

## RZ/T1 グループ

R20AN0450JJ0100

Rev.1.00

## RZ/T1 用 CMSIS-DSP ライブラリ：導入ガイド

2017.06.23

### 要旨

本資料は、RZ/T1 用 CMSIS-DSP ライブラリを導入するための情報を示します。

### 特長

RZ/T1 用 CMSIS-DSP ライブラリは、ARM<sup>®</sup> Ltd.提供の CMSIS-DSP Version 1.4.7(CMSIS V4.5.0)に対して、RZ/T1 内蔵の BTCM 領域に配置するため、および性能向上のために、以下の変更を行っています。

- ・サンプリング点数毎にテーブルを分割する変更を行っています。  
(Transform Functions の各 API 関数が対象)
- ・ユーザによって使用するサンプリング点数を指定することにより、使用するメモリサイズ  
(テーブル) を必要最小限に最適化できます。  
(Transform Functions の各 API 関数が対象)
- ・コンパイラ毎に、処理速度が向上するようにライブラリの最適化を行っています。

### 動作確認デバイス

RZ/T1 グループ

### 動作確認 統合開発環境 (IDE) / コンパイラ / Evaluation Board

e2 studio / Renesas GCC / Renesas Starter Kit+ for RZ/T1

EWARM / IAR CC / Renesas Starter Kit+ for RZ/T1

## 目次

1. 仕様.....	3
2. 動作確認条件.....	3
3. ライブラリ作成時のオプション .....	3
4. ソフトウェアの説明 .....	4
4.1 構成 .....	4
4.2 高速メモリ配置 .....	4
4.3 フォルダの構成 .....	5
4.4 メモリサイズ.....	6
5. コンフィグレーション.....	7
5.1 コンフィグレーション方法 .....	7
5.2 消費メモリサイズ .....	9
5.3 注意事項 .....	9
6. Transform Functions 情報.....	10
6.1 初期化関数 .....	10
6.2 サンプリング点数 .....	11
6.3 マクロ定義 .....	11
6.4 消費メモリサイズ .....	12
6.5 注意事項 .....	15
7. CMSIS-DSP プリプロセッサマクロ .....	16
8. ライブラリ使用方法 .....	17
9. ライブラリの最適化の効果.....	18
10. ライセンス .....	19
11. ホームページとサポート窓口 .....	20

## 1. 仕様

本ライブラリの API 仕様は、オリジナルの CMSIS-DSP Version 1.4.7 から変更ありません。

API 仕様は、ARM Ltd が提供している CMSIS-DSP のマニュアルを参照してください。

- CMSIS

<http://www.arm.com/ja/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php>

- CMSIS-DSP

<http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>

## 2. 動作確認条件

本ライブラリは、下記表の条件で動作を確認しております。

表 1. 動作確認条件

項目	名称	バージョン
ターゲット	CMSIS-DSP Library	CMSIS Version 4.5.0 (DSP Revision : Version1.4.7)
統合開発環境	Renesas GCC	Eclipse SDK Version:4.5.2 e2 studio
	IAR CC	IAR Embedded Workbench for ARM
コンパイラ	Renesas GCC	V16.01
	IAR CC	-
プロセッサ	RZ/T1	-
Evaluation Board	RENESAS Starter Kit+ for RZ/T1 Board (R7S910018)	-
エミュレータ	Renesas GCC	J-Link LITE
	IAR CC	I-jet

## 3. ライブラリ作成時のオプション

本ライブラリ作成時の各コンパイラのオプションは、下記表のとおりです。

表 2. ライブラリ作成時のオプション

コンパイラ	項目	設定値
Renesas GCC	Compiler	-nostdinc -O2 -mcpu=cortex-r4r -march=armv7-r -mlittle-endian -mthumb -mthumb-interwork -mfloat-abi=softfp -mfpu=vfp -DDOUBLE_HAS_64_BITS -std=c99 -fno-strict-aliasing
	Assembler	-mlittle-endian -mcpu=cortex-r4r -march=armv7-r -mthumb -mfloat-abi=softfp -mfpu=vfp
IAR CC	Compiler	--endian=little --cpu=Cortex-R4 -e --fpu=VFPv3_D16 --cpu_mode thumb -Ohs --no_size_constraints
	Assembler	-s+ -M<> -w+ --cpu Cortex-R4 --fpu VFPv3_D16

