

RZ/A1H グループ

ADC サンプルプログラム

R01AN2083JJ0100
Rev.1.00
2014.6.24

要旨

本アプリケーションノートでは、RZ/A1H の A/D 変換器を使用しキー入力を A/D 変換するサンプルプログラムについて説明します。また、以後 A/D 変換器は ADC と表記します。

ADC サンプルプログラムの特長を以下に示します。

- ・チャンネル AN0 に接続されたキー入力を A/D 変換し、ターミナルにキー番号を表示します。
- ・A/D 変換結果の読み込みタイミングは A/D 変換終了割り込みで検出しています。

対象デバイス

RZ/A1H

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	3
2. 動作確認条件	4
3. 関連アプリケーションノート	5
4. 周辺機能説明	6
5. ハードウェア説明	7
5.1 ハードウェア構成例	7
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 動作説明	10
6.1.2 使用準備	11
6.2 コマンド一覧	12
6.3 コマンド詳細	13
6.3.1 ADC	13
6.4 メモリマップ	14
6.4.1 サンプルプログラムのセクション配置	15
6.4.2 MMU の設定	18
6.4.3 例外処理ベクタテーブル	19
6.5 使用割り込み一覧	20
6.6 固定幅整数一覧	20
6.7 定数/エラーコード一覧	21
6.8 構造体/共用体一覧	23
6.9 変数一覧	24
6.10 関数一覧	24
6.11 関数仕様	25
6.11.1 R_ADC_SampleMain	25
6.11.2 AdcInitialize	25
6.11.3 AdcStart	25
6.11.4 AdcReadSingle	26
6.11.5 AdcStop	26
6.11.6 AdcUninitialize	26
6.11.7 ReadKey1To4	27
6.11.8 IntAdcEnd	27
6.11.9 Delay10ms	27
6.12 フローチャート	28
6.12.1 メイン処理	28
6.12.2 キー変換結果の取得処理	29
7. サンプルコード	31
8. 参考ドキュメント	31

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
ADC Ch(AN0)	キー入力の A/D 変換
INTC(割り込み ID : 170)	A/D 変換終了割り込み
SCIF(UART) Ch(2)	デバッグ情報表示、キー番号表示

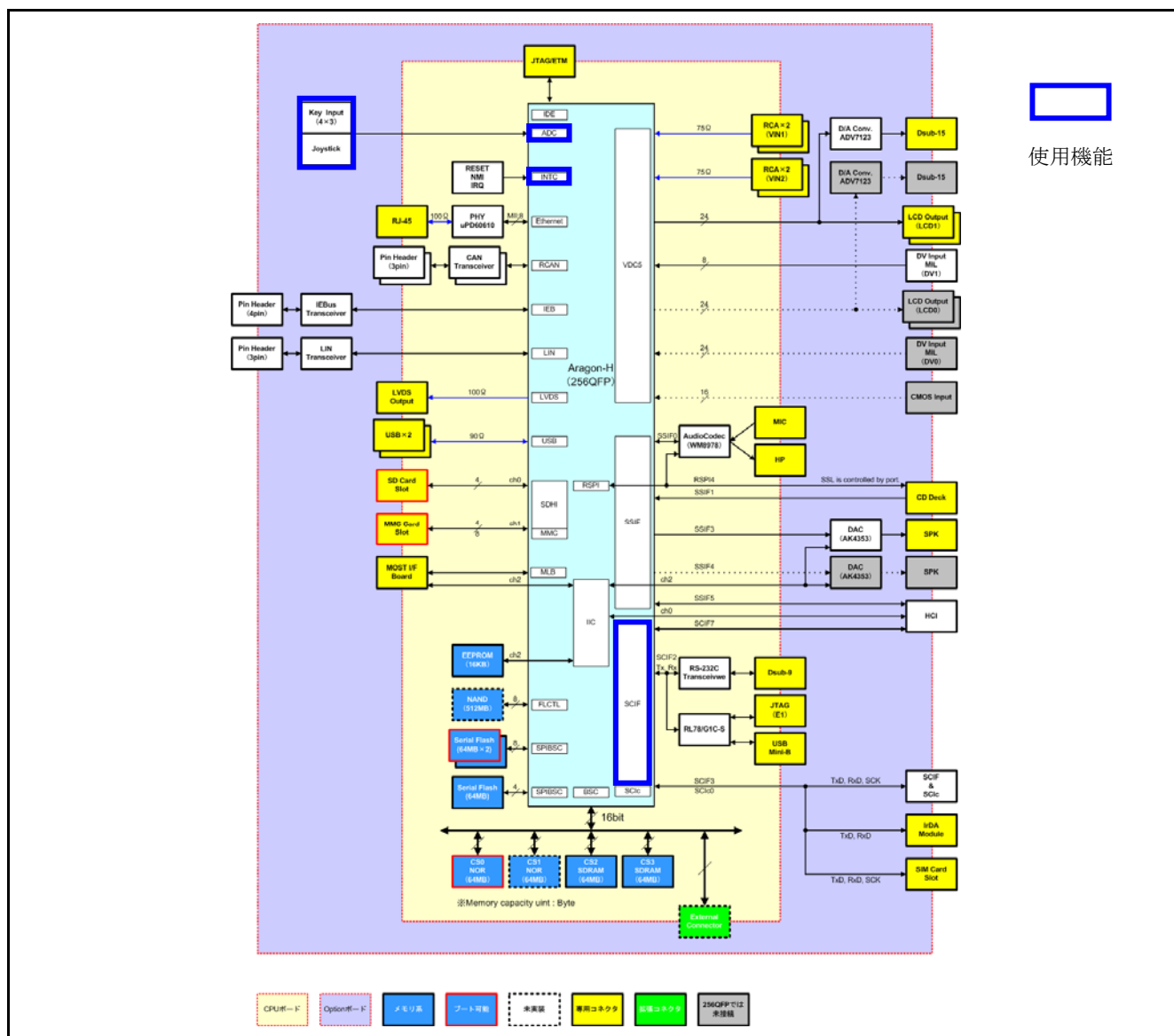


図 1.1 動作環境

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	RZ/A1H
動作周波数	CPU クロック (I ϕ) : 400MHz 画像処理クロック (G ϕ) : 266.67MHz 内部バスクロック (B ϕ) : 133.33MHz 周辺クロック 1 (P1 ϕ) : 66.67MHz 周辺クロック 0 (P0 ϕ) : 33.33MHz
動作電圧	電源電圧 (I/O) : 3.3V 電源電圧 (内部) : 1.18V
統合開発環境	ARM [®] 統合開発環境 ARM Development Studio 5 (DS-5 [™]) Version 5.16
C コンパイラ	ARM C/C++ Compiler/Linker/Assembler Ver.5.03 [Build 102] コンパイラオプション(ディレクトリパスの追加は除く) -O3 -Ospace --cpu=Cortex-A9 --littleend --arm --apcs=/interwork --no_unaligned_access --fpu=vfpv3_fp16 -g --asm
動作モード	ブートモード 0 (CS0 空間 16 ビットブート)
ターミナルソフトの通信設定	・ 通信速度 : 115200bps ・ データ長 : 8 ビット ・ パリティ : なし ・ ストップビット長 : 1 ビット ・ フロー制御 : なし
使用ボード	GENMAI ボード ・ R7S72100 CPU ボード RTK772100BC00000BR ・ RTS72100 CPU ボード用オプションボード RTK7721000B00000BR
使用デバイス (ボード上で使用する機能)	・ シリアルインタフェース (Dsub-9 コネクタ) ・ キー入力スイッチ (SW1~4) 使用(オプションボード)

