセット誤差補正値配列に格納する。 フラッシュ・メモリに取得したデータを書き込みする。



5.4 24ビット $\Delta\Sigma$ A/Dコンバータ・指定ch1ショット差動入力A/D変換実行関数

MD_STATUS R_I1E_PGA_DSAD_GetResult_1Shot
 (e_pga_dsad_mpx_t dsad_mpx, int32_t *p_dsad_value)

説明	24ビット $\Delta\Sigma$ A/Dコンバータ・指定ch1ショット差動入力A/D変換実行関数
引数	dsad_mpx: 24ビット $\Delta\Sigma$ A/Dコンバータ 入力マルチプレクサ番号 *p_dsad_value: 24ビット $\Delta\Sigma$ A/Dコンバータ 24ビットA/D値 格納ポインタ
グローバル変数	なし
SFR	DSADIF: $\Delta\Sigma$ A/D変換 変換完了割り込み要求フラグ DSADSIF: $\Delta\Sigma$ A/D変換 スキャン完了割り込み要求フラグ DSADMK: $\Delta\Sigma$ A/D変換 変換完了割り込みマスクフラグ DSADSMK: $\Delta\Sigma$ A/D変換 スキャン完了割り込みマスクフラグ DSADST: A/D変換 (AUTOSCAN) の制御
戻り値	MD_STATUS: MD_OK MD_ERROR
処理内容	$\Delta\Sigma$ A/Dコンバータで入力マルチプレクサ指定チャンネルからの入力によりA/D変換を行い、*p_dsad_valueに結果を格納する。 A/D変換がタイムアウトした場合はMD_ERRORを返す。



5.5 24ビット $\Delta\Sigma$ A/Dコンバータ・ゲイン設定関数

```
void R_I1E_PGA_DSAD_GainRegSet(e_pga_dsad_mpx_t dsad_mpx, e_pga_dsad_gain_t dsad_gain)
```

説明	24ビット $\Delta\Sigma$ A/Dコンバータ・ゲイン設定関数
引数	dsad_mpx: 24ビット $\Delta\Sigma$ A/Dコンバータ 入力マルチプレクサ番号 dsad_gain: 24ビット $\Delta\Sigma$ A/Dコンバータ ゲイン設定値
グローバル変数	なし
SFR	PGAxCTL0: 入力マルチプレクサx (x=0~4) 設定レジスタ 0
戻り値	なし
処理内容	$\Delta\Sigma$ A/Dコンバータ入力マルチプレクサの指定チャンネル用プログラマブル・ゲイン計装アンプゲインレジスタに指定値を設定する。

5.6 24ビット $\Delta\Sigma$ A/Dコンバータオフセット設定関数

```
void R_I1E_PGA_DSAD_OffsetRegSet  
(e_pga_dsad_mpx_t dsad_mpx, e_pga_dsad_offset_t dsad_offset)
```

説明	24ビット $\Delta\Sigma$ A/Dコンバータオフセット設定関数
引数	dsad_mpx: 24ビット $\Delta\Sigma$ A/Dコンバータ 入力マルチプレクサ番号 dsad_offset: 24ビット $\Delta\Sigma$ A/Dコンバータ オフセット設定値
グローバル変数	なし
SFR	PGAxCTL1: 入力マルチプレクサx (x=0~4) 設定レジスタ 1
戻り値	なし
処理内容	$\Delta\Sigma$ A/Dコンバータ入力マルチプレクサの指定チャンネル用オフセット電圧レジスタに指定値を設定する。

5.7 24ビット $\Delta\Sigma$ A/Dコンバータ・OSR設定関数

```
void R_I1E_PGA_DSAD_OsrRegSet(e_pga_dsad_mpx_t dsad_mpx, e_pga_dsad_osr_t dsad_osr)
```

説明	24ビット $\Delta\Sigma$ A/Dコンバータ・OSR設定関数
引数	dsad_mpx: 24ビット $\Delta\Sigma$ A/Dコンバータ 入力マルチプレクサ番号 osr: 24ビット $\Delta\Sigma$ A/Dコンバータ OSR設定値
グローバル変数	なし
SFR	PGAxCTL0: 入力マルチプレクサx (x=0~4) 設定レジスタ 0
戻り値	なし
処理内容	$\Delta\Sigma$ A/Dコンバータ入力マルチプレクサの指定チャンネル用オーバ・サンプリング比レジスタに指定値を設定する。

5.8 オートスキャンモード、各入力マルチプレクサ設定保持変数格納関数

```
void R_I1E_PGA_DSAD_SettingRegGet
(e_pga_dsad_autoscan_mode_t *dsad_autoscan, str_pga_dsad_setting_t dsad_setting[])
```

説明	オートスキャンモード、各入力マルチプレクサ設定保持変数格納関数
引数	*dsad_autoscan: オートスキャンモード設定 dsad_setting[]: 24ビットΔΣA/Dコンバータ設定値
グローバル変数	gs_gain_set_2: 後段ゲインの倍率
SFR	なし
戻り値	なし
処理内容	ΔΣA/Dコンバータのオートスキャンモード設定を取得し、*dsad_autoscanに格納する。 ΔΣA/Dコンバータ全入力マルチプレクサ用の各種設定を取得し、dsad_setting[]配列に格納する。 PGA誤差測定で使用する入力マルチプレクサ用のプログラマブル・ゲイン計装アンプの後段ゲイン倍率を取得し、グローバル変数gs_gain_set_2に格納する。

5.9 DSAD設定に構造体変数情報を設定

```
void R_I1E_PGA_DSAD_SettingRegSet
(e_pga_dsad_autoscan_mode_t dsad_autoscan, str_pga_dsad_setting_t dsad_setting[])
```

説明	DSAD設定に構造体変数情報を設定
引数	dsad_autoscan: オートスキャンモード設定 dsad_setting[]: 24ビットΔΣA/Dコンバータ設定値
グローバル変数	なし
SFR	なし
戻り値	なし
処理内容	ΔΣA/Dコンバータのオートスキャンモードを指定のモードに設定する。 ΔΣA/Dコンバータ全入力マルチプレクサ用の各種設定をdsad_setting[]配列で指定される値に設定する。

5.10 PGAオフセット誤差補正適用値取得関数

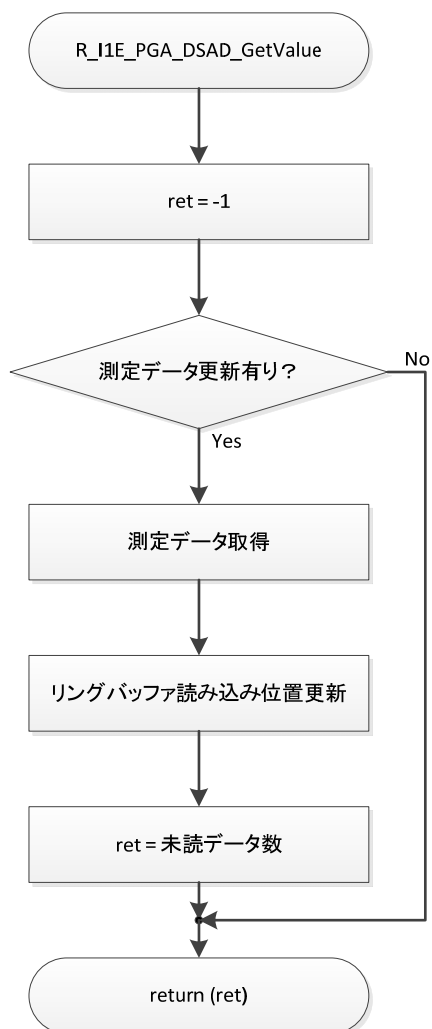
```
int32_t R_I1E_PGA_DSAD_CorrectValue(str_pga_dsad_value_t *p_value)
```

説明	PGAオフセット誤差補正適用値取得関数
引数	*p_value: 24ビットΔΣA/Dコンバータ変換値ポインタ
グローバル変数	g_flash_value: データ・フラッシュ格納データ構造体
SFR	なし
戻り値	int32_t: PGAオフセット誤差適用値。p_value > adc_correctにも同値格納。
処理内容	指定されたA/D変換情報格納用構造体に格納されたデータを元にオフセット誤差適用値を計算し、これを返す。 同時に指定された構造体のメンバadc_correctにも算出された値を格納する。

5.11 24ビット $\Delta\Sigma$ A/Dコンバータ・変換値取得関数

int16_t R_I1E_PGA_DSAD_GetValue(str_pga_dsad_value_t *p_value)

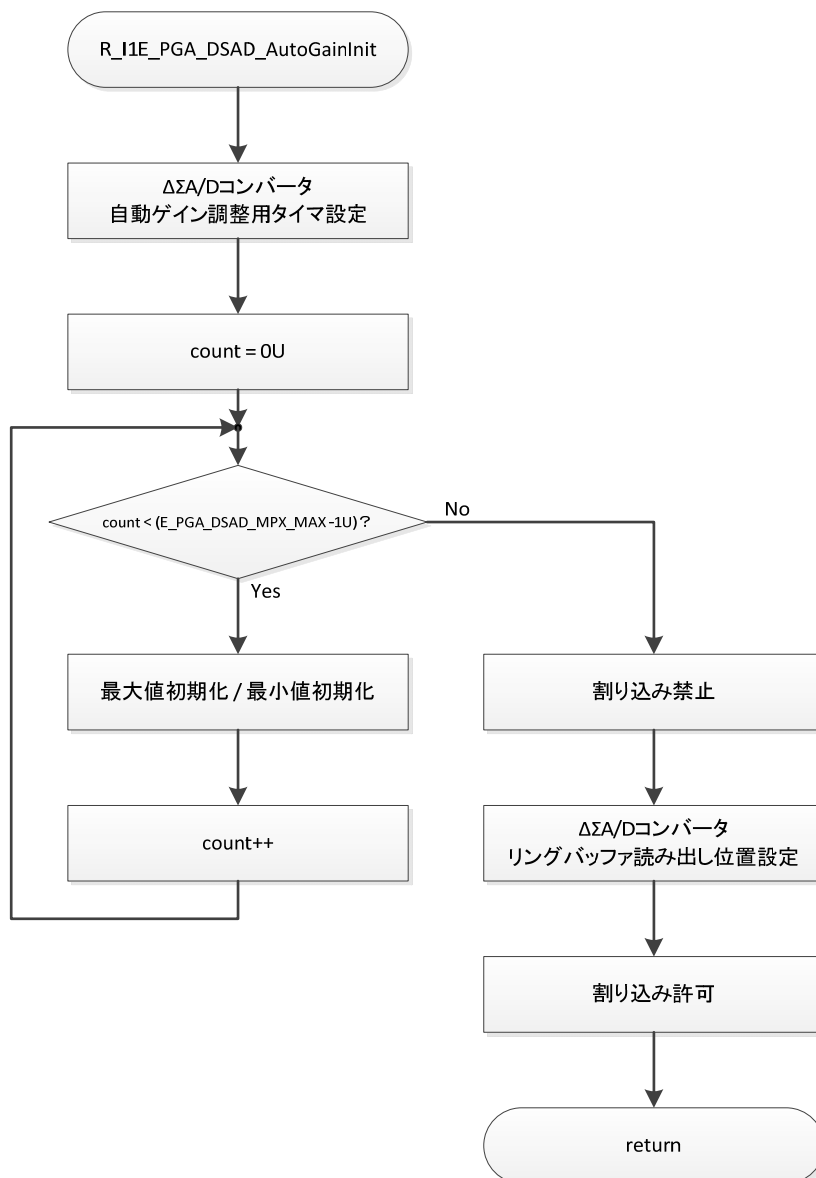
説明	24ビット $\Delta\Sigma$ A/Dコンバータ・変換値取得関数
引数	*p_value: 24ビット $\Delta\Sigma$ A/Dコンバータ 変換値ポインタ
グローバル変数	g_dsad_value_write_pos: DSAD変換値格納バッファ書き込み位置 g_dsad_value_user_read_pos: DSAD変換値格納バッファユーザ読み出し位置 g_dsad_value[]: DSAD変換値格納バッファ
SFR	なし
戻り値	int16_t: 0以上 : データ読み出し完了+未読データ数 1 : バッファエンプティ
処理内容	$\Delta\Sigma$ A/Dコンバータ変換値リングバッファから*p_valueにデータをコピーし、リングバッファ読み出し位置を更新する。 データの読み出しが正常に行われた場合、読み出し後の未読データ数を返す。(0~) 未読データが無い場合、何もせずに-1を返す。



5.12 PGA 自動ゲイン用変数初期化

```
void R_I1E_PGA_DSAD_AutoGainInit(void)
```

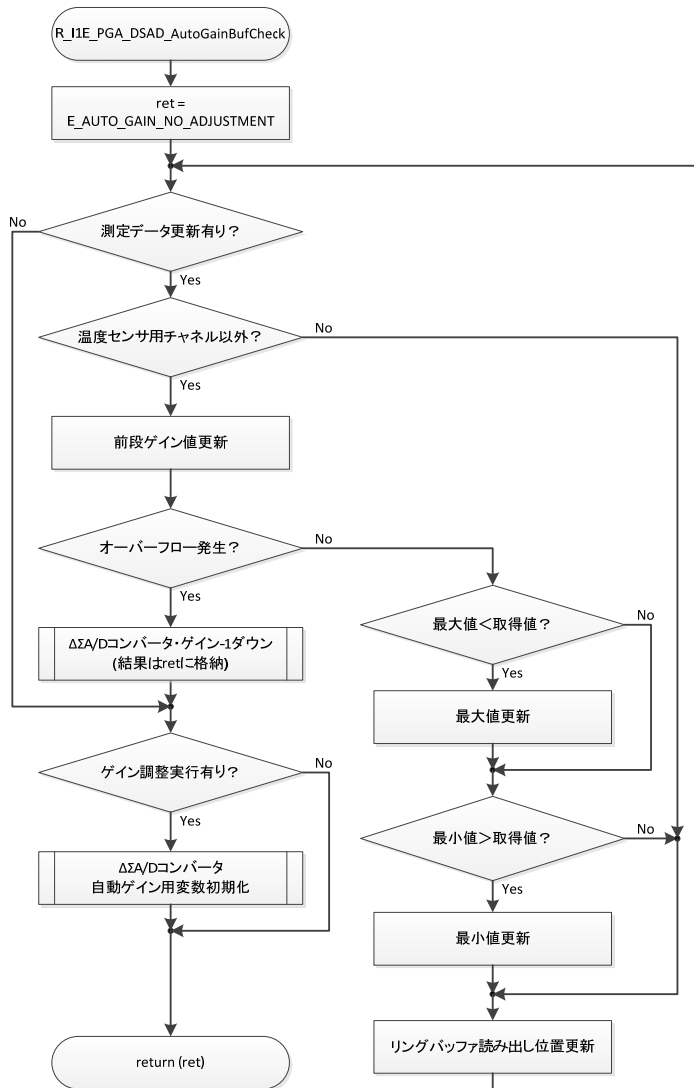
説明	PGA 自動ゲイン用変数初期化
引数	なし
グローバル変数	gs_dsad_auto_gain_buffer[]: PGA 自動ゲイン調整用バッファ g_dsad_value_write_pos: DSAD 変換値格納バッファ書き込み位置 g_dsad_value_read_pos: DSAD 変換値格納バッファ読み出し位置
SFR	なし
戻り値	なし
処理内容	PGA 自動ゲイン調整用変数を初期化する。



5.13 24ビット $\Delta\Sigma$ A/Dコンバータ変換完了後PGA自動ゲイン調整検査処理関数

e_auto_gain_return_t R_I1E_PGA_DSAD_AutoGainBufCheck(void)