## 4. ソフトウェアの説明

### 4.1 構成

本ライブラリ使用時のソフトウェアの構成例を以下に示します。

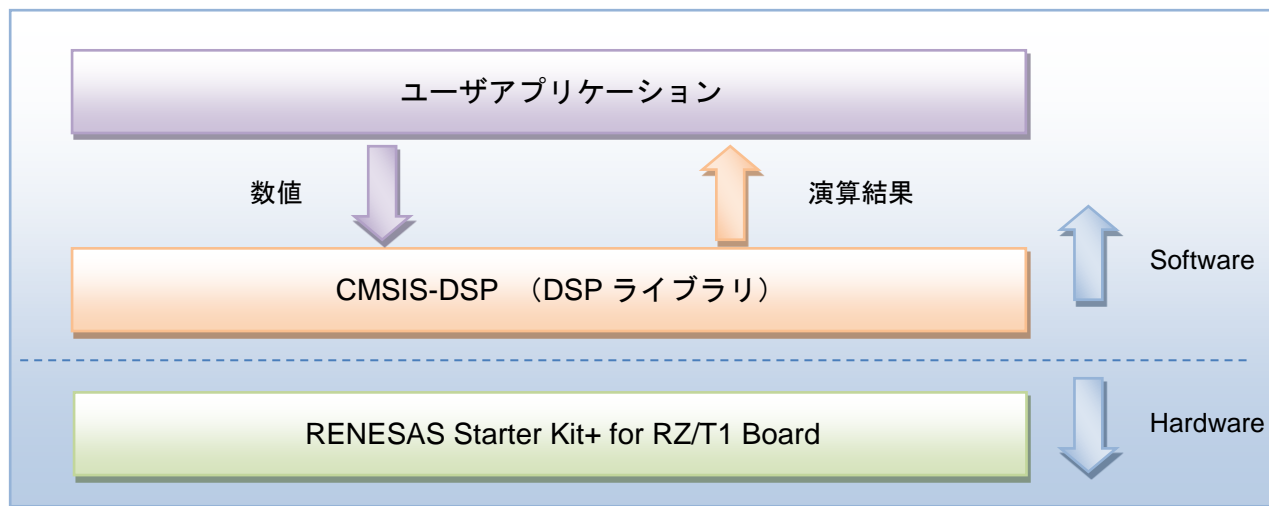


図 1. 構成例

### 4.2 高速メモリ配置

本ライブラリは、以下の条件を満たすようにメモリ配置することで、最大限のパフォーマンスを発揮することができます。

- 高速メモリ配置条件（1～3の and 条件）
  1. API 関数と、参照・更新されるメモリが密結合メモリ（以降、TCM）に配置されていること。
  2. API 関数と、参照・更新されるメモリが異なる TCM へ配置されていること。
  3. 特に、CPU クロック周波数が 450MHz 以上の場合、API 関数本体が ATCM、参照・更新されるメモリが BTCM に配置され、かつ、「ATCM ウェイト制御レジスタ」の「ATCM ウェイト設定ビット」が「1-wait 最適化あり」となっていること。