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/A1H グループ 初期設定例 (R01AN1646JJ0100)
- RZ/A1H グループ レジスタ定義ヘッダ・ファイル iodef.h (R01AN1860JJ0100)

4. 周辺機能説明

ADC についての基本的な内容は、RZ/A1H グループ・ユーザズマニュアル ハードウェア編に記載しています。

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。

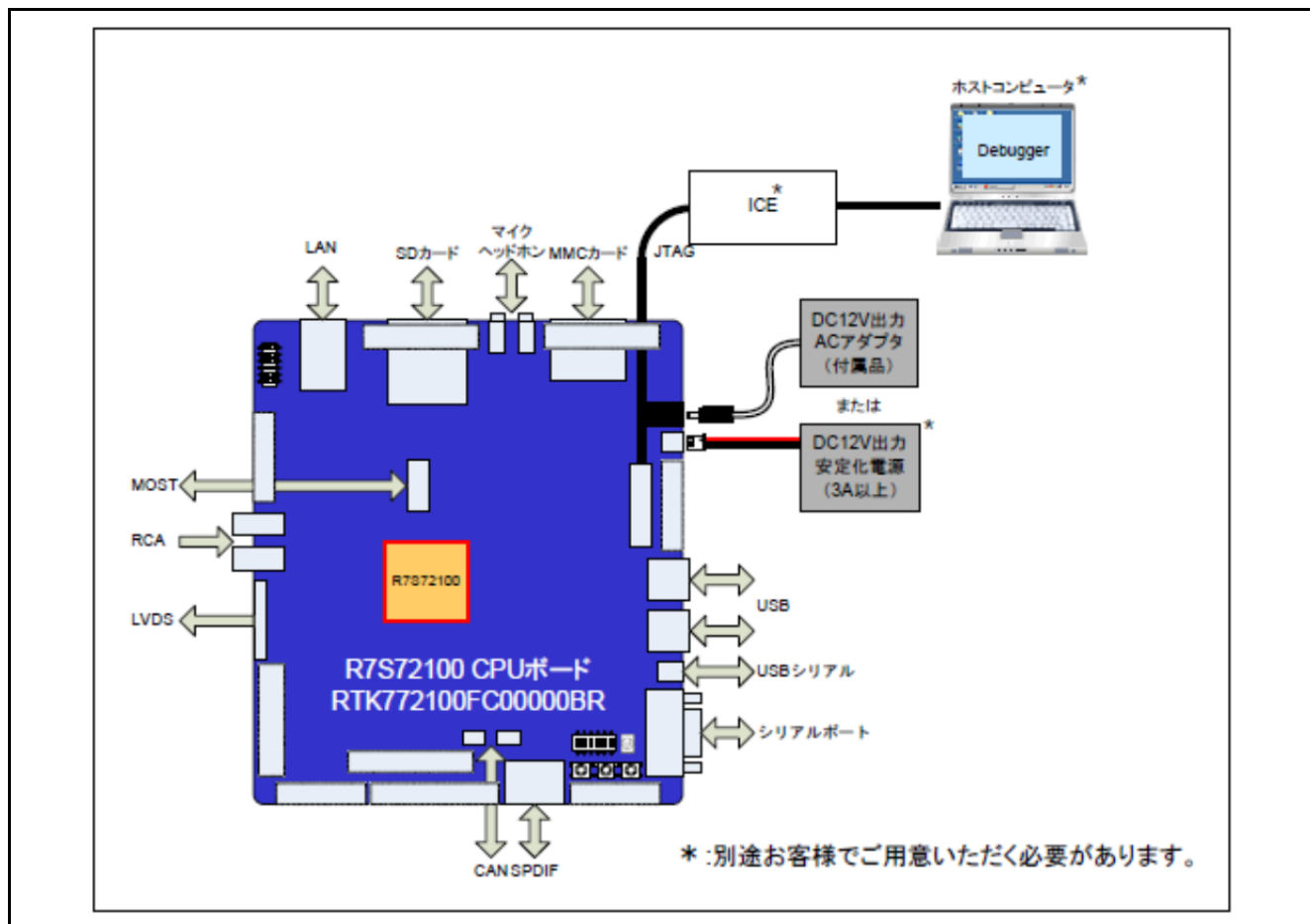


図 5.1 ハードウェア構成例

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
P1_8 : AN0	入力	ADC の入力端子。
P3_0 : TxD2	出力	FIFO 内蔵シリアルコミュニケーションインタフェースのデータ出力端子。
P3_2 : RxD2	入力	FIFO 内蔵シリアルコミュニケーションインタフェースのデータ入力端子。