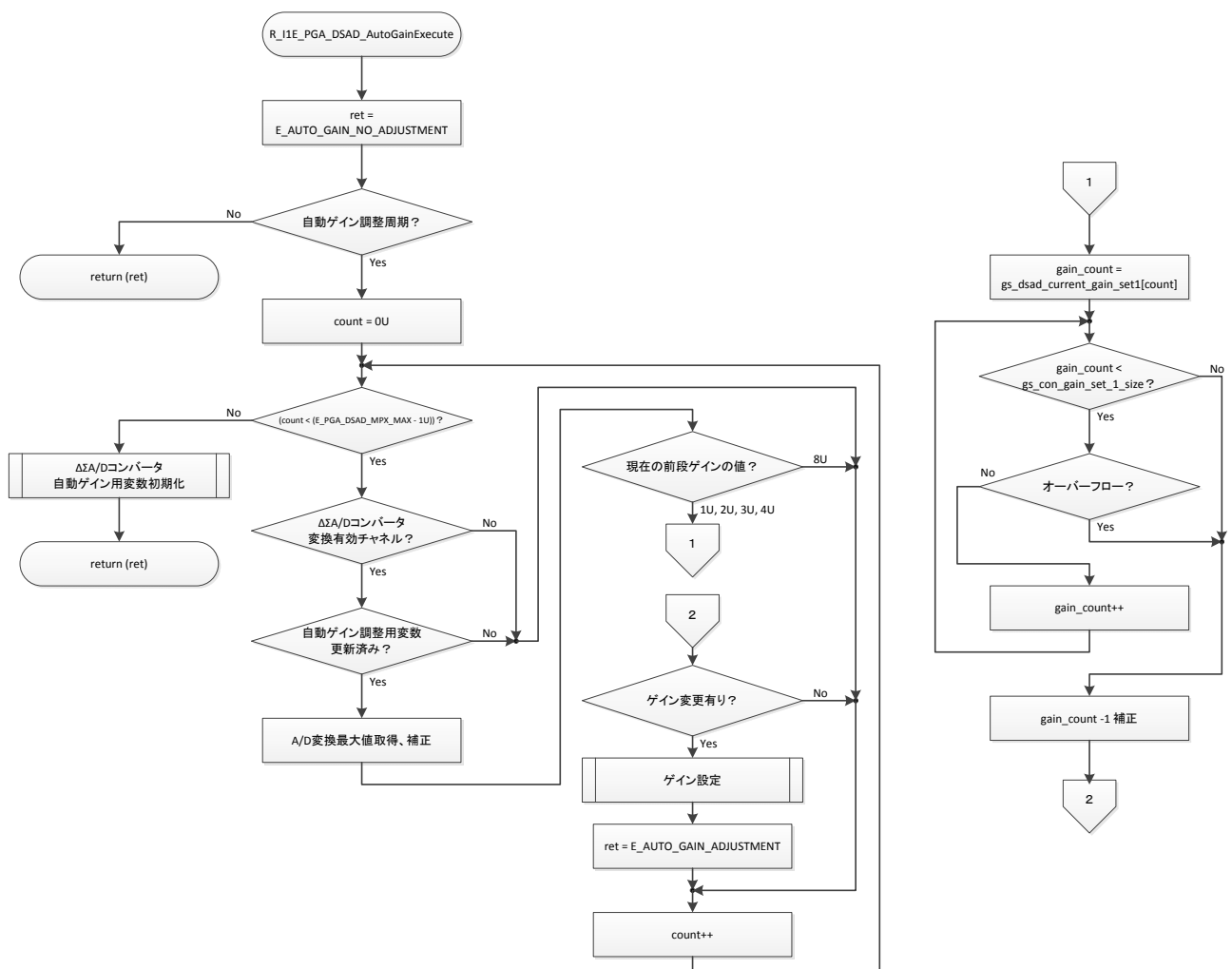
説明	24ビット $\Delta\Sigma$ A/Dコンバータ変換完了後PGA自動ゲイン調整検査処理関数	
引数	なし	
グローバル変数	g_dsad_value_write_pos: DSAD変換値格納バッファ書き込み位置 g_dsad_value_read_pos: DSAD変換値格納バッファ読み出し位置 g_dsad_value[]: DSAD変換値格納バッファ gs_dsad_current_gain_set1[]: 現在の前段ゲイン値 gs_dsad_auto_gain_buffer[]: PGA自動ゲイン調整用バッファ	
SFR	なし	
戻り値	e_auto_gain_return_t: E_AUTO_GAIN_NO_ADJUSTMENT PGAゲイン調整なし E_AUTO_GAIN_ADJUSTMENT PGAゲイン調整あり	
処理内容	$\Delta\Sigma$ A/Dコンバータの測定結果から最大値/最小値を更新し、この値からゲインが適切でないと判断される場合、PGAゲインの自動調整を行う。	



5.14 24ビットΔΣA/Dコンバータ・PGA自動ゲイン調整関数

e_auto_gain_return_t R_I1E_PGA_DSAD_AutoGainExecute(void)

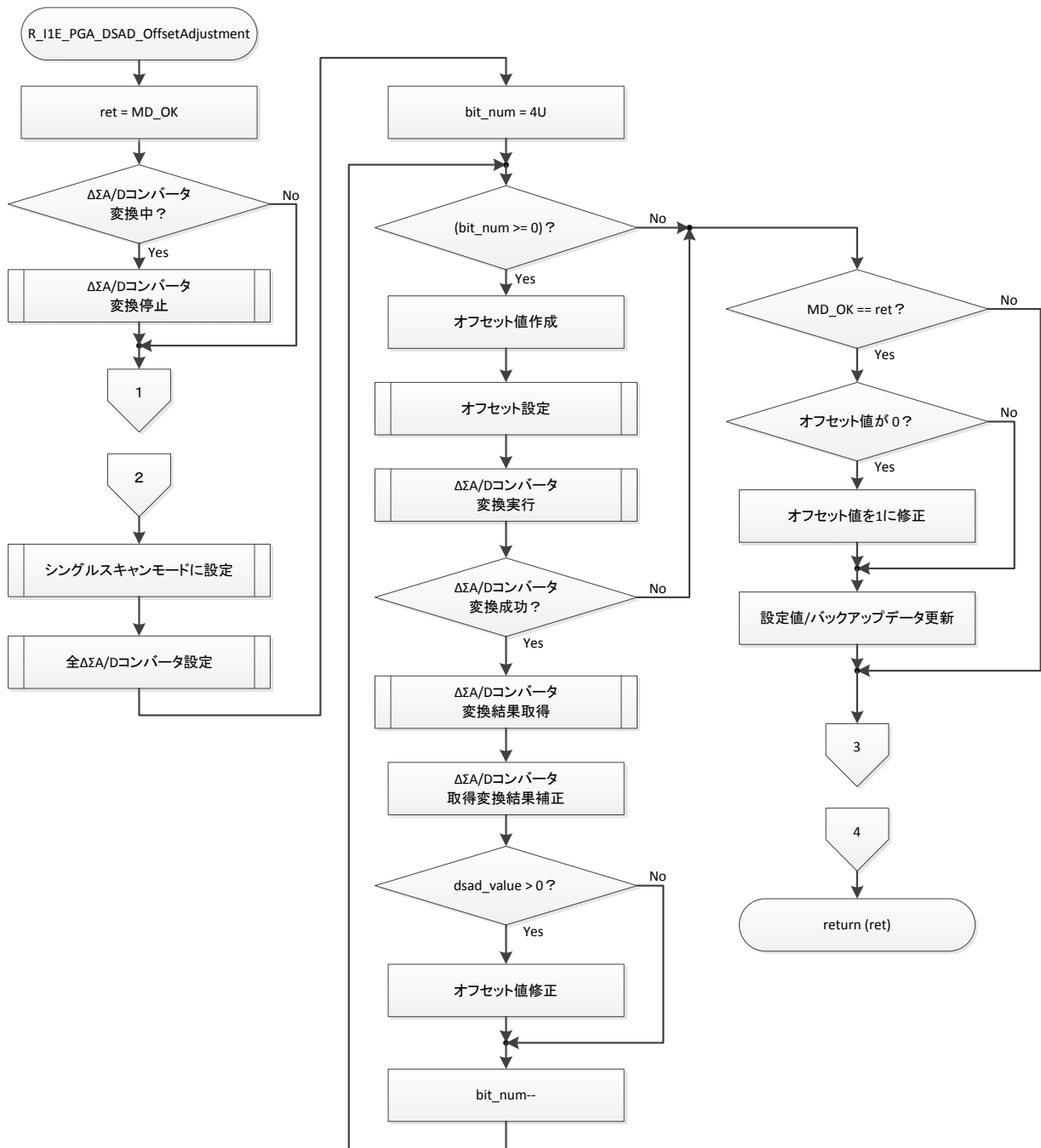
説明	24ビットΔΣA/Dコンバータ・PGA自動ゲイン調整関数	
引数	なし	
グローバル変数	gs_dsad_auto_gain_timer: PGA自動ゲイン調整用タイマ g_dsad_setting[]: 24ビットΔΣA/Dコンバータ設定値 gs_dsad_auto_gain_buffer[]: PGA自動ゲイン調整用バッファ gs_dsad_current_gain_set1[]: 現在の前段ゲイン値 gs_con_gain_set_1_size: 前段ゲインの倍率数	
SFR	なし	
戻り値	e_auto_gain_return_t: E_AUTO_GAIN_NO_ADJUSTMENT PGAゲイン調整なし E_AUTO_GAIN_ADJUSTMENT PGAゲイン調整あり	
処理内容	ΔΣA/Dコンバータの指定入力マルチプレクサにおいてPGA自動ゲイン調整を行う。	

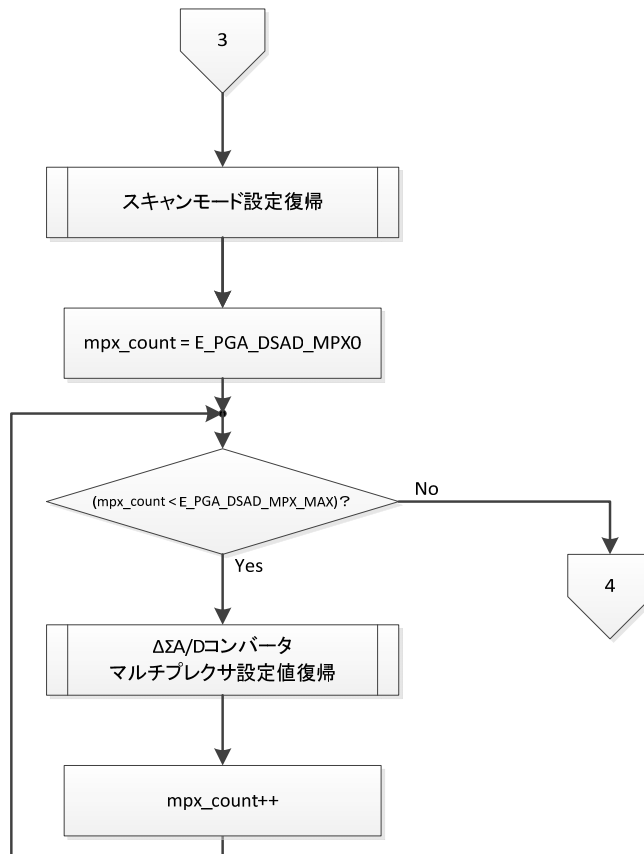
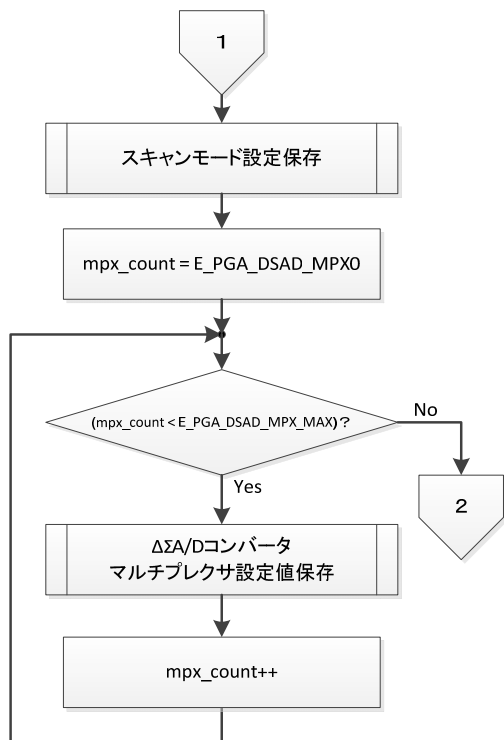


5.15 PGA オフセット調整実行関数

MD_STATUS R_I1E_PGA_DSAD_OffsetAdjustment(e_pga_dsad_mpx_t dsad_mpx)

説明	PGA オフセット調整実行関数
引数	dsad_mpx: 24 ビット $\Delta\Sigma/A/D$ コンバータ 入力マルチプレクサ番号
グローバル変数	なし
SFR	なし
戻り値	MD_STATUS: MD_OK MD_ERROR
処理内容	$\Delta\Sigma/A/D$ コンバータの指定入力マルチプレクサにおいてプログラマブル・ゲイン計装アンプのオフセット調整を行い、結果をフラッシュ格納用構造体のオフセット誤差補正值配列に格納する。 オフセット調整に失敗した場合、MD_ERROR を返す。





5.16 断線検知チェック関数

e_disconnection_return_t R_I1E_PGA_DSAD_DisconnCheck(e_pga_dsad_mpx_t dsad_mpx)

説明	断線検知チェック関数	
引数	dsad_mpx: 24ビットΔΣA/Dコンバータ 入力マルチプレクサ番号	
グローバル変数	なし	
SFR	なし	
戻り値	e_disconnection_return_t: E_DISCONNECTION_NO_DETECT 非断線状態 E_DISCONNECTION_DETECT 断線状態	
処理内容	ΔΣA/Dコンバータの指定入力マルチプレクサにおいて断線検知を行う。 断線を検知した場合、E_DISCONNECTION_DETECTを返す。	



5.17 RL78/I1E AFE 停止処理関数

```
void R_I1E_AFE_Stop(void)
```

説明	RL78/I1E AFE 停止処理関数
引数	なし
グローバル変数	なし
SFR	PER0: 周辺イネーブル・レジスタ 0 AFEPCS: アナログ・フロントエンド電源選択レジスタ AFEPCN: AFE 基準電源 (ABGR) 部の電源制御レジスタ PER1: 周辺イネーブル・レジスタ 1 AFEEN: AFE 電源・クロック制御部入カクロック供給の制御レジスタ
戻り値	なし
処理内容	アナログ・フロントエンドを停止状態にする。

5.18 RL78/I1E AFE 再始動処理関数

```
void R_I1E_AFE_ReStart(void)
```

説明	RL78/I1E AFE 再始動処理関数
引数	なし
グローバル変数	なし
SFR	AFEEN: AFE 電源・クロック制御部入カクロック供給の制御レジスタ AFEPCN: AFE 基準電源 (ABGR) 部の電源制御レジスタ AFESTAT: AFE 基準電源 (ABGR) 部の電源状態 AFECKS: アナログ・フロントエンド・クロック選択レジスタ PGAEN: PGA および 24 ビット $\Delta\Sigma$ /D コンバータの入カクロック供給の制御 DSADMR: $\Delta\Sigma$ /D コンバータ・モード・レジスタ PGAPON: プログラマブル・ゲイン計装アンプ (PGA) 部の電源制御 PGASTAT: プログラマブル・ゲイン計装アンプ (PGA) 部の電源状態
戻り値	なし
処理内容	アナログ・フロントエンドを使用可能状態にする。