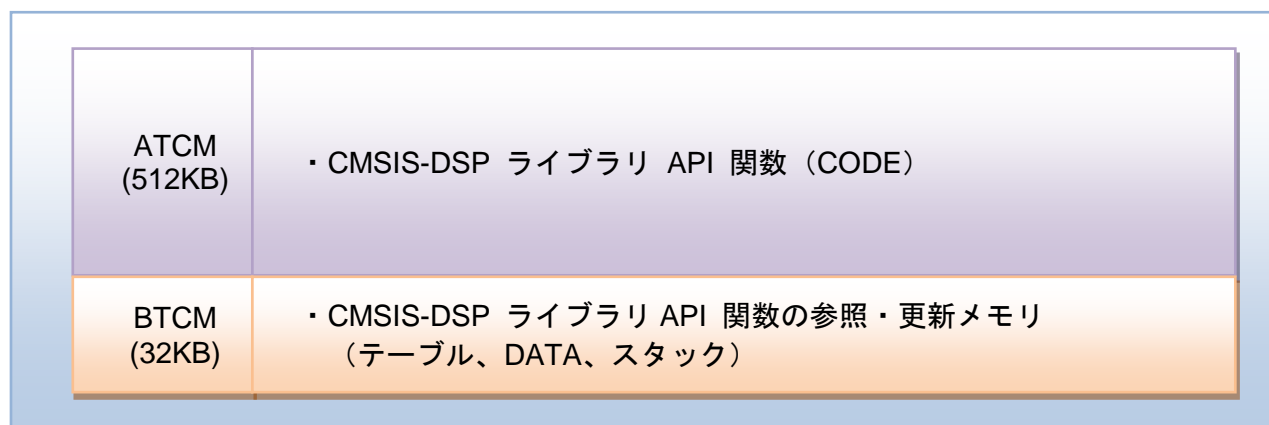


図 2. 高速メモリ配置例（CPU クロック周波数が 450MHz 以上時）

## 4.3 フォルダの構成

本ライブラリ使用時のフォルダ構成は、下記表のとおりです。

表 3. フォルダ構成

¥an-r20an0450jj0100-rzt1	<DIR>	CMSIS-DSP ソフトウェアのフォルダ
r20an0450jj0100-rzt1.pdf		本アプリケーションノート
¥workspace	<DIR>	プログラム格納用フォルダ
¥build	<DIR>	環境設定ファイル格納用フォルダ
¥gcc	<DIR>	e2 studio 用フォルダ
¥RZ_T1_CMSIS_DSP_LIB	<DIR>	
.cproject		環境設定ファイル
.info		環境設定ファイル
.project		環境設定ファイル
¥iccarm	<DIR>	EWARM 用フォルダ
RZ_T1_CMSIS_DSP_LIB.dep		環境設定ファイル
RZ_T1_CMSIS_DSP_LIB.ewd		環境設定ファイル
RZ_T1_CMSIS_DSP_LIB.ewp		環境設定ファイル
RZ_T1_CMSIS_DSP_LIB.ewt		環境設定ファイル
RZ_T1_CMSIS_DSP_LIB.eww		環境設定ファイル
¥DSP_Lib	<DIR>	CMSIS-DSP ライブラリソースコード格納用フォルダ (*1)
Source	<DIR>	CMSIS-DSP ライブラリソースコード格納用フォルダ
license.txt		ライセンスファイル
¥Lib	<DIR>	CMSIS-DSP ライブラリ格納用フォルダ
¥gcc	<DIR>	Renesas GCC 用ライブラリ格納用フォルダ
libRZ_T1_CMSIS_DSP_LIB_gcc.a		ライブラリファイル
¥iarcc	<DIR>	IAR CC 用ライブラリ格納用フォルダ
libRZ_T1_CMSIS_DSP_LIB_iarcc.a		ライブラリファイル
license.txt		ライセンスファイル
¥Include	<DIR>	ヘッダファイル格納用フォルダ
arm_common_tables.h		ヘッダファイル
arm_const_structs.h		ヘッダファイル
arm_math.h		ヘッダファイル
cmsis_armcc.h		ヘッダファイル
cmsis_armcc_V6.h		ヘッダファイル
cmsis_gcc.h		ヘッダファイル
core_cm0.h		ヘッダファイル
core_cm0plus.h		ヘッダファイル
core_cm3.h		ヘッダファイル
core_cm4.h		ヘッダファイル
core_cm7.h		ヘッダファイル
core_cmFunc.h		ヘッダファイル
core_cmInstr.h		ヘッダファイル
core_cmSimd.h		ヘッダファイル
core_sc000.h		ヘッダファイル
core_sc300.h		ヘッダファイル

¥InitSrc (*2)	<DIR>	ソースファイル格納用フォルダ
arm_cfft_radix2_init_f32.c		ソースファイル
arm_cfft_radix2_init_q15.c		ソースファイル
arm_cfft_radix2_init_q31.c		ソースファイル
arm_cfft_radix4_init_f32.c		ソースファイル
arm_cfft_radix4_init_q15.c		ソースファイル
arm_cfft_radix4_init_q31.c		ソースファイル
arm_dct4_init_f32.c		ソースファイル
arm_dct4_init_q15.c		ソースファイル
arm_dct4_init_q31.c		ソースファイル
arm_rfft_init_f32.c		ソースファイル
arm_rfft_fast_init_f32.c		ソースファイル
arm_rfft_init_q15.c		ソースファイル
arm_rfft_init_q31.c		ソースファイル

\*1 フォルダ構成は、ARM Ltd.提供の CMSIS-DSP と同じため、省略しています。

\*2 DSP\_Lib フォルダ配下の Transform Functions フォルダに含まれる init ファイルを格納したフォルダです。

#### 4.4 メモリサイズ

本ライブラリ使用時のメモリサイズは、下記表のとおりです。

表 4. ライブラリのメモリサイズ

コンパイラ	ROM (KB)	RAM (KB)	スタック (KB)
Renesas GCC	309	0	384
IAR CC	336	0	268

## 5. コンフィグレーション

RZ/T1 用 CMSIS-DSP ライブラリの Transform Functions の API 関数は、ユーザアプリケーションが使用するサンプリング点数に合わせて、使用するテーブルを最適化できる拡張を行っています。