6. ソフトウェア説明

6.1 動作概要

オプションボード上のキー入力スイッチ SW1~4 を押下した時の電圧値を A/D 変換しキーを識別します。

図 6.1 にシステムブロック図を示します。

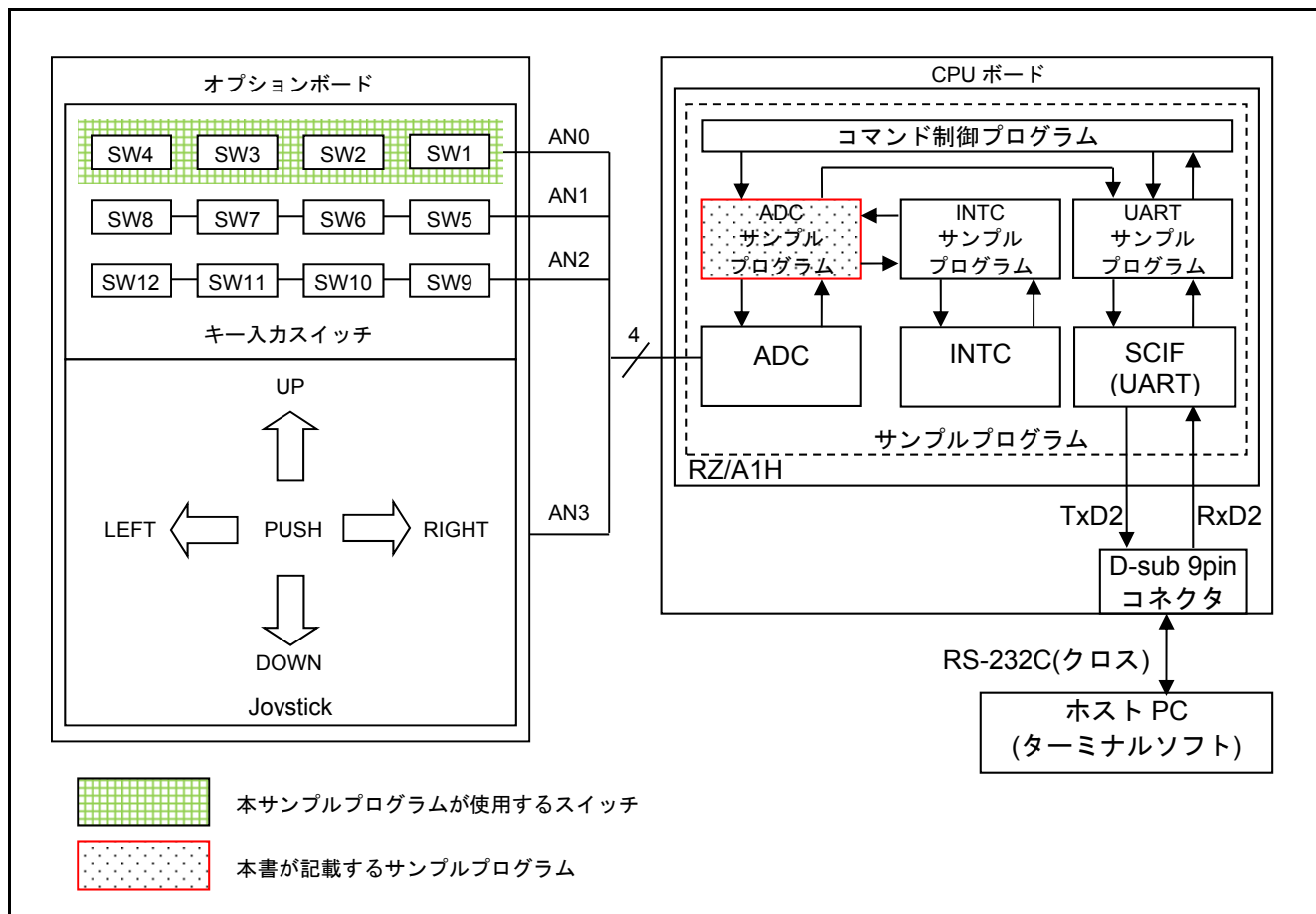


図 6.1 システムブロック図

6.1.1 動作説明

キーを押した時にキー番号”1”~”4”を、離れた時に”0”を UART 送信します。キーを長押しした場合は、押されたキー番号を一回だけ UART 送信します。また、押されたキーが SW4 の場合は ADC サンプルプログラムを終了します。図 6.2 に SW1 が押された場合の動作概要を示します。

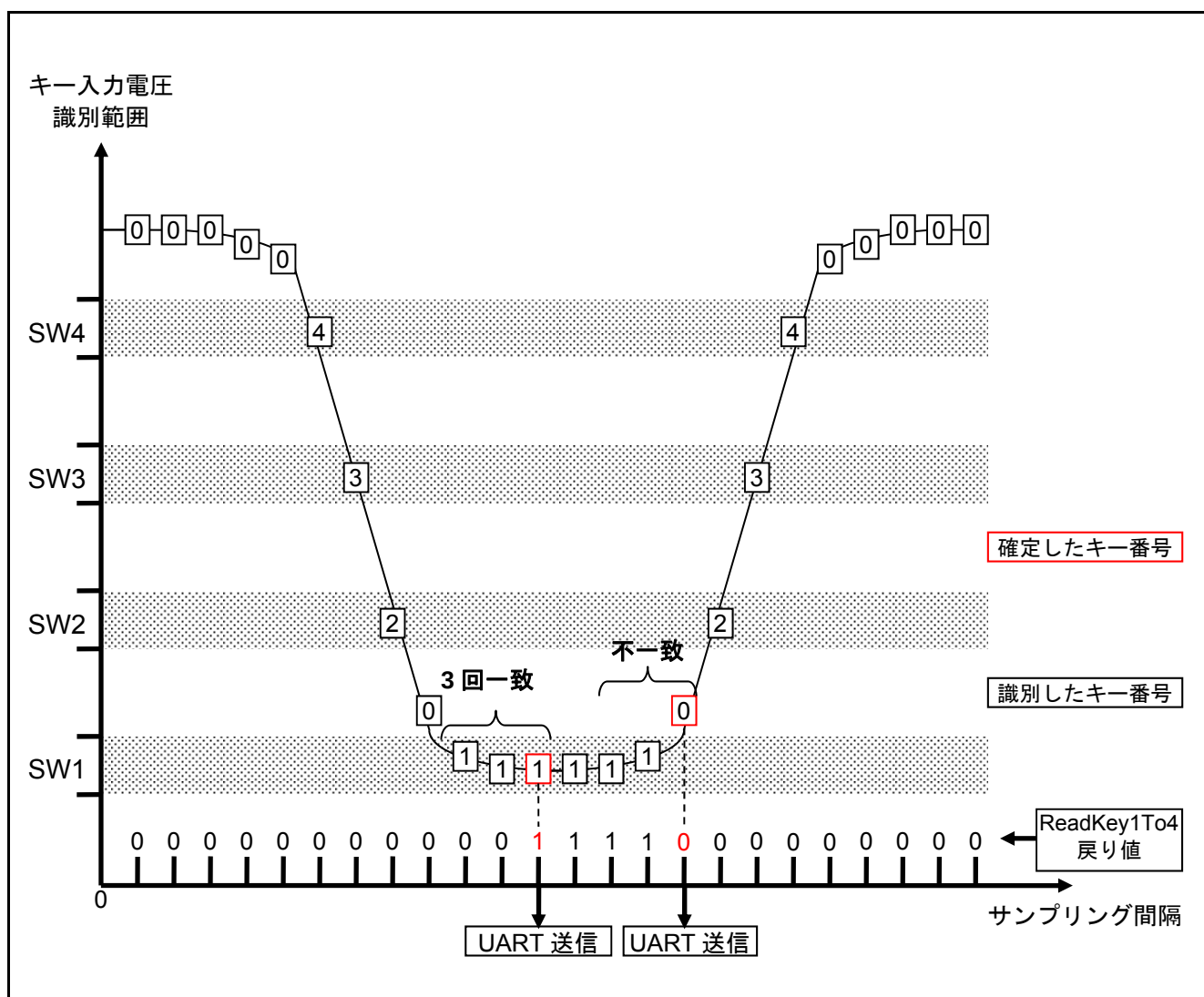


図 6.2 SW1 の入力電圧に対する ADC サンプルプログラム動作概要

6.1.2 使用準備

本サンプルプログラムの実行準備を説明します。

- 1) ホスト PC にてターミナルソフトを起動し、次のように設定します。(Tera Term の場合)



図 6.3 シリアルポートの設定

- 2) サンプルプログラムを実行すると、次のようにターミナルでコマンド待ちになります。次章の説明にあるコマンドを入力することでサンプルプログラムの各動作を実行させることが可能となります。