5.19 コンフィギュラブル・アンプ トリミング実行関数

```
void R_I1E_CAMP_Calibration(uint8_t *camp_trimming)
```

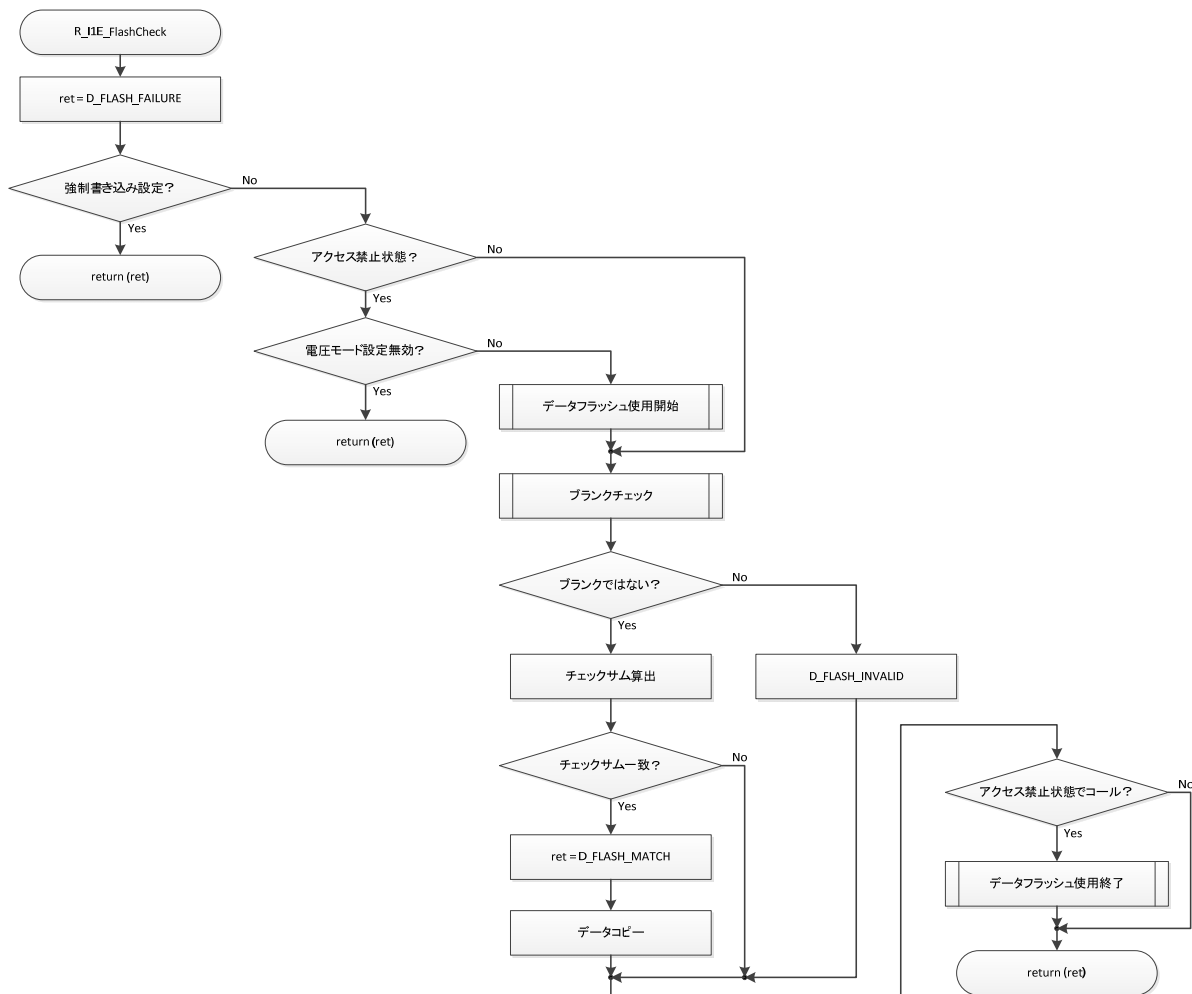
説明	コンフィギュラブル・アンプ トリミング実行関数
引数	*camp_trimming: コンフィギュラブル・アンプ n トリミング設定値
グローバル変数	なし
SFR	AMP0CAL: コンフィギュラブル・アンプ 0 トリミング・レジスタ AMP1CAL: コンフィギュラブル・アンプ 1 トリミング・レジスタ AMP2CAL: コンフィギュラブル・アンプ 2 トリミング・レジスタ
戻り値	なし
処理内容	コンフィギュラブル・アンプの入力オフセットトリミングを行い、*camp_trimming に結果を格納する。

6. フラッシュ・メモリ関連 API

6.1 データフラッシュ・メモリ格納データチェック関数

uint8_t R_IIE_FlashCheck(void)

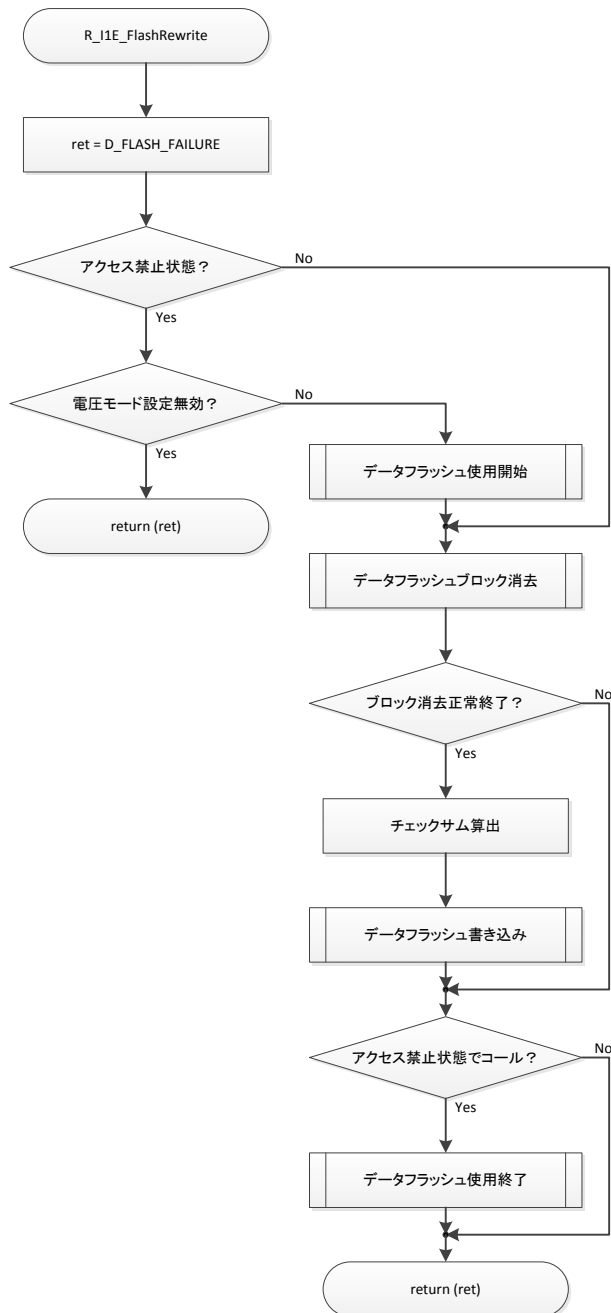
説明	データフラッシュ・メモリ格納データチェック関数
引数	なし
グローバル変数	g_flash_value: データ・フラッシュ格納データ構造体
SFR	DFLCTL: データ・フラッシュ・コントロール・レジスタ
戻り値	uint8_t: D_FLASH_MATCH D_FLASH_FAILURE D_FLASH_INVALID
処理内容	<p>データ・フラッシュの特定ブロックが空白ではなかった場合、かつ、データ・フラッシュに書き込まれているデータとチェックサムに齟齬がなかった場合、フラッシュ格納用構造体にデータを読み出し、D_FLASH_MATCH を返す。</p> <p>データ・フラッシュの特定ブロックが空白だった場合は、データの読み出しは行わずに D_FLASH_INVALID を返す。</p> <p>データ・フラッシュに記録されているデータが異常だった場合、データ・フラッシュへのアクセスができなかった場合、及び、フラッシュデータ強制書き込み設定の場合、フラッシュ格納用構造体への読み出しは行わずに D_FLASH_FAILURE を返す。</p>



6.2 データフラッシュ・メモリ格納データ上書き関数

uint8_t R_I1E_FlashRewrite(str_flash_data_t *p_flash_value)

説明	データフラッシュ・メモリ格納データ上書き関数
引数	*p_flash_value: フラッシュ・メモリ書き込みデータ格納バッファポインタ
グローバル変数	なし
SFR	DFLCTL: データ・フラッシュ・コントロール・レジスタ
戻り値	uint8_t: D_FLASH_SUCCESS D_FLASH_FAILURE
処理内容	データ・フラッシュの特定ブロックに指定データとチェックサムの上書きを行う。 書き込みに失敗した場合は D_FLASH_FAILURE を返す。

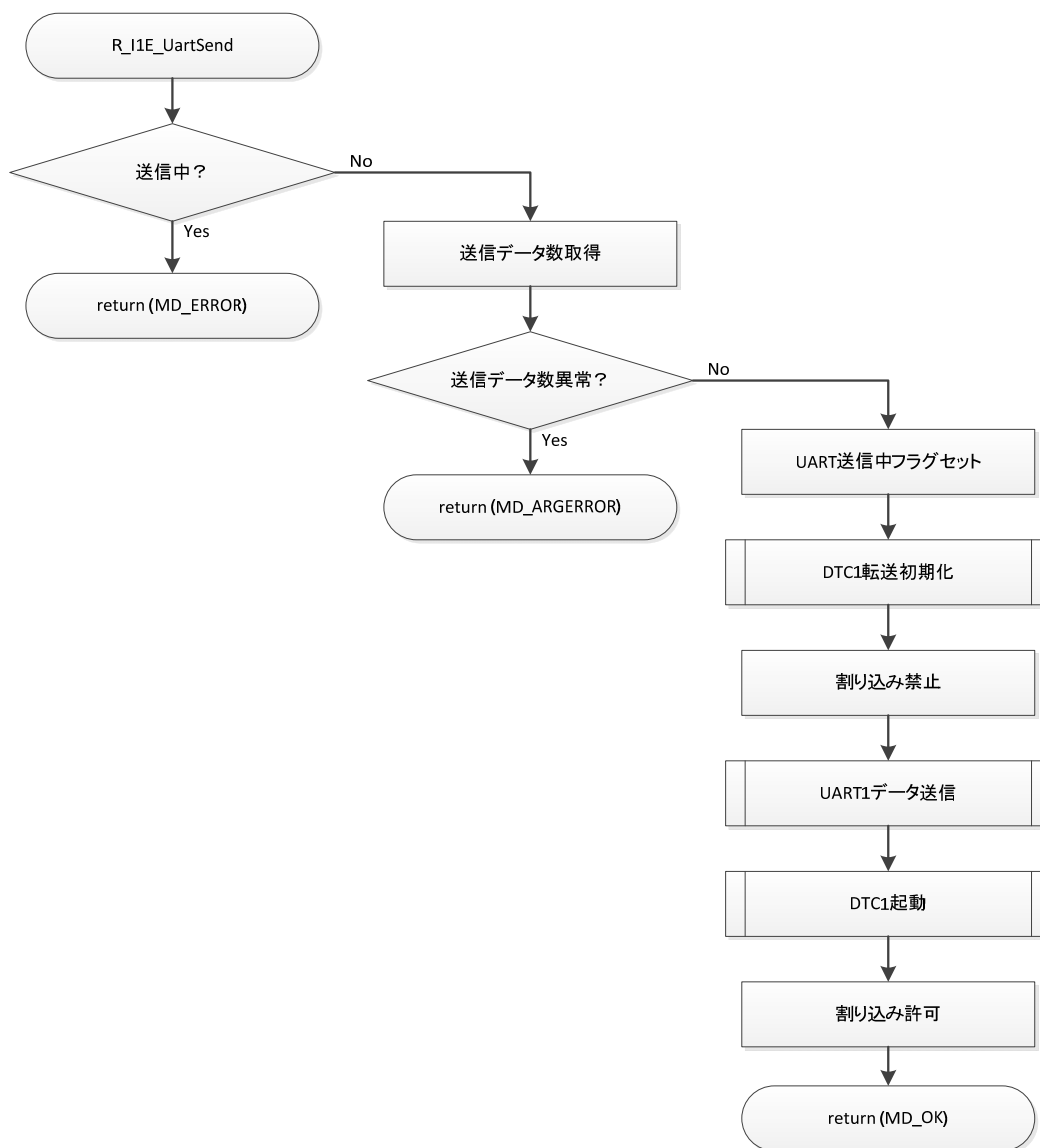


7. UART 通信関連 API

7.1 PC 送信処理設定関数(ASCII)。[1 回の送信で 256byte までの制限付き]

MD_STATUS R_I1E_UartSend(int8_t *p_send_data)

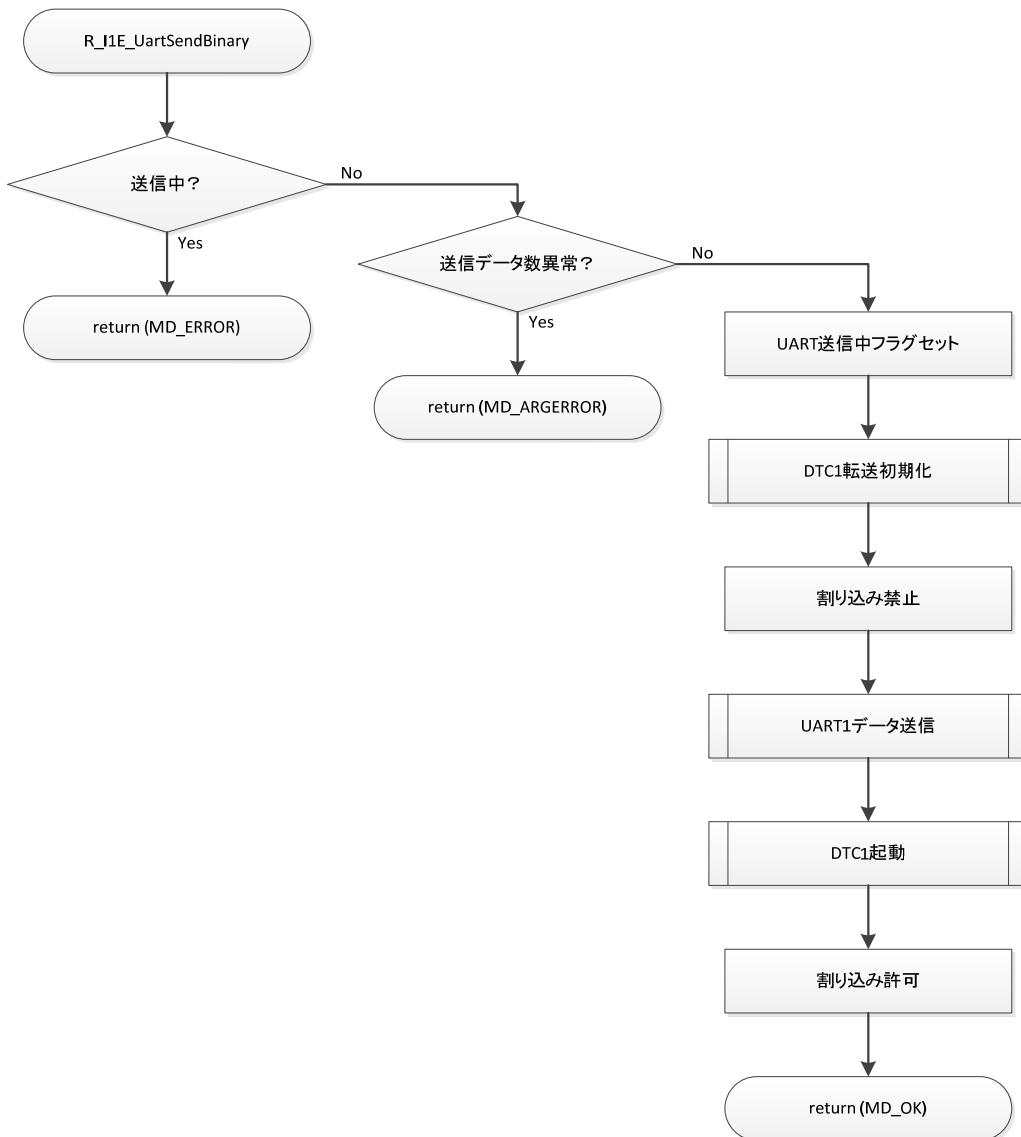
説明	PC 送信処理設定関数(ASCII)。[1 回の送信で 256byte までの制限付き]
引数	*p_send_data: 送信バッファアドレス
グローバル変数	g_tx_in_process_flag: UART 送信中フラグ
SFR	なし
戻り値	MD_STATUS: MD_OK MD_ERROR MD_ARGERROR
処理内容	データ・トランスファ・コントローラのコントロールデータ 1 を使用し、UART1 より指定文字列の送信を行う。 既に送信中の場合、何もせずに MD_ERROR を返す。 指定文字列サイズが不正な場合、何もせずに MD_ARGERROR を返す。



7.2 PC 送信処理設定関数(Binary)。[1 回の送信で 256byte までの制限付き]

MD_STATUS R_I1E_UartSendBinary(int8_t *p_send_data, uint16_t send_length)

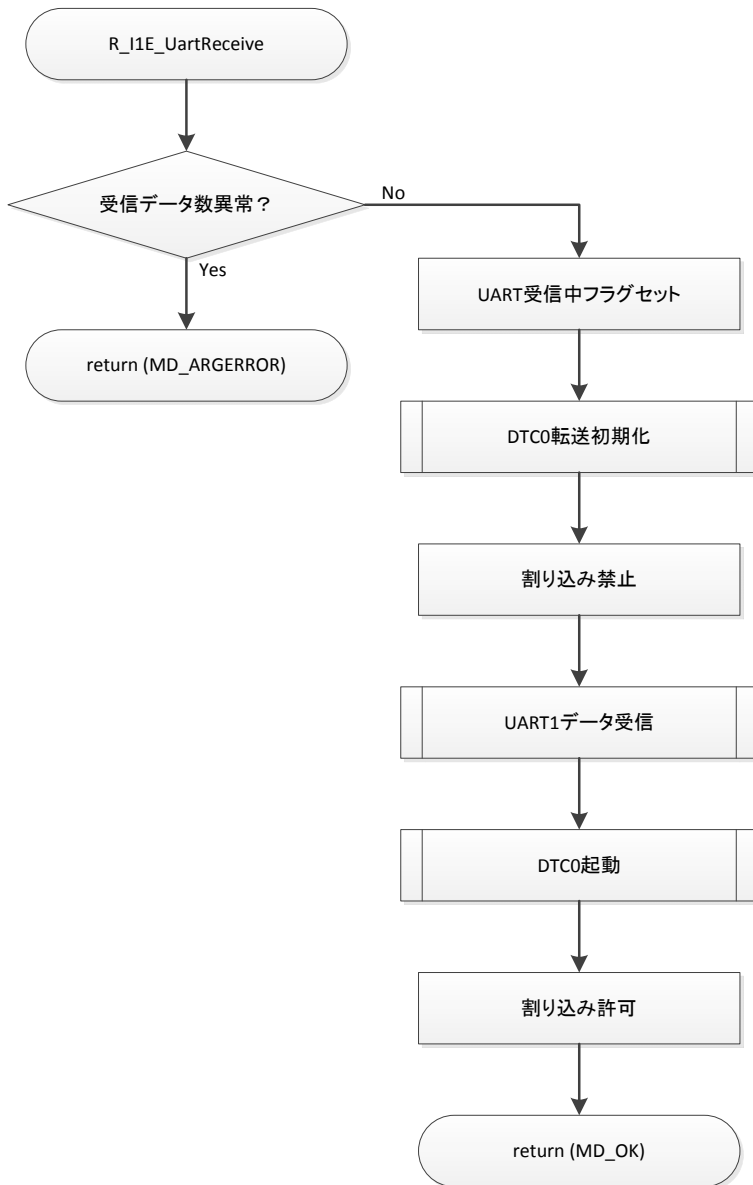
説明	PC 送信処理設定関数(Binary)。[1 回の送信で 256byte までの制限付き]
引数	*p_send_data: 送信バッファアドレス send_length: 送信サイズ
グローバル変数	g_tx_in_process_flag: UART 送信中フラグ
SFR	なし
戻り値	MD_STATUS: MD_OK MD_ERROR MD_ARGERROR
処理内容	データ・トランスファ・コントローラのコントロールデータ 1 を使用し、UART1 より指定バイト数の送信を行う。 既に送信中の場合、何もせずに MD_ERROR を返す。 送信サイズが不正な場合、何もせずに MD_ARGERROR を返す。



7.3 PC からの受信処理(DTC0 使用)。[1 回の受信で 256byte までの制限付き]

MD_STATUS R_I1E_UartReceive(int8_t *p_receive_data, uint8_t receive_length)

説明	PC からの受信処理(DTC0 使用)。[1 回の受信で 256byte までの制限付き]
引数	*p_receive_data: 受信バッファアドレス receive_length: 指定受信サイズ
グローバル変数	g_rx_in_process_flag: UART 受信中フラグ
SFR	なし
戻り値	MD_STATUS: MD_OK MD_ARGERROR
処理内容	データ・トランスファ・コントローラのコントロールデータ 0 を使用し、UART1 より指定バイト数の受信を行う。 指定受信サイズが不正な場合、何もせずに MD_ARGERROR を返す。



8. フリーランタイム関連 API

8.1 フリーランタイム・設定初期化関数

```
void R_TAU_FreeRunTimerInit(void)
```

説明	フリーランタイム・設定初期化関数
引数	なし
グローバル変数	なし
SFR	TDR11: タイマ・データ・レジスタ 11
戻り値	なし
処理内容	フリーランタイムの設定を初期化し、タイマ動作を開始する。

8.2 フリーランタイム・停止関数

```
void R_TAU_FreeRunTimerStop(void)
```

説明	フリーランタイム・停止関数
引数	なし
グローバル変数	なし
SFR	なし
戻り値	なし
処理内容	フリーランタイムを停止する。

8.3 フリーランタイム・現在カウント値取得関数

```
uint32_t R_GetTickCount(void)
```

説明	フリーランタイム・現在カウント値取得関数
引数	なし
グローバル変数	g_freerun_timer: フリーランタイムカウンタ変数
SFR	TCR11: タイマ・カウンタ・レジスタ 11
戻り値	uint32_t: 32bit のフリーランタイム値
処理内容	フリーランタイムの現在のカウント値を取得し、これを返す。

8.4 フリーランタイム・カウント値比較関数

```
int32_t R_CmpTickCount(uint32_t freerun_counter)
```

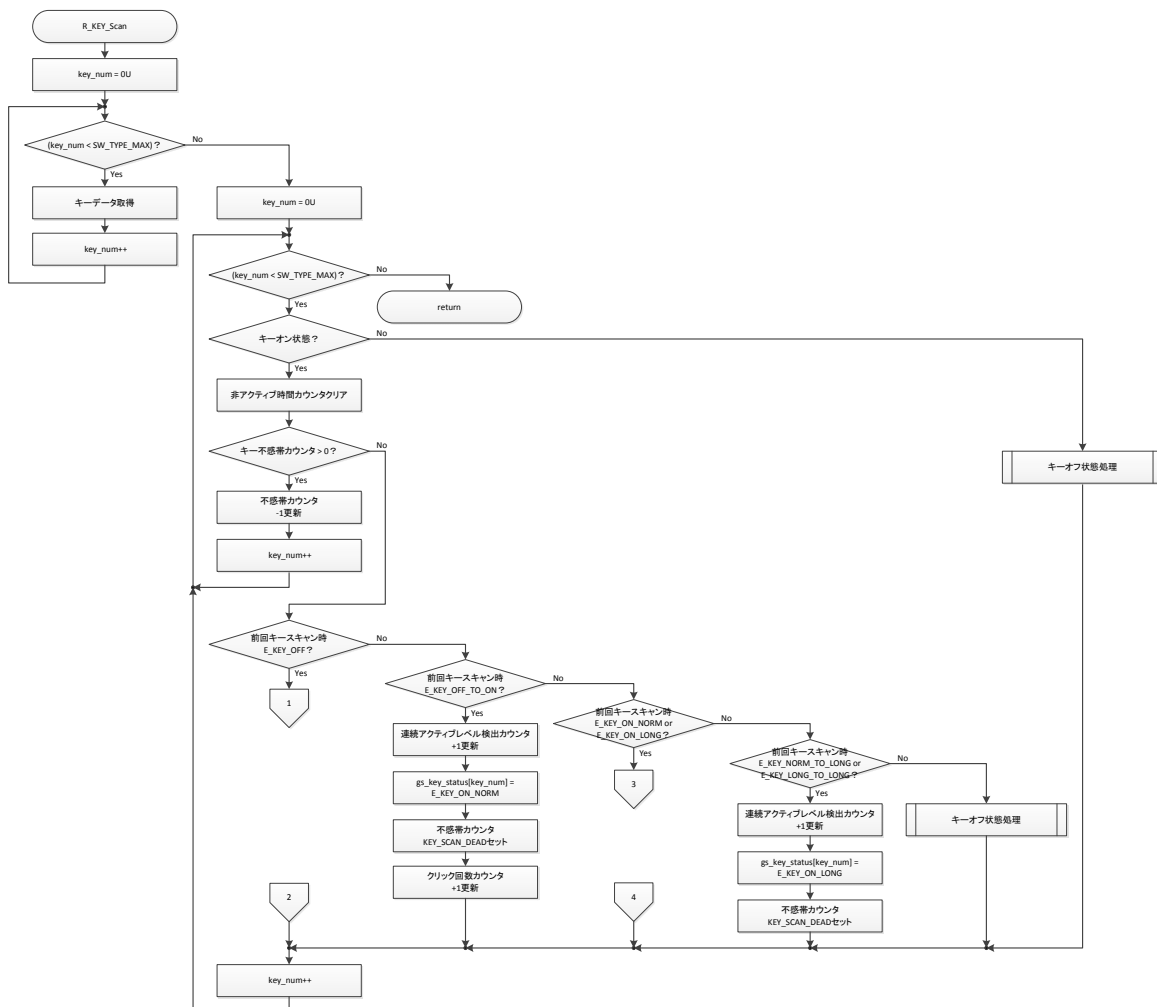
説明	フリーランタイム・カウント値比較関数
引数	freerun_counter: 比較対象のタイマ・カウント値
グローバル変数	なし
SFR	なし
戻り値	int32_t: 0 未満 : 未経過状態 0 以上 : 経過状態
処理内容	フリーランタイムの現在のカウント値と指定のカウント値を比較し、結果を返す。

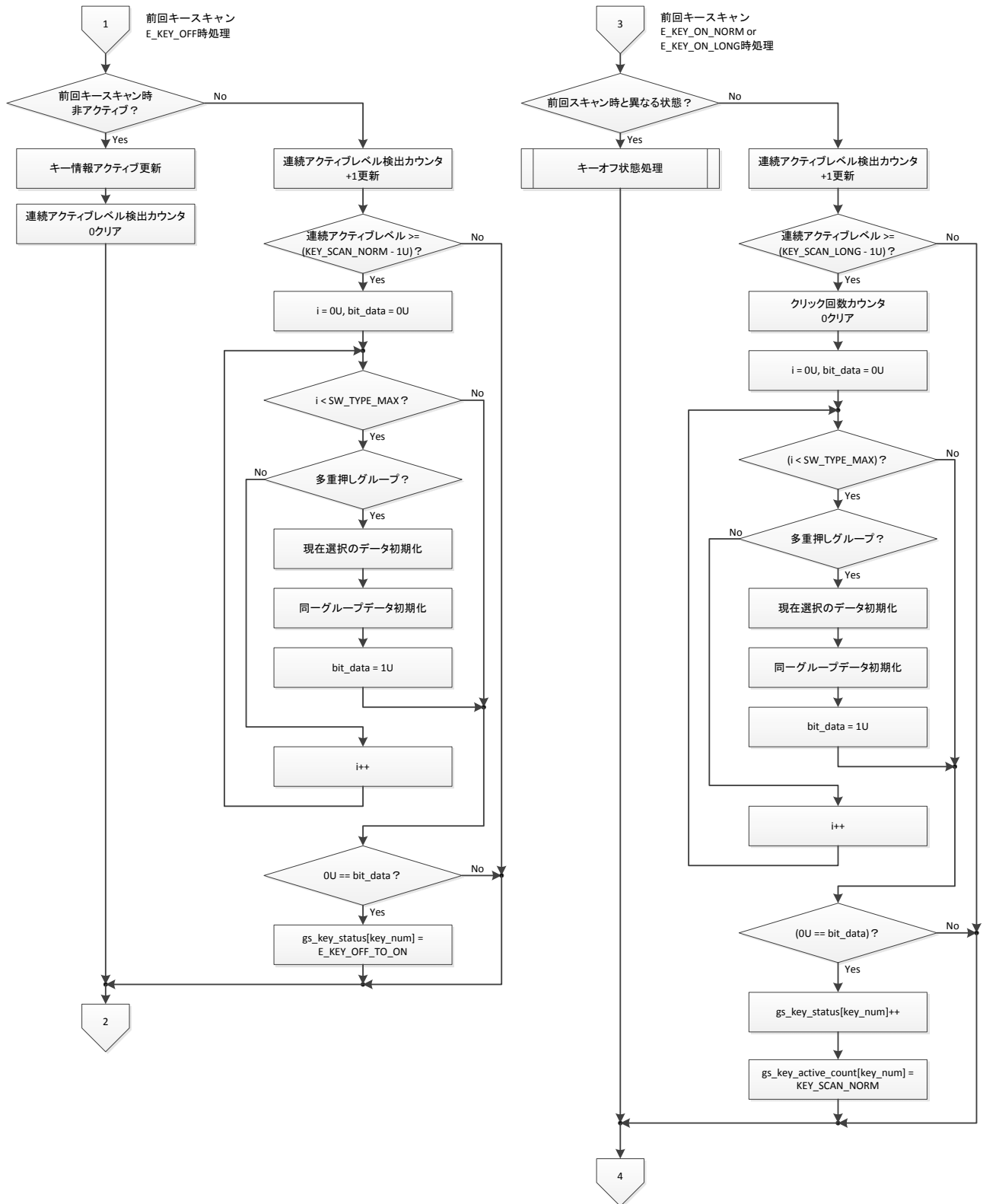
9. キー判定関連 API

9.1 キースキャン処理

void R_KEY_Scan(void)

説明	キースキャン処理
引数	なし
グローバル変数	g_con_key_setting[]: ユーザー設定キー情報 gs_key_status[]: キー情報配列 gs_key_setting_old: 最終キー情報 gs_key_active_count[]: 連続アクティブレベル検出回数カウンタ配列 gs_key_dead_zone_count[]: スイッチ状態変更後の不感帯カウンタ配列 gs_key_non_active_count[]: 連続非アクティブレベル検出回数カウンタ配列 gs_key_click_count[]: クリック回数カウンタ配列
SFR	なし
戻り値	なし
処理内容	キー入力として設定されたポートの入力状態を確認し、各キー情報を更新する。