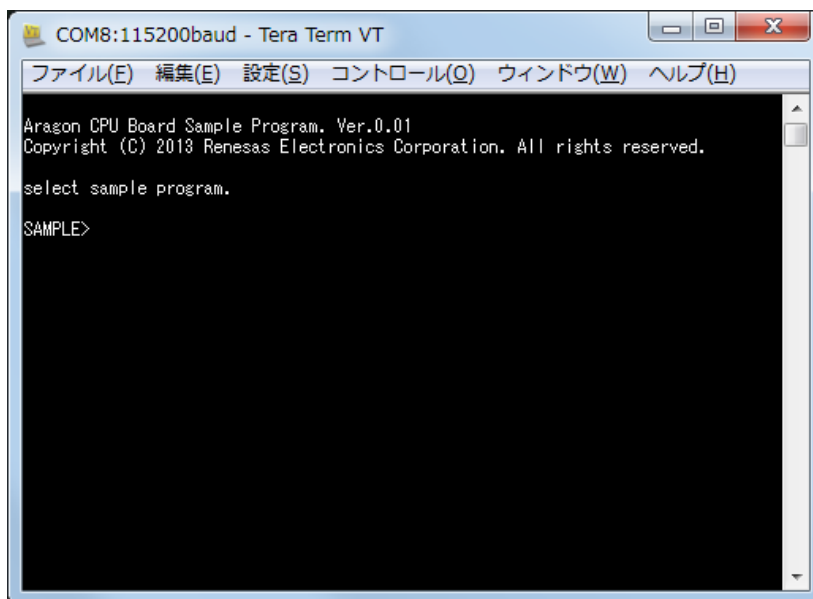


図 6.4 サンプルプログラムモジュール選択のコマンド待ち

6.2 コマンド一覧

サンプルプログラムで実装したコマンド一覧を示します。なおコマンドは大文字・小文字を区別しません。

表 6.1 コマンド一覧

コマンド	概要
ADC	ADC サンプルプログラムを起動します。
HELP	ヘルプ

6.3 コマンド詳細

サンプルプログラムのコマンド仕様を示します。

6.3.1 ADC

ADC

概 要	ADC サンプルプログラムを起動します。
入 力	ADC
説 明	ADC コマンドを入力後に、オプションボード上の押下されたキー入力スイッチ SW1～SW4 を識別し、キー番号をターミナルに表示します。
パラメータ	なし

6.4 メモリマップ

図 6.5 に、RZ/A1H グループのアドレス空間と GENMAI ボードのメモリマッピングを示します。

参考プログラムでは、ROM 領域を使用するコードおよびデータを CS0 空間に接続した NOR フラッシュメモリに配置し、RAM 領域を使用するコードおよびデータを大容量内蔵 RAM に配置するようにしています。

		RZ/A1H グループの アドレス空間	GENMAI ボードメモ リマッピング
ミラー空間	H'FFFF FFFF	その他 (2550MB)	その他 (2550MB)
	H'60A0 0000	大容量内蔵 RAM (10MB)	大容量内蔵 RAM ミラー空間
	H'6000 0000	SPI マルチ I/O バス 空間 2 (64MB)	SPI マルチ I/O バス ミラー空間 2
	H'5C00 0000	SPI マルチ I/O バス 空間 1 (64MB)	SPI マルチ I/O バス ミラー空間 1
	H'5800 0000	CS5 空間 (64MB) CS4 空間 (64MB)	CS5 ミラー空間 CS4 ミラー空間
	H'5000 0000	CS3 空間 (64MB)	CS3 ミラー空間
	H'4C00 0000	CS2 空間 (64MB)	CS2 ミラー空間
	H'4800 0000	CS1 空間 (64MB)	CS1 ミラー空間
	H'4400 0000	CS0 空間 (64MB)	CS0 ミラー空間
	H'4000 0000	その他 (502MB)	その他 (502MB)
	H'20A0 0000	大容量内蔵 RAM (10MB)	大容量内蔵 RAM (10MB)
	H'2000 0000	SPI マルチ I/O バス 空間 2 (64MB)	シリアルフラッシュ メモリ (64MB)
	H'1C00 0000	SPI マルチ I/O バス 空間 1 (64MB)	シリアルフラッシュ メモリ (64MB)
	H'1800 0000	CS5 空間 (64MB) CS4 空間 (64MB)	ユーザ領域
通常空間	H'1000 0000	CS3 空間 (64MB)	SDRAM (64MB)
	H'0C00 0000	CS2 空間 (64MB)	SDRAM (64MB)
	H'0800 0000	CS1 空間 (64MB)	NOR フラッシュ メモリ (64MB)
	H'0400 0000	CS0 空間 (64MB)	NOR フラッシュ メモリ (64MB)
	H'0000 0000		

図 6.5 メモリマップ

6.4.1 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は、割り込み処理の高速化のため、例外処理ベクタテーブルと IRQ 割り込みハンドラを大容量内蔵 RAM 上に配置して、これらの処理を大容量内蔵 RAM 上で実行するようにしています。例外処理ベクタテーブルおよび IRQ 割り込みハンドラのプログラムコードの NOR フラッシュメモリ領域から大容量内蔵 RAM 領域への転送処理、初期値なしデータセクションのゼロクリア処理、および初期値ありデータセクションの初期化処理は、スキッタローディング機能を使用して行っています。スキッタローディング機能の詳細は、ARM より提供される「ARM コンパイラツールチェーン／リンクの使用／イメージの構造と生成の章」を参照してください。

表 6.2 に参考プログラムで使用するセクションを示し、図 6.6 に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）と、スキッタローディング機能を使用後のセクション配置（実行ビュー）を示します。

表 6.2 使用するセクション

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_TABLE	例外処理ベクタテーブル	Code	FLASH	FLASH
RESET_HANDLER	リセットハンドラ処理のプログラムコード領域 この領域は以下のセクションから構成されています ・ INITCA9CACHE (L1 キャッシュ設定) ・ INIT_TTB (MMU 設定) ・ RESET_HANDLER (リセットハンドラ)	Code	FLASH	FLASH
CODE_BASIC_SETUP	動作周波数とフラッシュメモリ最適化のためのプログラムコード領域	Code	FLASH	FLASH
InRoot	この領域は C 標準ライブラリなどのルート領域に配置するセクションから構成されています	Code および RO Data	FLASH	FLASH
CODE_FPU_INIT	NEON および VFP 初期設定のプログラムコード領域 この領域は以下のセクションから構成されています ・ CODE_FPU_INIT ・ FPU_INIT	Code	FLASH	FLASH
CODE_RESET	ハードウェア初期設定のプログラムコード領域 この領域は以下のセクションから構成されています ・ CODE_RESET (スタートアップ処理) ・ INIT_VBAR (ベクタベース設定)	Code	FLASH	FLASH
CODE_IO_REGRW	IO レジスタのリード/ライト関数のプログラムコード領域	Code	FLASH	FLASH
CODE	デフォルトのプログラムコード領域 C ソースでセクション名を定義しない Code タイプのセクションは、すべてこの領域に配置されます	Code	FLASH	FLASH
CONST	デフォルトの定数データ領域 C ソースでセクション名を定義しない RO Data タイプのセクションは、すべてこの領域に配置されます	RO Data	FLASH	FLASH

領域の名前	内容	タイプ	ロード領域	実行領域
VECTOR_MIRROR_TABLE	例外処理ベクタテーブル (大容量内蔵 RAM に転送して実行するためのセクション)	Code	FLASH	LRAM
CODE_HANDLER_JMPTBL	IRQ 割り込みハンドラのユーザ定義関数のプログラムコード領域	Code	FLASH	LRAM
CODE_HANDLER	IRQ 割り込みハンドラのプログラムコード領域 この領域は以下のセクションから構成されています ・ CODE_HANDLER ・ IRQ_FIQ_HANDLER	Code	FLASH	LRAM
DATA_HANDLER_JMPTBL	IRQ 割り込みハンドラのユーザ定義関数の登録テーブルデータ領域	RW Data	FLASH	LRAM
ARM_LIB_STACK	アプリケーションスタック領域	ZI Data	—	LRAM
IRQ_STACK	IRQ モードのスタック領域	ZI Data	—	LRAM
FIQ_STACK	FIQ モードのスタック領域	ZI Data	—	LRAM
SVC_STACK	スーパーバイザ (SVC) モードのスタック領域	ZI Data	—	LRAM
ABT_STACK	アボート (ABT) モードのスタック領域	ZI Data	—	LRAM
TTB	MMU 変換テーブル領域	ZI Data	—	LRAM
ARM_LIB_HEAP	アプリケーションヒープ領域	ZI Data	—	LRAM
DATA	デフォルトの初期値ありデータ領域 C ソースでセクション名を定義しない RW Data タイプのセクションは、すべてこの領域に配置されます	RW Data	FLASH	LRAM
BSS	デフォルトの初期値なしデータ領域 C ソースでセクション名を定義しない ZI Data タイプのセクションは、すべてこの領域に配置されます	ZI Data	—	LRAM