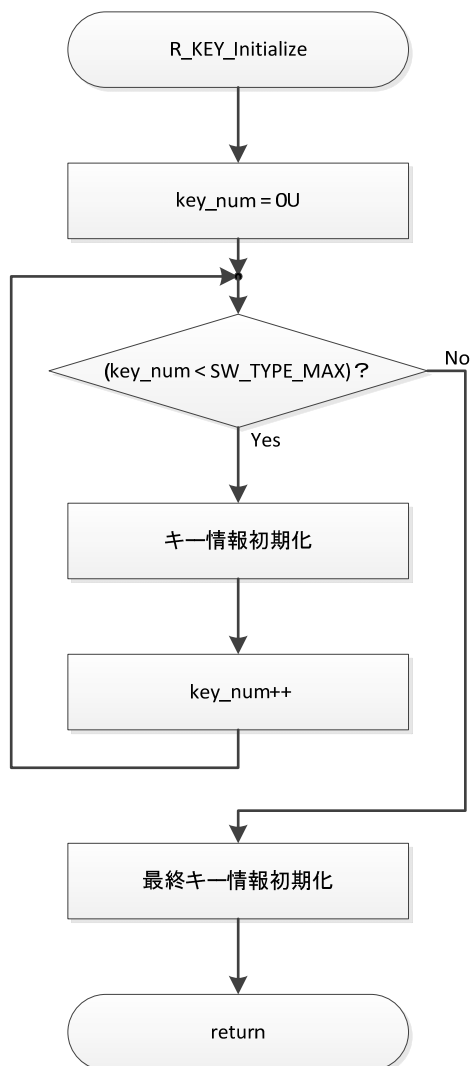




9.2 キー情報初期化

```
void R_KEY_Initialize(void)
```

説明	キー情報初期化
引数	なし
グローバル変数	gs_key_setting_old: 最終キー情報 gs_key_active_count[]: 連続アクティブレベル検出回数カウンタ配列 gs_key_dead_zone_count[]: スイッチ状態変更後の不感帯カウンタ配列 gs_key_non_active_count[]: 連続非アクティブレベル検出回数カウンタ配列 gs_key_click_count[]: クリック回数カウンタ配列 gs_key_status[]: キー情報配列
SFR	なし
戻り値	なし
処理内容	各キー情報の初期化を行う。



9.3 キー情報取得 (指定キーのみ)

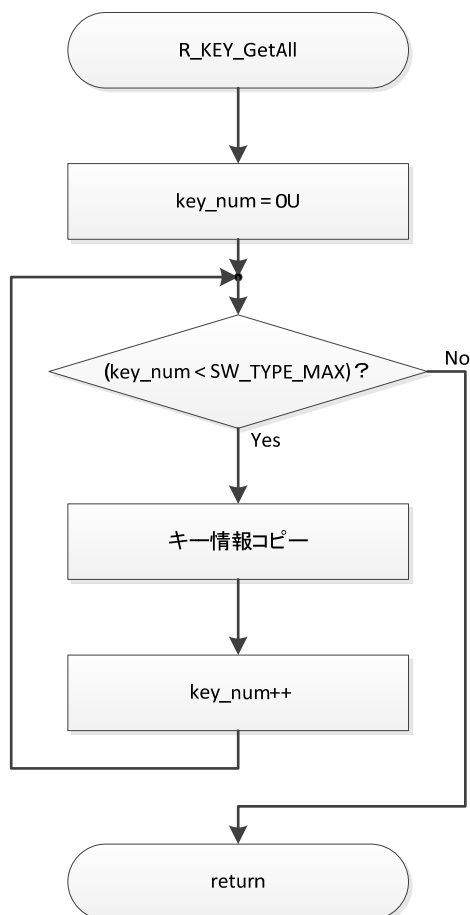
e_key_status_t R_KEY_Get(e_key_t const key)

説明	キー情報取得 (指定キーのみ)		
引数	key:		
		e_key_t で設定した SW 名	
グローバル変数	gs_key_status[]:		
		キー情報配列	
SFR	なし		
戻り値	e_key_status_t:		
	E_KEY_OFF	0	The key is not pushed
	E_KEY_OFF_TO_ON	1	The key was pushed (off to normal push)
	E_KEY_ON_NORM	2	The key was pushed (normal push)
	E_KEY_NORM_TO_LONG	3	The key was pushed (normal to long push)
	E_KEY_ON_LONG	4	The key was pushed (push and hold)
	E_KEY_LONG_TO_LONG	5	The key was pushed (long to long push)
	E_KEY_DOUBLE_CLICK	6	The key was double pushed
処理内容	指定されたキーの情報を取得し、これを返す。		
	※この関数は既に取得されたキー情報を返すだけなので、事前に R_KEY_Scan()の実行が必要。		

9.4 キー情報取得 (全キー)

```
void R_KEY_GetAll(e_key_status_t * p_status)
```

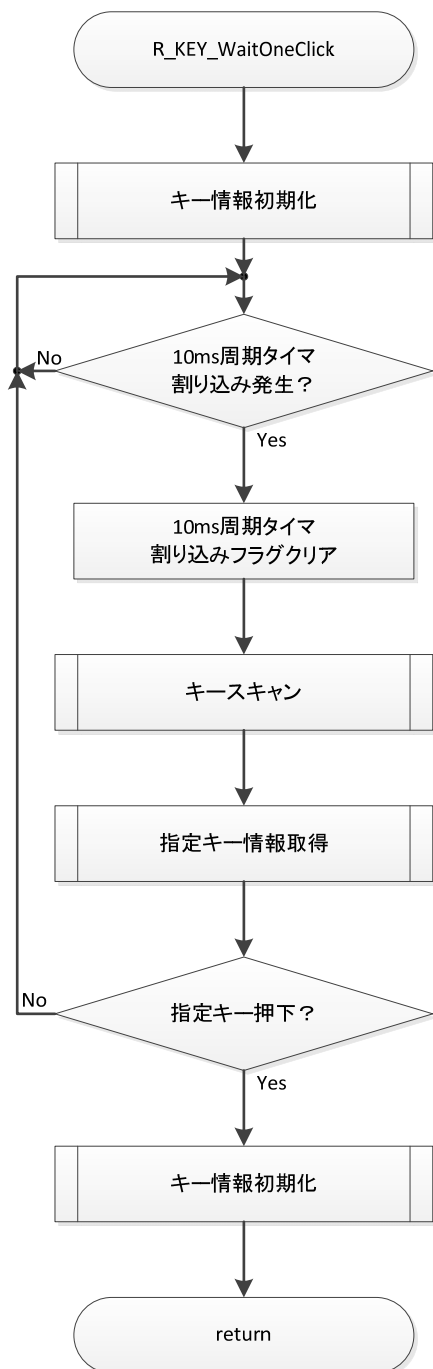
説明	キー情報取得 (全キー)
引数	p_status: キー情報配列ポインタ
グローバル変数	gs_key_status[]: キー情報配列
SFR	なし
戻り値	なし
処理内容	全てのキー情報を指定されたキー情報配列に格納する。 ※この関数は既に取得されたキー情報をコピーするだけなので、事前に R_KEY_Scan()の実行が必要。



9.5 指定キー押下待ち

```
void R_KEY_WaitOneClick(e_key_t const key)
```

説明	指定キー押下待ち
引数	key: e_key_t で設定した SW 名
グローバル変数	g_10ms_timer_flag: 10ms タイマ周期フラグ
SFR	なし
戻り値	なし
処理内容	指定されたキーが押下されるまでループして待つ。



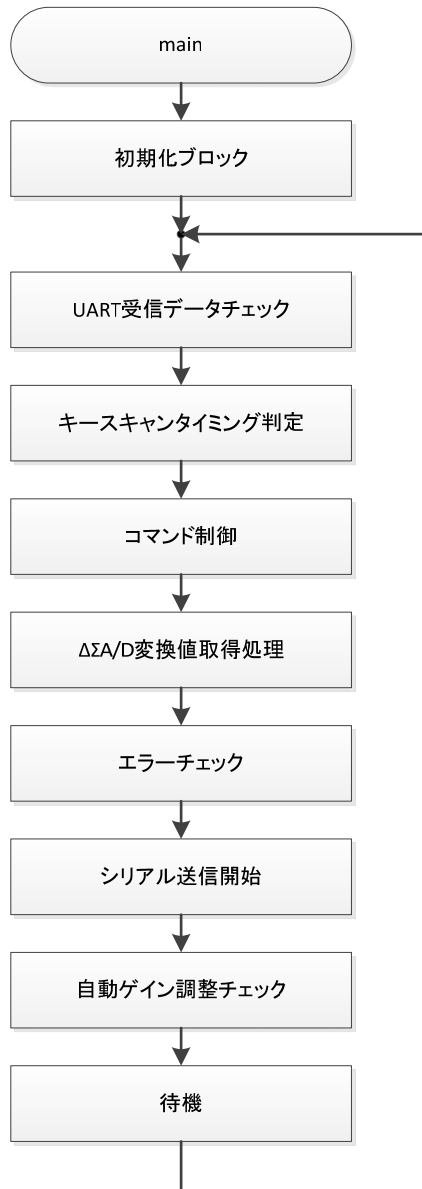
10. サンプルコード動作仕様

本サンプルコードでは24bit $\Delta \Sigma$ A/Dコンバータ動作を周期的に行い、シリアル・アレイ・ユニット (UART1) を介して PC へデータを送信します。CS+のコード生成で24bit $\Delta \Sigma$ A/Dコンバータのオートスキャンのモードを選択することで、A/D変換の間欠動作または連続動作となります。

メイン関数の動作説明およびPCへ送信する通信仕様について以下に示します。

10.1 メイン関数動作説明

メイン関数内の初期化後の処理は大きく分けて9つのブロックで構成されており、以下のフローチャートのようになっています。

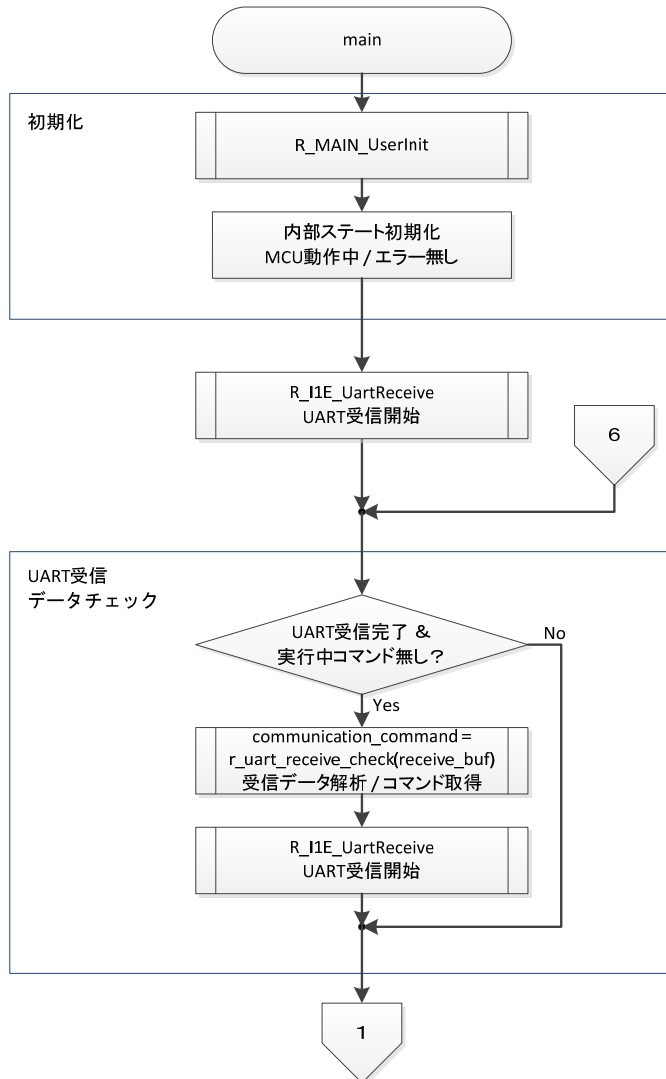


10.1.1 UART 受信データチェックブロック

UART 受信データチェックブロックでは UART からのデータ受信状態の確認と受信時のコマンド解析、取得、及び UART の受信再開処理を行います。

UART のデータ受信自体はデータ・トランスファ・コントローラにより行われており、このブロックはデータ受信が完了していた場合にのみ実行されます。

動作詳細は以下のフローチャートのようにになります。

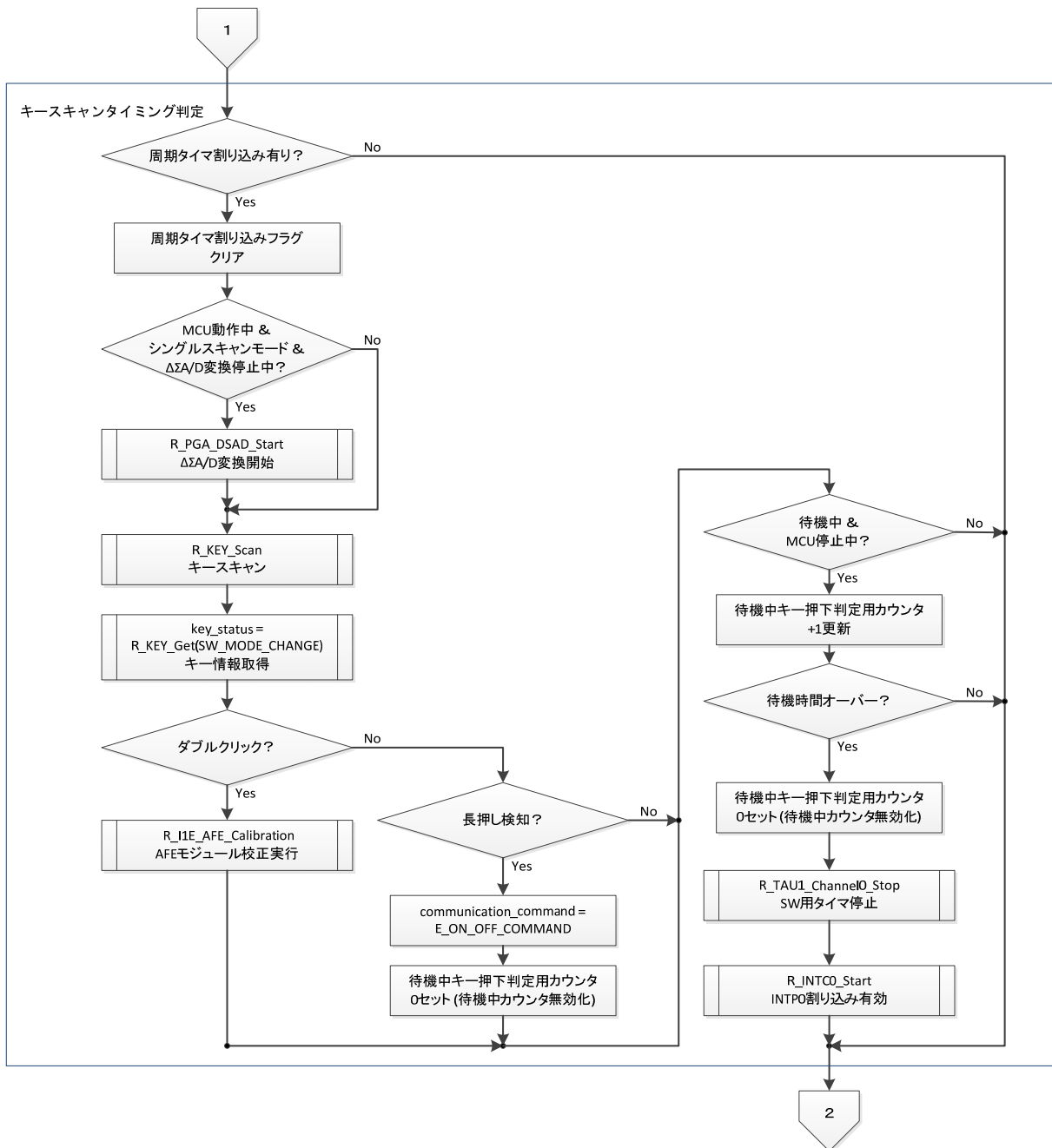


10.1.2 キースキャンタイミング判定ブロック

キースキャンタイミング判定ブロックでは周期タイマ割り込みのタイミングで以下の処理を行います。

- シングル・スキャンモード時、 $\Delta\Sigma A/D$ 変換の間欠実行。
- キー入力情報の取得。
- 取得したキー入力情報に合わせた各種処理。
 - ダブルクリック ⇒ AFE モジュール校正実行。
 - 長押し ⇒ A/D 変換 ON/OFF トグル処理。
- 待機中の有効キー入力監視と、一定時間キー入力が無かった場合のキー入力無効化。

動作詳細は以下のフローチャートのようにになります。

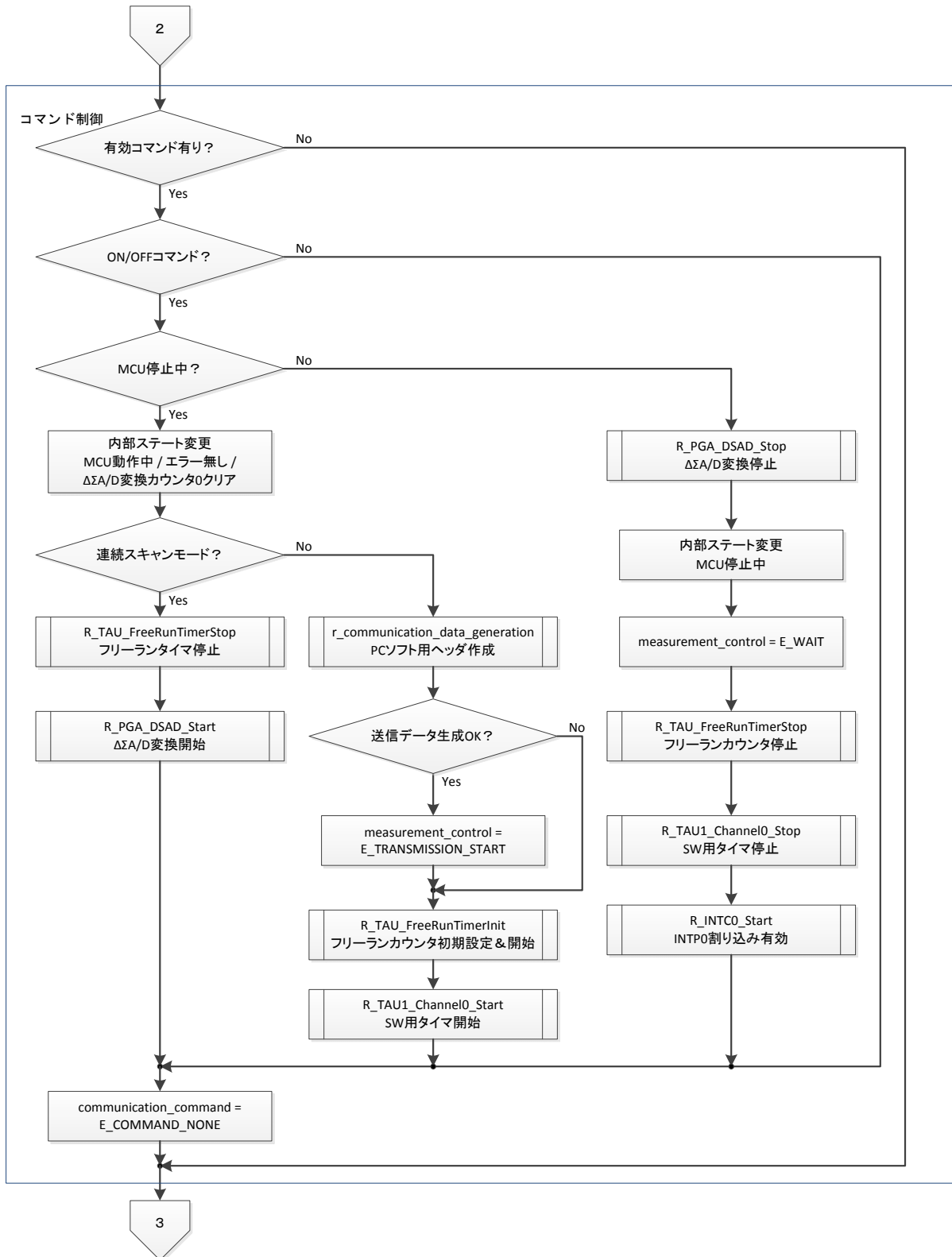


10.1.3 コマンド制御ブロック

コマンド制御ブロックでは受信したコマンドに合わせた各種処理を行います。

本サンプルでは ON/OFF コマンドのみ用意されています。

動作詳細は以下のフローチャートのようにになります。



10.1.4 ΔΣA/D 変換値取得処理ブロック

ΔΣA/D 変換値取得処理ブロックではスキャンモードに応じた PC への送信データ生成を行います。また、送信データ数のカウントも行っており、一定数の A/D 変換が行われると、ΔΣA/D 変換の停止処理も行います。

動作詳細は以下のフローチャートのようにになります。



10.1.5 エラーチェックブロック

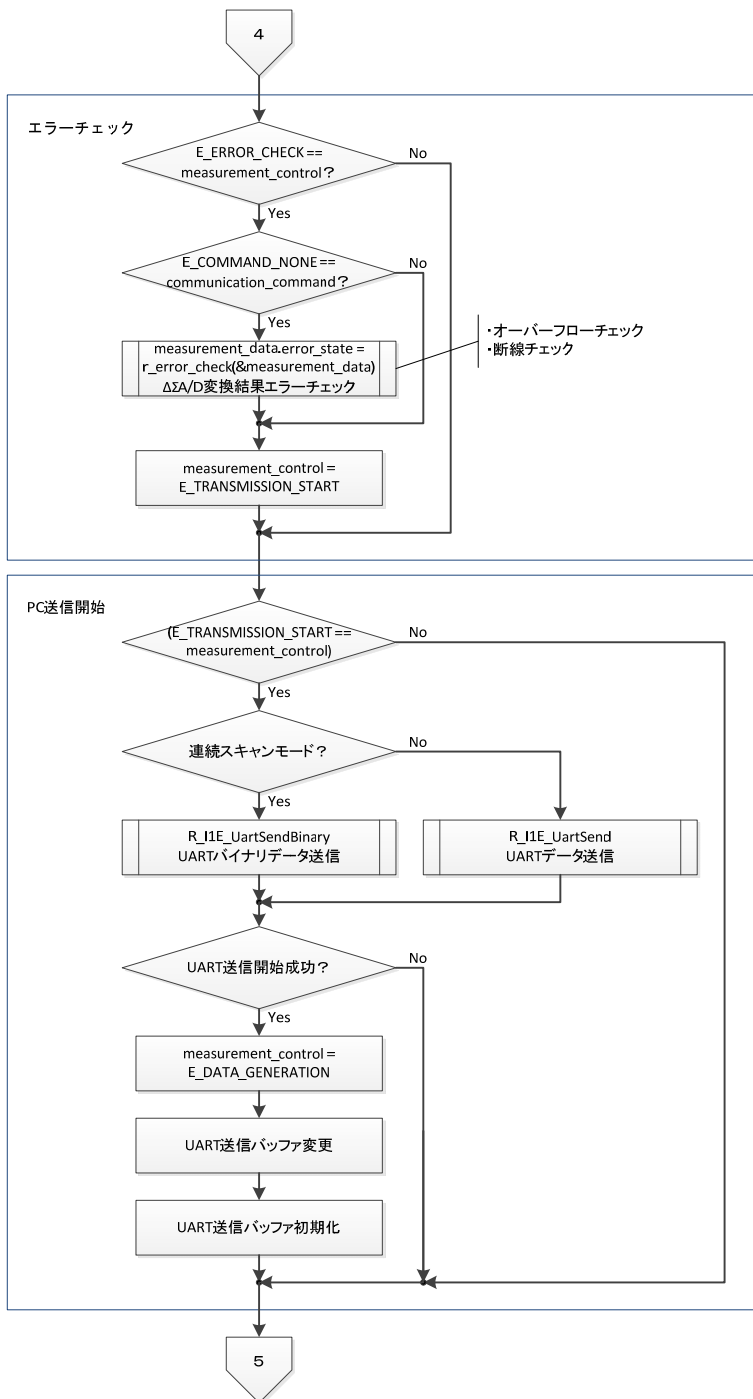
エラーチェックブロックでは $\Delta\Sigma/D$ 変換結果のエラーチェックを行います。

10.1.6 シリアル送信ブロック

シリアル送信ブロックではスキャンモードに合わせた方法(Binary/ASCII)でPCへの送信を開始します。データの送信自体はデータ・トランスファ・コントローラが行っており、このブロックでは送信の開始処理のみを行います。

また、このブロックでは送信バッファの管理も行っており、データ送信の開始に成功した場合、データを格納する送信バッファの変更処理を行います。

動作詳細は以下のフローチャートのようにになります。



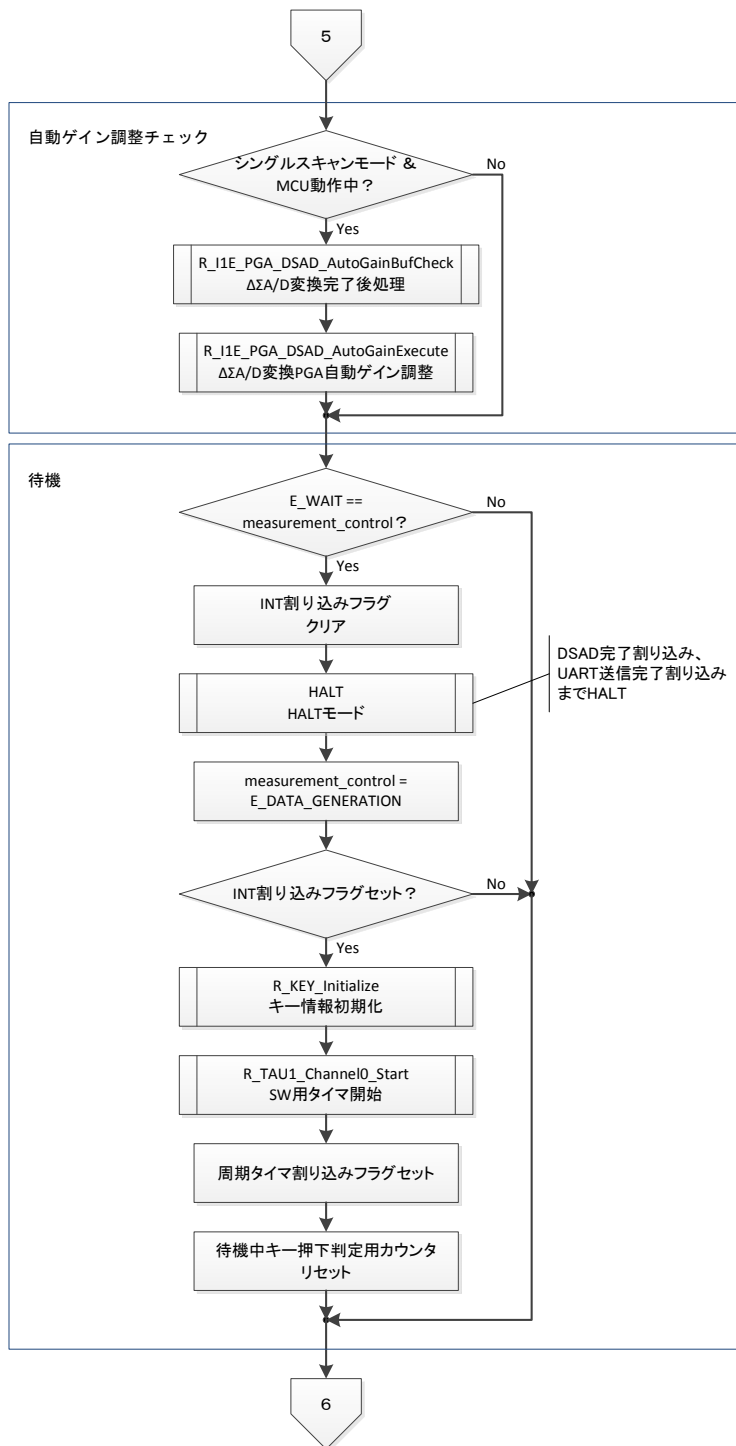
10.1.7 PGA 自動ゲイン調整チェックブロック

PGA 自動ゲイン調整チェックブロックでは、シングル・スキャンモードで動作している場合に PGA 自動ゲイン調整についての各種処理を行います。

10.1.8 待機ブロック

待機ブロックでは他ブロックの実行終了後の HALT 命令実行による待機状態への移行処理を行います。待機状態は各種割り込みにより解除され、復帰要因により各種初期化処理も合わせて実行します。

動作詳細は以下のフローチャートのようにになります。



10.1.9 タイミングチャート

測定動作のタイミングは、 $\Delta\Sigma$ A/D の DSADSCM ビット(オートスキャンのモード選択)により大きく2つに分かれます。シングル・スキャン・モードの場合を図 10.1 エラー! 参照元が見つかりません。に、連続スキャン・モードの場合図 10.2 に示します。