- 【注】 1. 表中のロード領域および実行領域において、FLASH は NOR フラッシュメモリの領域を、LRAM は大容量内蔵 RAM の領域を表します。
2. セクションの名前は基本的に領域と同じ名前にしていますが、RESET_HANDLER、InRoot、CODE_FPU_INIT、CODE_RESET、CODE、CONST、CODE_HANDLER、DATA、BSS の各領域は複数のセクションから構成されています。領域とセクションについては、ARM コンパイラツールチェーンのマニュアルを参照してください。

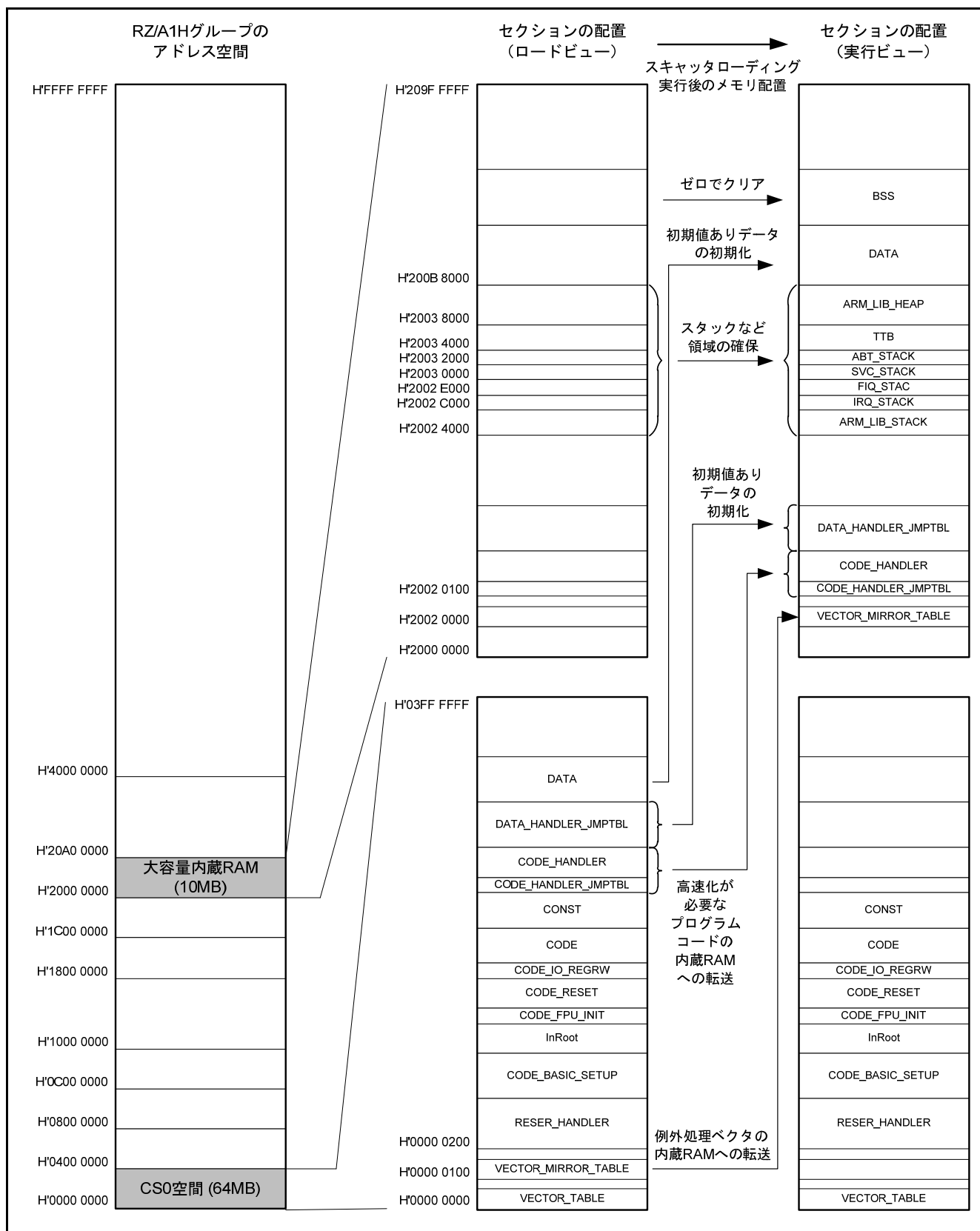


図 6.6 セクション配置

6.4.2 MMU の設定

RZ/A1H GENMAI ボードで使用するハードウェア資源のメモリマップにあわせて、H'0000 0000 番地から 1MB 単位で 4GB の領域を MMU で管理するように設定しています (ttb_init.s ファイルで設定しています)。システムにあわせてカスタマイズする場合は、最少単位を 1MB としてください。

表 6.3 に、サンプルプログラムの MMU の設定を示します。

表 6.3 MMU の設定

定義名	内容	アドレス	サイズ	メモリタイプ
M_SIZE_NOR	CS0、CS1 空間 (NOR フラッシュメモリ)	H'0000 0000 ～ H'07FF FFFF	128MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_SDRAM	CS2、CS3 空間 (SDRAM)	H'0800 0000 ～ H'0FFF FFFF	128MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_CS45	CS4、CS5 空間	H'1000 0000 ～ H'17FF FFFF	128MB	ストロングリオーダメモリ (L1 キャッシュ無効)
M_SIZE_SPI	SPI マルチ IO バス空間 1 および 2 (シリアルフラッシュメモリ)	H'1800 0000 ～ H'1FFF FFFF	128MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_RAM	大容量内蔵 RAM 空間	H'2000 0000 ～ H'209F FFFF	10MB	L1 キャッシュ有効、 ノーマルメモリ
M_SIZE_IO_1	内蔵周辺モジュールおよび 予約エリア	H'20A0 0000 ～ H'3FFF FFFF	502MB	ストロングリオーダメモリ (L1 キャッシュ無効)
M_SIZE_NOR_M	CS0、CS1 ミラー空間	H'4000 0000 ～ H'47FF FFFF	128MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_SDRAM_M	CS2、CS3 ミラー空間	H'4800 0000 ～ H'4FFF FFFF	128MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_CS45_M	CS4、CS5 ミラー空間	H'5000 0000 ～ H'57FF FFFF	128MB	ストロングリオーダメモリ (L1 キャッシュ無効)
M_SIZE_SPI_M	SPI マルチ IO バス ミラー空間 1 および 2	H'5800 0000 ～ H'5FFF FFFF	128MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_RAM_M	大容量内蔵 RAM ミラー空間	H'6000 0000 ～ H'609F FFFF	10MB	L1 キャッシュ無効、 ノーマルメモリ
M_SIZE_IO_2	内蔵周辺モジュールおよび 予約エリア	H'60A0 0000 ～ H'FFFF FFFF	2550MB	ストロングリオーダメモリ (L1 キャッシュ無効)

6.4.3 例外処理ベクタテーブル

RZ/A1H には 7 種類の例外処理（リセット、未定義命令、ソフトウェア割り込み、プリフェッチアボート、データアボート、IRQ、FIQ）があり、ブートモード 0 の場合、リセット解除後に例外処理のベクタテーブルは H'0000 0000 番地から 32 バイトの領域（H'0000 0000 番地～H'0000 001F 番地）に配置されます。例外処理のベクタテーブルには、各例外処理への分岐命令を記述します。

図 6.7 に例外処理ベクタテーブルの記述例として、サンプルコードの例外処理ベクタテーブル内容を示します。

```
vector_table
LDR pc, =reset_handler      ; 0x0000_0000 : Reset exception
LDR pc, =undefined_handler  ; 0x0000_0004 : Undefined instructions exception
LDR pc, =svc_handler        ; 0x0000_0008 : Software interrupts exceptions
LDR pc, =prefetch_handler   ; 0x0000_000c : Prefetch abort exception
LDR pc, =abort_handler      ; 0x0000_0010 : Data abort exception
LDR pc, =reserved_handler   ; 0x0000_0014 : Reserved
LDR pc, =irq_handler        ; 0x0000_0018 : IRQ exception
LDR pc, =fiq_handler        ; 0x0000_001c : FIQ exception
```

図 6.7 例外処理ベクタテーブルの記述例

6.5 使用割り込み一覧

表 6.4 にサンプルコードで使用する割り込みを示します。

表 6.4 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
ADF (ID : 170)	5	処理概要 A/D 変換終了を知らせるフラグをセットし割り込み要求 ADF をクリアします。 処理時間 964.8ns(最大値) 周期 約 10ms

6.6 固定幅整数一覧

表 6.5 にサンプルコードで使用する固定幅整数を示します。

表 6.5 サンプルコードで使用する固定幅整数

シンボル	内容
char_t	8 ビット文字
bool_t	論理型。値は true (1) , false (0)
int_t	高速な整数、符号あり、本サンプルコードでは 32 ビット整数
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)
float32_t	32 ビット浮動小数 ("__ARM_NEON__" を指定時、標準ライブラリにて定義)
float64_t	64 ビット浮動小数 (標準ライブラリにて定義) ("__ARM_NEON__" を指定時、標準ライブラリにて定義)
float128_t	128 ビット浮動小数

6.7 定数/エラーコード一覧

表 6.6 にサンプルコードで使用する定数、表 6.7 にサンプルコードのエラーコードを示します。

表 6.6 サンプルコードで使用する定数(1/2)

定数名	設定値	内容
ADCSR_SET	0x40C0u	ADCSR 設定値
STBCR_6_7_SET	0x7Fu	ADC にクロック供給
PIBC1_8_SET	0x0100u	ポート初期化 (ポート入力モード、入力バッファ禁止状態)
PBDC1_8_SET	0x0100u	
PM1_8_SET	0x0100u	
PIPC1_8_SET	0x0100u	直接 I/O 兼用モードに設定
PMC1_8_SET	0x0100u	P1_8 を兼用モードに設定
PFCAE1_8_SET	0x0100u	兼用機能を、第一兼用機能の ADC の入力に設定
PFCE1_8_SET	0x0100u	
PFC1_8_SET	0x0100u	
ADST_ON	0x2000uL	ADC の起動
ADST_OFF	0xDFFFuL	ADC の停止
ADC_12BIT_SET	0x0FFFuL	ADDRA の上位 4 ビットを 0 にする
ADC_BIT_SHIFT_4	0x0004uL	ADDRA を 4 ビット右にシフトする。
ADIE_ON	0x4000uL	A/D 変換終了割り込みを許可
ADIE_OFF	0xBFFFuL	A/D 変換終了割り込みを禁止
ADF_CLEAR	0x7FFFuL	A/D 変換終了割り込みをクリア
SW1_Level_L	636L	SW1 の検出範囲下限値
SW1_Level_H	1008L	SW1 の検出範囲上限値
SW2_Level_L	1461L	SW2 の検出範囲下限値
SW2_Level_H	1833L	SW2 の検出範囲上限値
SW3_Level_L	2262L	SW3 の検出範囲下限値
SW3_Level_H	2634L	SW3 の検出範囲上限値
SW4_Level_L	3087L	SW4 の検出範囲下限値
SW4_Level_H	3459L	SW4 の検出範囲上限値
ADC_KEY_SW1	1u	キー番号 SW1
ADC_KEY_SW2	2u	キー番号 SW2
ADC_KEY_SW3	3u	キー番号 SW3
ADC_KEY_SW4	4u	キー番号 SW4
ADC_KEY_SW_OFF	0u	キーを識別できなかった場合
KEY_AGREE	2u	キー識別結果比較一致回数
KEY_AGREE_CONTINUE	1u	連続一致した後の比較回数
ADC_SET_FLG	1u	A/D 変換終了によるソフトウェアフラグ
DECREMENT	1L	一致回数のデクリメント

表 6.6 サンプルコードで使用する定数(2/2)

定数名	設定値	内容
ADC_TIME	44444uL	待ち時間を作るためのループ回数 (1 ループあたり 225ns)* 1
ADC_PRI_ADI	5uL	ADI 割り込み優先度
PRINTF_INTEGER	"%d"	printf 関数へ渡す第一引数
PRINTF_LF	"\n"	printf 関数へ渡す改行コード

*1 表 2.1 動作確認条件における実測値から算出した定数です。

表 6.7 サンプルコードのエラーコード

定数名	設定値	内容
ADC_READ_ERR	- 1	タイムアウトによるエラー

6.8 構造体/共用体一覧

本サンプルコードでは構造体/共用体を使用しません。

6.9 変数一覧

表 6.8 に static 型変数を、表 6.9 に const 型変数を示します。

表 6.8 static 型変数

型	変数名	内容	使用関数
volatile uint8_t	adc_end_flg	A/D 変換終了フラグ	AdcInitialize AdcReadSingle IntAdcEnd
uint8_t	key_count	キーの比較一致回数	AdcInitialize ReadKey1To4
uint8_t	key_before	前回のキー識別結果	AdcInitialize ReadKey1To4

表 6.9 const 型変数

本サンプルコードは const 型変数を使用しません。

6.10 関数一覧

表 6.10 に関数を示します。

表 6.10 関数一覧

関数名	ページ番号
R_ADC_SampleMain	25
AdcInitialize	25
AdcStart	25
AdcReadSingle	26
AdcStop	26
AdcUninitialize	26
ReadKey1To4	27
IntAdcEnd	27
Delay10ms	27

6.11 関数仕様

サンプルコードの関数仕様を示します。

6.11.1 R_ADC_SampleMain

R_ADC_SampleMain

概 要	ADC サンプルプログラムのメイン処理部	
ヘッダ	adc_sample.h	
宣 言	int32_t R_ADC_SampleMain(int32_t argc, char_t **argv)	
説 明	ADC サンプルプログラムのメイン処理部です。	
引 数	int32_t argc	: 未使用
	char_t **argv	: 未使用
リターン値	0	: 固定