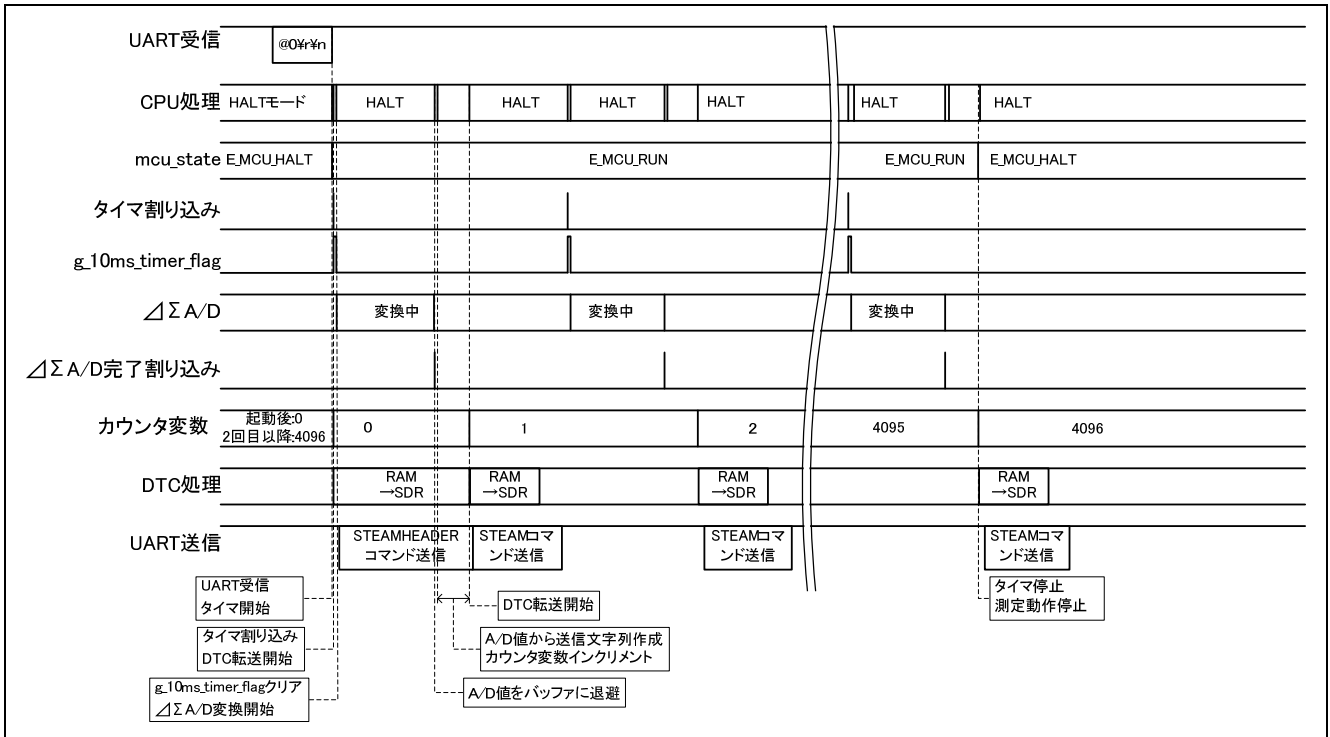


図 10.1 シングルスキャンの場合

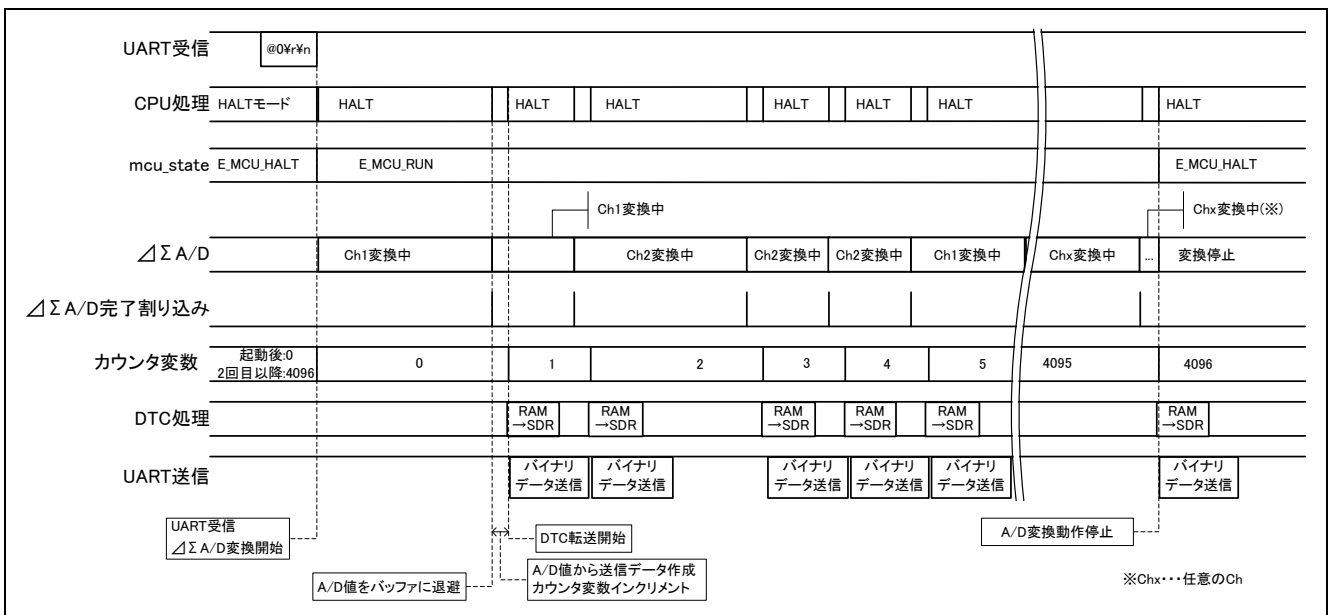


図 10.2 連続スキャンの場合 (Ch1 を 2 回、Ch2 を 3 回)

10.2 通信仕様

本サンプルコードは PC との通信を行い、動作の開始/停止や測定結果の送信を行っています。通信には UART を用います。通信仕様を以下に示します。

本サンプルコードでの UART 通信のデフォルト設定を表 10-1 に示します。

表 10-1 UART 通信設定

ボーレート	1Mbps
データ長	8 ビット
パリティ・ビット	なし
ストップ・ビット	1 ビット
オーダー	LSB ファースト

10.2.1 通信シーケンス

本サンプルコードの通信シーケンスを図 10.3 に示します。

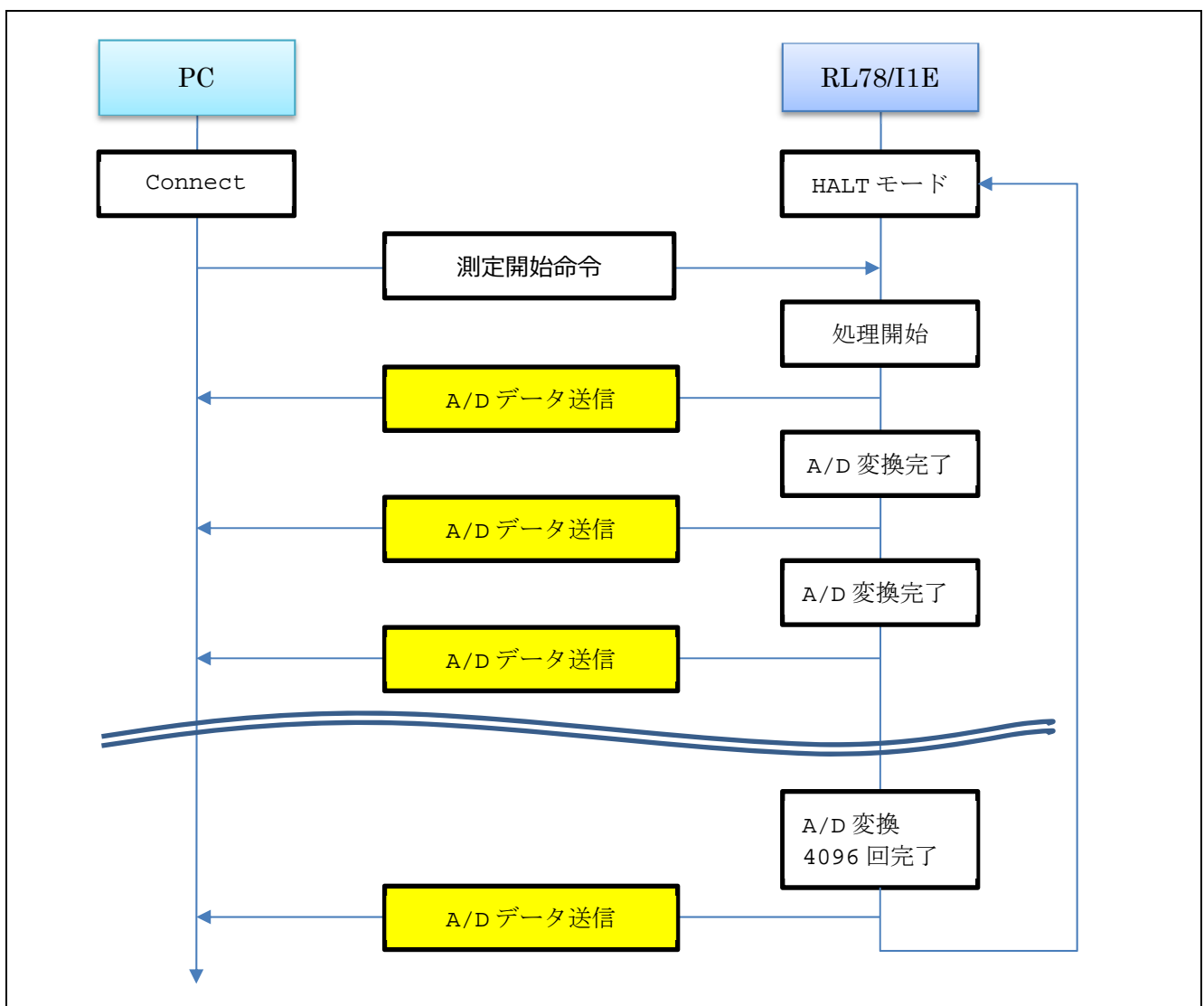


図 10.3 通信シーケンス

本サンプルコードは MCU から PC へ送信するデータを `r_communication_data_generation` 関数で生成しています。この関数は通信仕様のフォーマットとして「STREAM 転送」、「BULK 転送」、「バイナリ転送」を備えています。以降、これらのフォーマットの説明を行います。

なお、本サンプルコードでは「STREAM 転送」と「バイナリ転送」のみを使用しています。

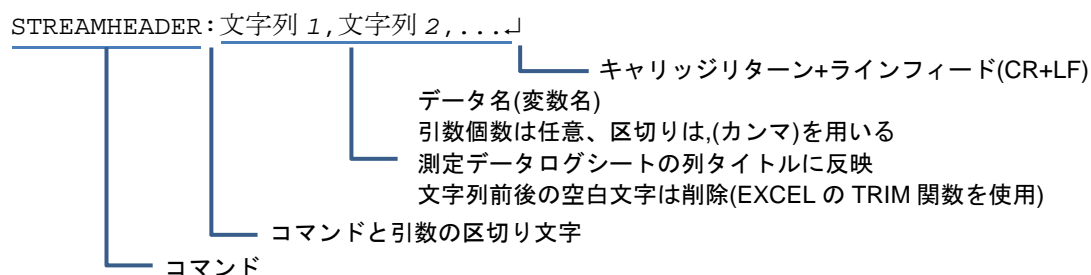
10.2.2 STREAM 転送

STREAM 転送は MCU が周期的に測定した結果を常に送りたい場合に用いられます。本ツールでは受信次第即座に表示に反映されます。

STREAM 転送では ASCII コードを用いており、A/D 変換結果などの数値データは `sprintf` 等により文字列に変換された形式で送られます。STREAM 転送には「STREAMHEADER」コマンドと「STREAM」コマンドが用いられます。

(1) 「STREAMHEADER」コマンド

「STREAMHEADER」は PC 側に送信するデータ数やデータの名前を通知するコマンドです。



データ名はカンマを区切り文字として複数指定可能です。ここで指定したデータ名が PC ツールのデータヘッダとして記録されます。

※ファイル「`r_cg_main.c`」の `define` 宣言「`D_UART_SEND_HEADER`」を修正してください。

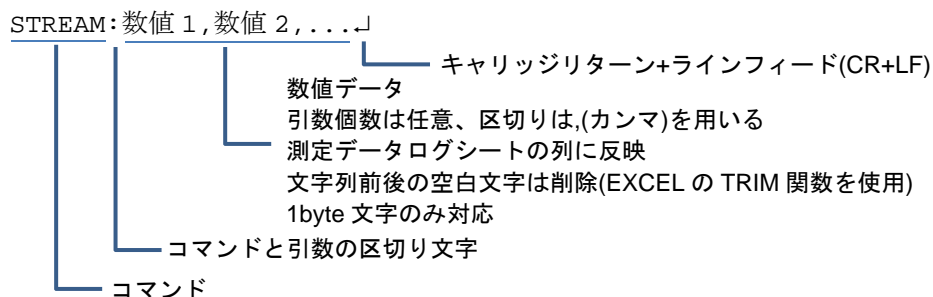
※各区切り文字前後にスペースは挿入しないでください。

例)

```
STREAMHEADER:count,gain_set1,gain_set2,offset,rawdata,correct,error
STREAMHEADER:Count,Filter,Gram,Stability,Status
```

(2) 「STREAM」コマンド受信

「STREAM」は測定データを ASCII ベースの数値として送信するコマンドです。



データはカンマを区切り文字として複数指定可能です。ヘッダで指定したデータ名に対応したデータを記入してください。

PC ツールの設定により、送信データ形式を DEC/HEX に切り替えることが可能です。ただし、DEC/HEX 形式の混在はできません。

※各区切り文字前後にスペースは挿入しないで下さい。

例)

```
STREAM:0,1,8,16,363278,45409,0
```

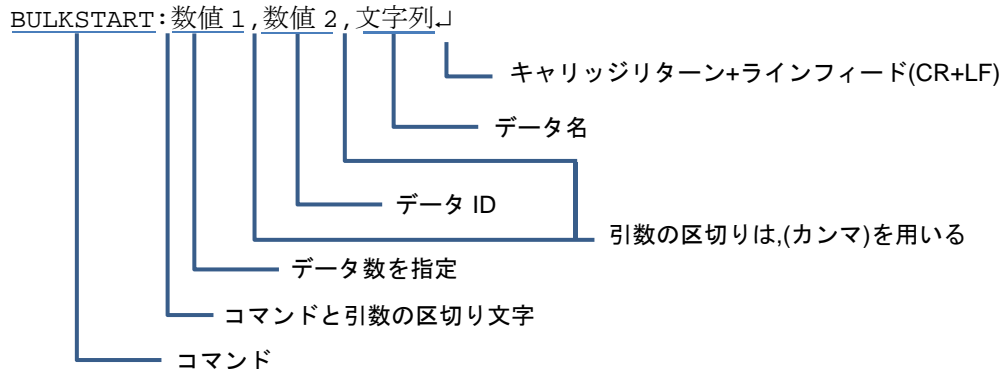
10.2.3 BULK 転送

BULK 転送は一定周期毎にデータを送信するのではなく、あるまとまった一塊のデータを送信したい場合に用います。

BULK 転送には「BULKSTART」コマンド、「BULK」コマンドと「BULKEND」コマンドを1セットとして用います。

(1) 「BULKSTART」コマンド

「BULKSTART」コマンドは、PC に以降に送信する「BULK」コマンドの送信データ数、データ種類を通知するためのコマンドです。



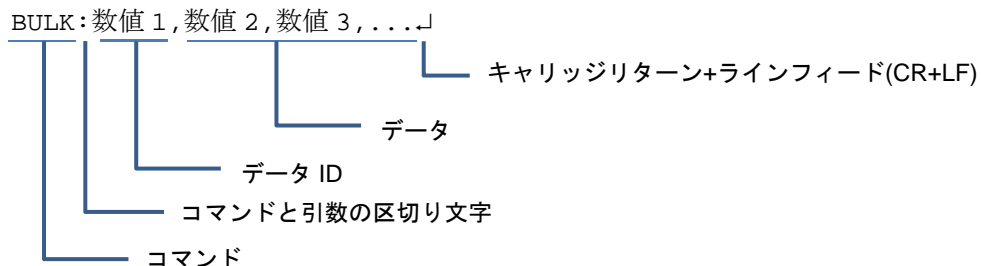
例)

```
BULKSTART:100,0,RAWDATA
```

(2) 「BULK」コマンド

「BULK」コマンドは、測定データを ASCII ベースの数値として送信するコマンドです。

「BULKSTART」で指定したデータ ID と異なる値を送ると、PC アプリはそれまでに受信したデータを破棄します。

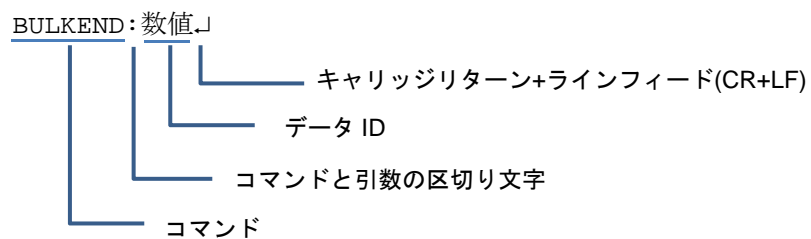


例)

```
BULK:0,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
BULK:0,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39
BULK:0,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59
BULK:0,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79
BULK:0,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99
```