6.11.2 AdcInitialize

AdcInitialize

概 要	ADC の初期化	
ヘッダ	なし	
宣 言	static void AdcInitialize(void);	
説 明	本関数は ADC の初期化を下記の設定で行います。 1. A/D 変換の終了による割り込み(ADI)要求を許可 2. A/D 変換時間を 382tcyc に設定 3. 動作モードをシングルモードに設定 4. アナログ入力を AN0 に設定 ADC を制御する全ての関数の前に実行してください。	
引 数	なし	
リターン値	なし	

6.11.3 AdcStart

AdcStart

概 要	ADC の起動	
ヘッダ	なし	
宣 言	static void AdcStart(void);	
説 明	ADC の起動を行います。 AdcInitialize の後に本関数を実行してください。	
引 数	なし	
リターン値	なし	

6.11.4 AdcReadSingle

AdcReadSingle

概 要	変換結果のリード	
ヘッダ	なし	
宣 言	static int32_t AdcReadSingle(void);	
説 明	本関数は ADC を起動した後、AN0 の変換結果を取得し、下位ビットに詰めて返却します。 変換が終わらない場合は、タイムアウトによる読み込みエラーの ADC_READ_ERR を返却します。 AdcStart の後に実行してください。	
引 数	なし	
リターン値	0~4095	: A/D 変換値
	ADC_READ_ERR (-1)	: タイムアウトによる読み込みエラー

6.11.5 AdcStop

AdcStop

概 要	ADC の停止
ヘッダ	なし
宣 言	static void AdcStop(void);
説 明	ADC の停止を行います。
引 数	なし
リターン値	なし

6.11.6 AdcUninitialize

AdcUninitialize

概 要	ADC の終了処理
ヘッダ	なし
宣 言	static void AdcUninitialize(void);
説 明	ADC の初期設定を無効にします。 本関数の実行後に他の ADC 制御関数を実行する場合は、先に AdcInitialize を実行してください。
引 数	なし
リターン値	なし

6.11.7 ReadKey1To4

ReadKey1To4

概 要	キー入力の識別	
ヘッダ	なし	
宣 言	static uint8_t ReadKey1To4(void);	
説 明	本関数は AN0 に接続されたキー入力を識別します。 AdcReadSingle を用いて変換結果を取得し、変換結果を元に押されたキーを判断し、キー番号を戻り値として返却します。キー入力なし、または変換結果がキーの識別範囲外で識別できなかった場合は、ADC_KEY_SW_OFF を返却します。 AdcStart の後に実行してください。	
引 数	なし	
リターン値	ADC_KEY_SW1(1u)	: キー番号 SW1
	ADC_KEY_SW2(2u)	: キー番号 SW2
	ADC_KEY_SW3(3u)	: キー番号 SW3
	ADC_KEY_SW4(4u)	: キー番号 SW4
	ADC_KEY_SW_OFF(0u)	: キー入力なし、または SW1~4 に該当しなかった場合

6.11.8 IntAdcEnd

IntAdcEnd

概 要	A/D 変換の終了割り込み関数	
ヘッダ	なし	
宣 言	static void IntAdcEnd(uint32_t int_sense);	
説 明	A/D 変換が終了したことを示すフラグをセットし、割り込み要求(ADF)をクリアします。	
引 数	uint32_t int_sense	: 割り込み発生要因を指定するための変数 レベルセンス (0) エッジセンス (1) (本プログラムでは特に指定しないため使用しない)
リターン値	なし	

6.11.9 Delay10ms

Delay10ms

概 要	サンプリング間隔生成用遅延関数	
ヘッダ	なし	
宣 言	static void Delay10ms(void);	
説 明	本関数の実行開始から約 10ms 間遅延します。 本関数を実行中は割り込み処理を除く他の処理は行えません。	
引 数	なし	
リターン値	なし	
補 足	本関数の実行時間は実測値から算出した表 6.6 の定数 ADC_TIME により決定します。 表 2.1 動作確認条件と異なる開発環境の場合は、実行時間が変動します。	