「BULKEND」コマンド

「BULKEND」コマンドを送信により、PCにBULK転送の停止を通知します。



例)

BULKEND:0

10.2.4 バイナリ転送

連続スキャンモード選択時には 4byte のバイナリデータを PC に送信します。

以下に各フォーマットを示します。

① 送信回数カウンタ 1byte	② 24bit A/D 変換値 3byte
---------------------	--------------------------

① 送信回数カウンタ

1byte の送信回数カウンタです。

送信データごとにインクリメントされ、255 を超えると 0 に戻ります。

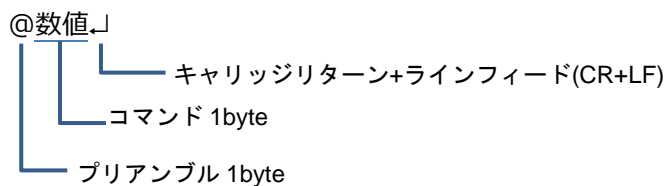
② 24bit A/D 変換値

24bit の $\Delta \Sigma$ A/D 変換値です。PC ツールでは差動入力モードを想定しています。

10.2.5 PC からのコマンド

PC から MCU へ指示を出す場合に送るコマンド仕様です。MCU はリソースが限られているため、非常にシンプルな構文(4Byte 固定)としています。

以下に各フォーマットを示します。



本サンプルコードでは、@0↓を受信する事により RL78/I1E ボード上の SW 長押しと同等の処理(A/D 変換処理の開始/終了)を行います。

11. 設定変更手順

本サンプルコードの動作を変更するには、CS+のコード生成を使用したハードウェアリソースの変更、および本サンプルコードのユーザー定義宣言を変更することで対応可能です。以下に変更手順を示します。

(1) メイン・システム・クロックの変更

- コード生成->クロック設定->メイン・システム・クロック(fMAIN)ソースの設定を変更する。
- D_MCU_CLOCK_MHZ 定義を変更する。(表 4-4 参照)

(2) SBIAS 出力電圧の変更

コード生成->PGA+ Δ Σ A/D コンバータ->SBIAS 出力電圧を変更する。

(3) PGA+ Δ Σ A/D コンバータの変更

① 24 ビット Δ Σ A/D コンバータの動作クロック変更

コード生成->クロック設定->24 ビット Δ Σ A/D コンバータの動作クロック(fDSAD)ソースの設定を変更する。

② Δ Σ A/D コンバータ動作モード設定変更

コード生成->PGA+ Δ Σ A/D コンバータ-> Δ Σ A/D コンバータ動作モード設定を変更する。

③ オートスキャンモード設定変更

コード生成->PGA+ Δ Σ A/D コンバータ->オートスキャンのモード設定を変更する。

④ 使用マルチプレクサ設定変更

コード生成->PGA+ Δ Σ A/D コンバータ->入力マルチプレクサ設定のチェックを変更する。

【注意】 サンプルコードでは 1ch のみの対応ですが、API としては複数 ch に対応しています。

⑤ マルチプレクサ毎の変更

(a) 入力モード変更

コード生成->PGA+ Δ Σ A/D コンバータ->使用マルチプレクサ設定の入力モードを変更する。

(b) PGA ゲイン設定変更

- コード生成->PGA+ Δ Σ A/D コンバータ->マルチプレクサ n->GSET1、GSET2 設定を変更する。
- D_DSAD_AUTO_GAIN_USE 定義を PGA 自動ゲイン調整無効に設定する。(表 4-4 参照)

(c) PGA ゲイン設定の自動調整設定(GSET1 自動設定)

- コード生成->PGA+ Δ Σ A/D コンバータ->マルチプレクサ n->GSET1、GSET2 設定を変更する
- D_DSAD_AUTO_GAIN_USE 定義を PGA 自動ゲイン調整有効に設定する。(表 4-4 参照)
- D_DSAD_AUTO_GAIN_TRIGGER_SEC 定義を変更する。(表 4-4 参照)

(d) PGA オフセット調整電圧設定変更

コード生成->PGA+ Δ Σ A/D コンバータ->マルチプレクサ n->オフセット調整電圧設定を変更する。

(e) オーバ・サンプリング比の変更

コード生成->PGA+ Δ Σ A/D コンバータ->マルチプレクサ n->オーバ・サンプリング比を変更する。

(f) Δ Σ A/D コンバータ、A/D 変換回数設定

コード生成->PGA+ Δ Σ A/D コンバータ->マルチプレクサ n->A/D 変換回数を変更する。

- (g) $\Delta \Sigma$ A/D コンバータ、平均化処理の変更
 コード生成->PGA+ $\Delta \Sigma$ A/D コンバータ->マルチプレクサ n->平均化処理、平均化データ数
 を変更する。

(4) A/D 変換周期の変更

- ① オートスキャンのモード選択->連続スキャン・モード時
 OSR 設定変更、 $\Delta \Sigma$ A/D コンバータ動作モード設定変更参照
- ② オートスキャンのモード選択->シングル・スキャン・モード設定時
 コード生成->タイマ・アレイ・ユニット 1->チャンネル 0->インターバル時間を変更する。
【注意】本サンプルコードでは、シングル・スキャン・モードの A/D 開始タイミングと KEY スキャン
 タイミングで共通のタイマを用いています。

(5) DSAD 変換値格納バッファサイズの変更

D_DSAD_VALUE_BUFFER_SIZE 定義を変更する。(表 4-4 参照)

(6) シリアル通信設定の変更

- ① UART 送信有効/無効
 D_UART_SEND_USE 定義を変更する。(表 4-4 参照)
- ② UART 通信ボー・レート変更
- コード生成->シリアル・アレイ・ユニット->シリアル・アレイ・ユニット 0->UART1->ボー・
 レートを変更する。
 - RL78I1E_Analog_Characteristics_Evaluation.xlsm の Serial Communication Settings->Baudrate:を
 変更する。
- ③ UART を ch0 に変更
- コード生成->シリアル・アレイ・ユニット->シリアル・アレイ・ユニット 0->チャンネル 1 の
 プルダウンメニューから UART0 を選択。
 - UART0 に、UART1 に設定していた内容を写す。
 - r_cg_sau_user.c の r_uart0_callback_receiveend 関数に以下の処理を追加
`g_rx_in_process_flag = 0U;`
 - r_cg_sau_user.c の r_uart0_callback_sendend 関数に以下の処理を追加
`g_tx_in_process_flag = 0U;`
 - コード生成->データ・トランスファ・コントローラ->DTC 設定 起動要因に UART0 送信と
 UART0 受信に変更する。
 - コード生成->データ・トランスファ・コントローラ->DTCD0->転送元 0xFF46 から 0xFF12
 - コード生成->データ・トランスファ・コントローラ->DTCD1->転送先 0xFF44 から 0xFF10
 - r_cg_dtc.c の R_DTC0_Init 関数の内容を修正
`dtc_controldata_0.dtsar = _FF12_DTCD0_SRC_ADDRESS;`
 - r_cg_dtc.c の R_DTC1_Init 関数の内容を修正
`dtc_controldata_1.dtdar = _FF10_DTCD1_DEST_ADDRESS;`
 - r_cg_main.c にある R_UART1~関数を R_UART0~関数に置き換える
- ④ UART 送信バッファサイズ変更
 D_UART_SEND_BUFFER_SIZE 定義を変更変更する。(表 4-4 参照)

(7) 通信データの変更

- ① STREAM データ変更
- D_STREAMHEADER 定義を変更する。(表 4-3 参照)
 - r_communication_data_generation 関数を変更する。(10.2.2 章参照)

- ② BULK データ変更
 - `r_communication_data_generation` 関数を変更する。(10.2.3 章参照)
 - ③ BULKSTART コマンドから BULKEND コマンドまでの転送データの総数変更
`D_BULK_NUM` 定義を変更する。(表 4-3 参照)
 - ④ 1 回の BULK コマンド毎の転送データ数変更
`D_BULK_COMMAND_NUM` 定義を変更する。(表 4-3 参照)
- (8) フラッシュ・メモリ
- ① フラッシュ・メモリを使用する/しない変更
`D_FLASH_MEMORY_DATA_USE` 定義を変更する。(表 4-4 参照)
 - ② フラッシュ・メモリに格納するデータの変更
`str_flash_data_t` 定義を変更する。(表 4-36 参照)
 - ③ `R_MAIN_UserInit` 関数内でフラッシュ・メモリに値を書き込む/書き込まない
 - `D_FLASH_FORCE_WRITING` 定義を変更する。(表 4-4 参照)
 - データ・フラッシュ格納データ構造体変数(`g_flash_value`)の初期値を変更する。
- (9) PGA 誤差測定の変更
- ① 誤差測定の実施する/しない
`D_DSAD_CORRECT_USE` 定義を変更する。(表 4-4 参照)
 - ② 誤差測定に用いる測定基準電圧変更
`D_GAIN_ERROR_REFERENCE_mV` 定義を変更する。(表 4-4 参照)
 - ③ 誤差測定する入力マルチプレクサの設定
`D_DSAD_CORRECT_MPXn` 定義を変更する。(表 4-4 参照)
- (10) 断線検知処理の変更
- ① 断線チェック回数の変更
`D_DISCONNECTION_CHECK_COUNT` 定義を変更する。(表 4-4 参照)
 - ② 断線判定電圧の変更
`D_DISCONNECTION_THRESHOLD_mV` 定義を変更する。(表 4-4 参照)
- (11) デバッグ用 LED の変更
- ① デバッグ用 LED 使用/不使用
`D_DEBUG_LED_USE` 定義を変更する。(表 4-6 参照)
 - ② デバッグ用 LED 駆動ポートの変更
`D_DEBUG_LED_PORT` 定義を変更する。(表 4-6 参照)
- (12) KEYSKAN の変更
- ① KEY ポート有効レベルの変更
`DEF_KEY_ACTIVE` 定義を変更する。(表 4-5 参照)
 - ② KEY ポート割り当て、処理割り当て、グループ情報変更
 - `e_key_t` 列挙体を変更する。(表 4-12 参照)
 - `g_con_key_setting` グローバル定数を変更する。(4.5.1 章参照)

③ KEY 判定時間の変更

- KEY_SCAN_NORM 定義を変更する。(表 4-5 参照)
- KEY_SCAN_LONG 定義を変更する。(表 4-5 参照)
- KEY_SCAN_DEAD 定義を変更する。(表 4-5 参照)
- KEY_SCAN_NOT 定義を変更する。(表 4-5 参照)
- KEY_SCAN_DOUBLE 定義を変更する。(表 4-5 参照)

④ KEY スキャンタイミングの変更

コード生成->タイマ・アレイ・ユニット 1->チャンネル 0->インターバル時間を変更する。

【注意】本サンプルコードでは、シングル・スキャン・モードの A/D 開始タイミングと KEY スキャンタイミングで共通のタイマを用いています。

12. API 使用時の注意事項

本章では、API を使用して RL78/I1E を制御する場合の制約、注意事項について記載します。本章の内容について必ずご理解頂いた上で API をご使用ください。

12.1 API 使用上の注意

12.1.1 校正データについて

本サンプルコードでは、校正データはデータフラッシュ・メモリに格納されます。何等かの理由により、誤った校正データがデータフラッシュ・メモリに保存された場合、校正データを消去する手段として次に示す方法を推奨します。

- ・ CS+のデバッグ・ツール設定を変更して校正データを消去する
 - ① デバッグ・ツールと切断した状態にする
 - ② CS+のプロジェクト・ツリーにある「RL78 E1(Serial) (デバッグ・ツール)」をダブルクリックする。
 - ③ プロパティのタブ「接続用設定」を表示し、「フラッシュ」の「起動時にフラッシュ ROM を消去する」を「はい」にする。(図 12.1 CS+デバッグ・ツール設定画面 参照)
 - ④ 画面上部「デバッグ」メニューより、「デバッグ・ツールヘダウンロードする」を選択する。

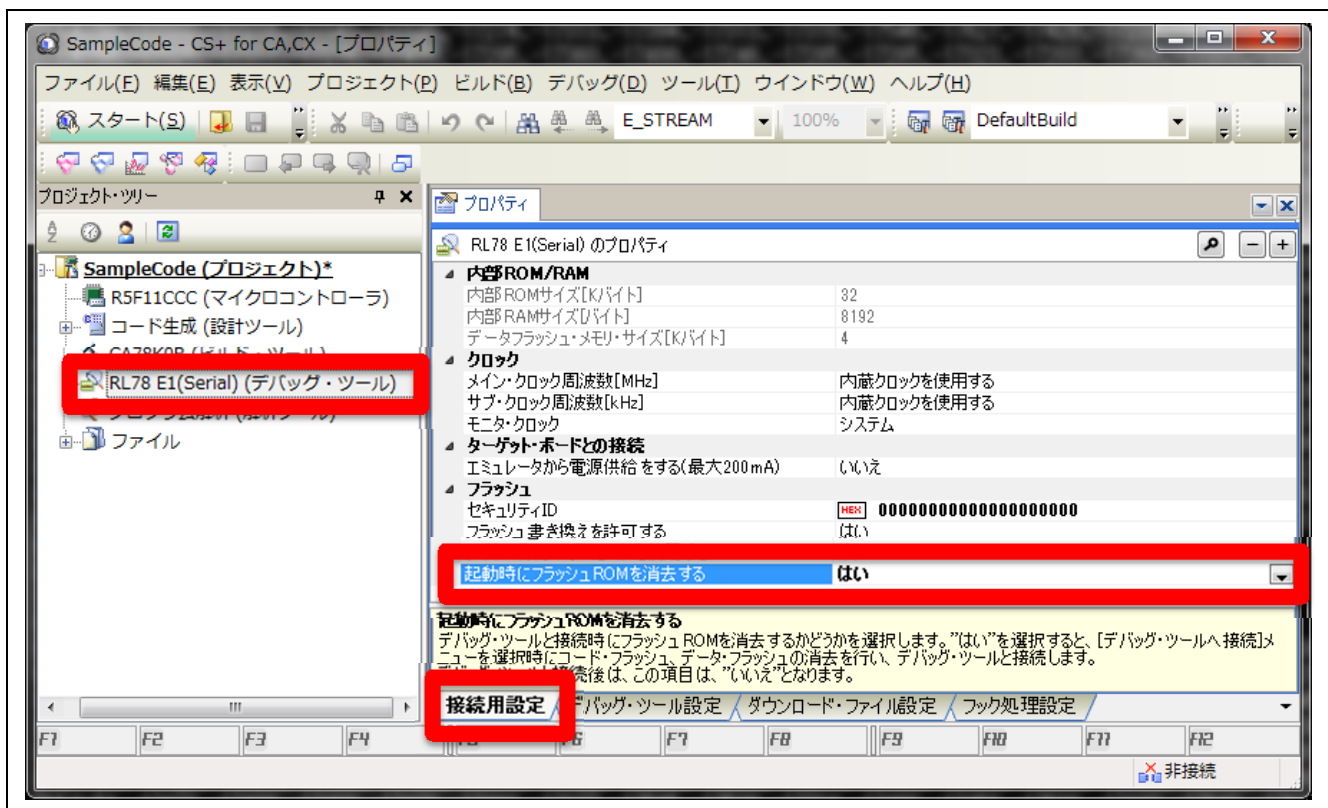


図 12.1 CS+デバッグ・ツール設定画面

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	2015.11.09	---	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っていると、内部ROM、レイアウトパターンの相違などにより、電气的特性の範囲で、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て、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 〒100-0004 千代田区大手町2-6-2（日本ビル）

■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/>