6.12 フローチャート

6.12.1 メイン処理

図 6.8 にメイン処理のフローチャートを示します。

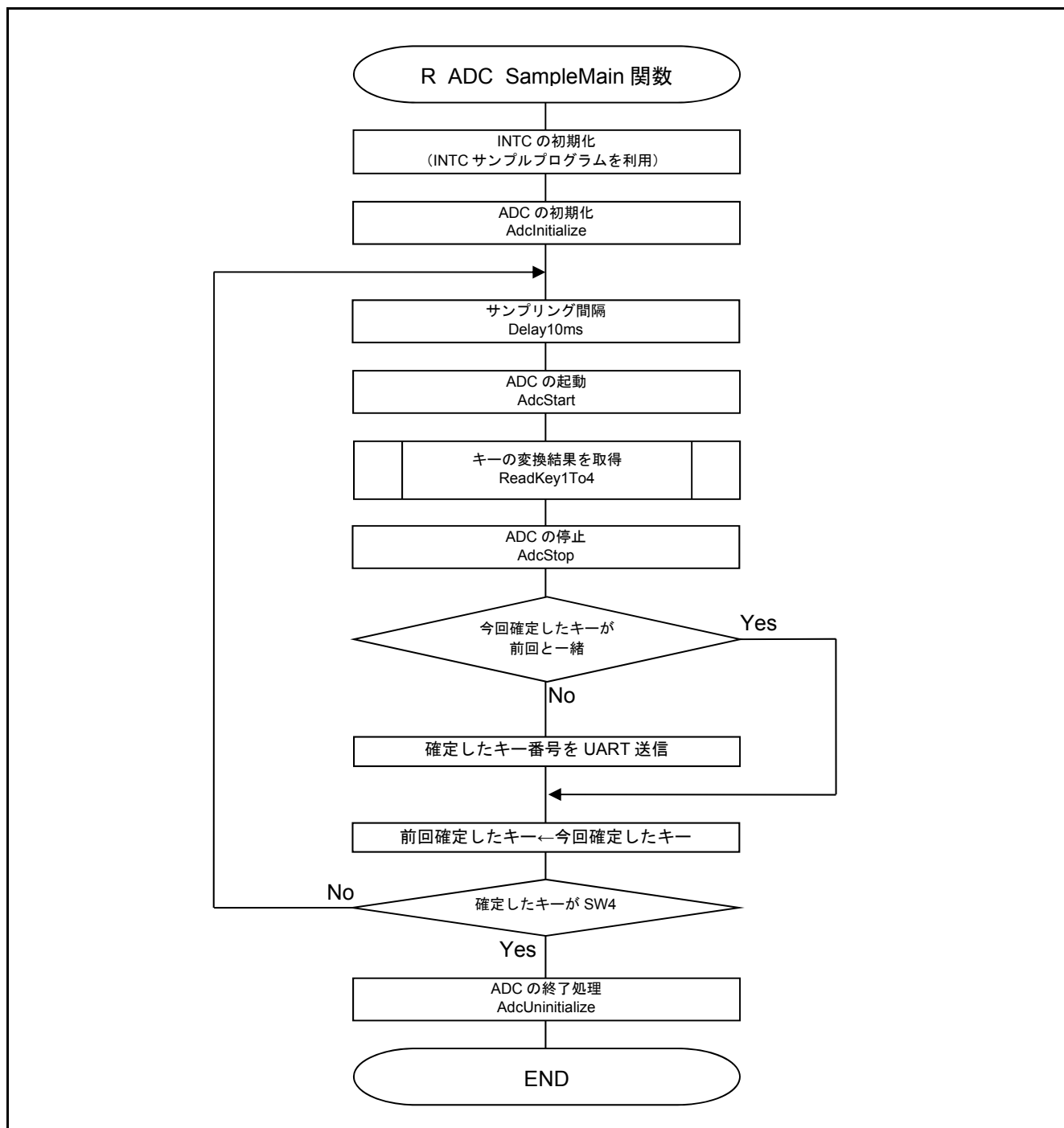


図 6.8 メイン処理

6.12.2 キー変換結果の取得処理

図 6.9 にキーの変換結果を取得するフローチャートを示します。

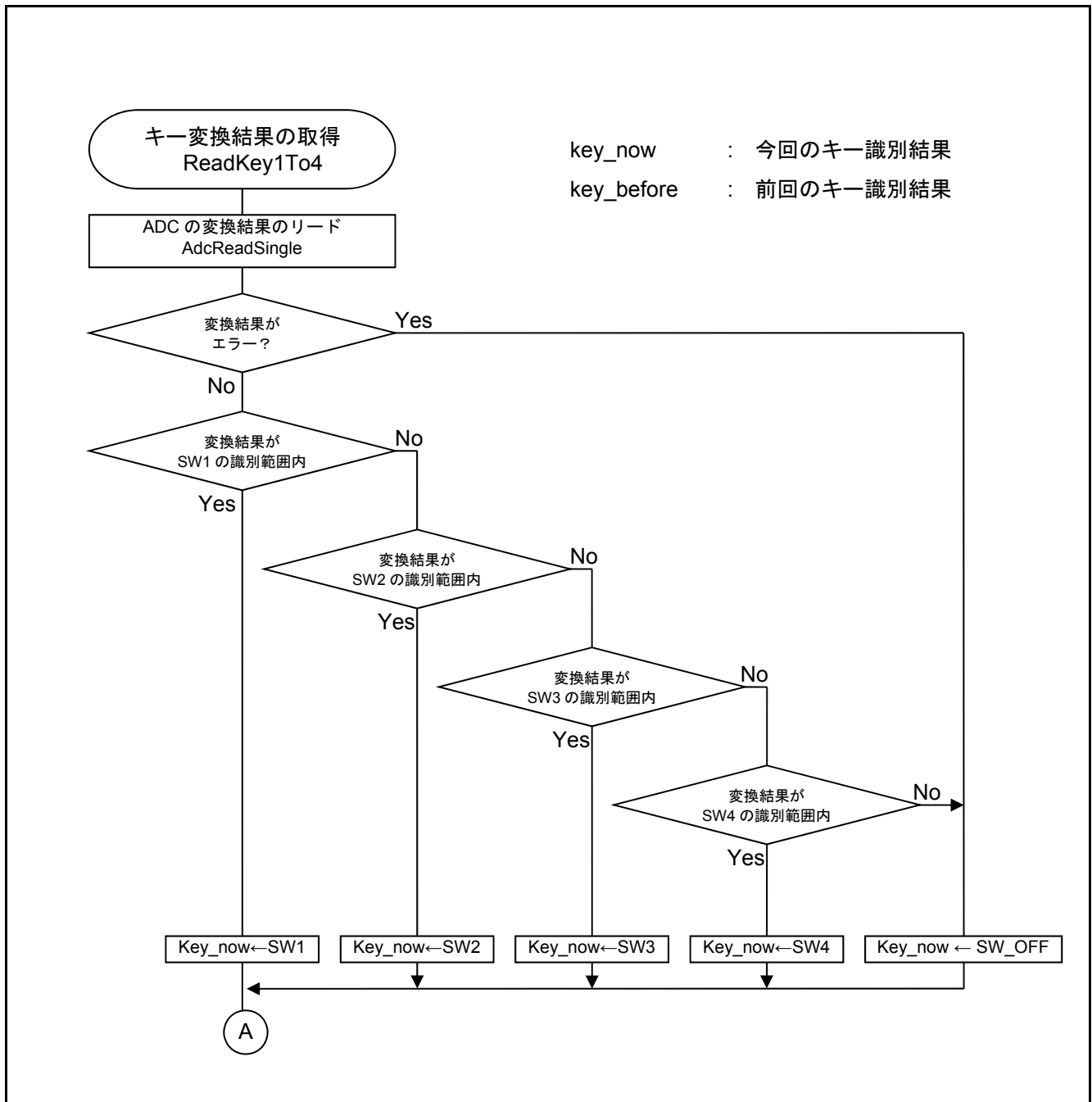


図 6.9 キー変換結果の取得処理(1/2)

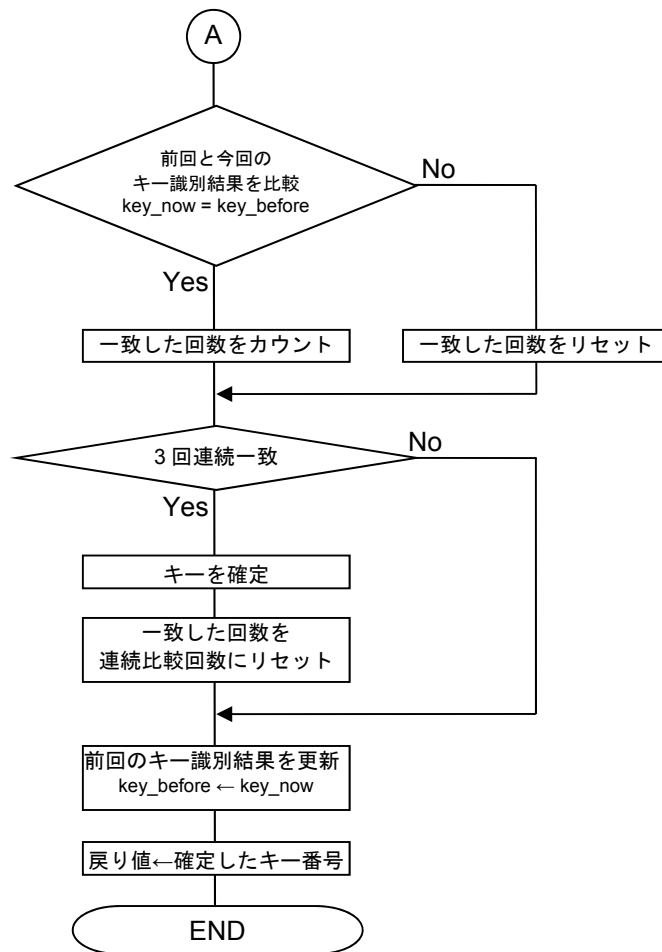


図 6.9 キー変換結果の取得処理(2/2)

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/A1H 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。)

R7S72100 CPU ボード RTK772100BC00000BR (GENMAI) 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。)

R7S72100 CPU ボード用オプションボード RTK7721000B00000BR (GENMAI) 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。)

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C

(最新版を ARM ホームページから入手してください。)

ARM Generic Interrupt Controller Architecture Specification Architecture version 1.0

(最新版を ARM 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

ARM ソフトウェア開発ツール (ARM Compiler toolchain、ARM DS-5 等) に関しては、ARM ホームページから入手してください。

(最新版を ARM ホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問い合わせ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
0.80	2014.04.16	—	初版発行
1.00	2014.06.24	22 27	複数コンパイラ対応 コンパイラが変わっても実行時間の変動が小さくなるよう修正

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. 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うと、内部ROM、レイアウトパターンの相違などにより、電氣的特性の範囲で、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も、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事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がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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



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/>