本拡張によって、例えば、API 関数「arm\_cfft\_f32」の場合は、最大サンプリング点数 1 0 2 4 (\*1) まで、本ライブラリの高速メモリ配置が実現できます。

ユーザは、使用する API 関数のサンプル点数に対応するマクロ定義を統合開発環境に追加するだけで、本拡張を使用することができます。詳細は、後述するコンフィグレーション方法の手順と、マクロ定義を参照してください。

\*1 ユーザアプリケーションが使用するメモリサイズは考慮されていません。高速メモリ配置は、「4.2 高速メモリ配置」を参照してください。

### 5.1 コンフィグレーション方法

本ライブラリのコンフィグレーション方法を説明します。

#### ● 手順（概要）

1. 使用する API 関数とサンプリング点数を選択します。
2. 1 で選択した API 関数のサンプリング点数は、「6.3 マクロ定義」より取得したマクロを、統合開発環境のコンパイルオプションのマクロ定義（シンボル定義）に追加することにより、有効にできます。
3. 複数のサンプリング点数を使用する場合は、1～2の手順を繰り返します。

#### ● 手順（詳細）

1. 使用する API 関数とサンプリング点数を選択します。選択可能な組み合わせは、「表 6. サポートするサンプリング点数一覧」を参照してください。

例、API 関数「arm\_cfft\_f32」のサンプリング点数を選択する場合は、下図より、サンプリング点数 1 6 ～ 4 0 9 6 のいずれかが選択できます。

No.	API 関数名	サポートするサンプリング点数									
		16	32	64	128	256	512	1024	2048	4096	8192
1	arm_cfft_f32	○	○	○(*1)	○	○	○(*1)	○	○	○(*1)	—

図 3. API 関数「arm\_cfft\_f32」がサポートするサンプリング点数 (\*2)

\*2 「表 6. サポートするサンプリング点数一覧」より抜粋

2. 1 で選択した API 関数のサンプリング点数は、「6.3 マクロ定義」より取得したマクロを、統合開発環境のコンパイルオプションのマクロ定義（シンボル定義）に追加することにより、有効にできます。

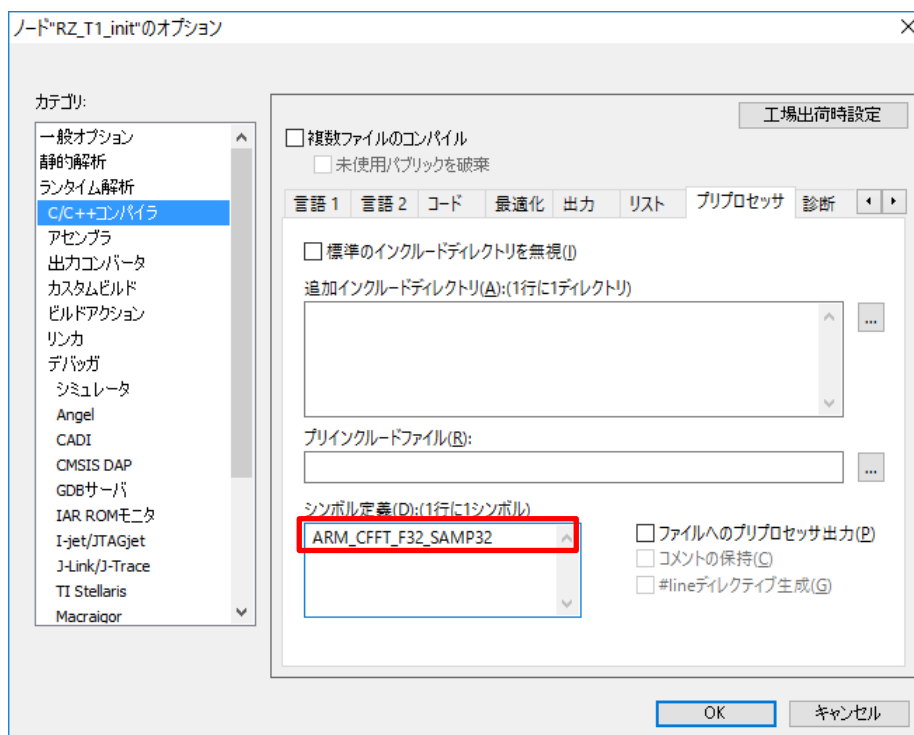


図 4. EWARM にて、API 関数「arm\_cfft\_f32」の「サンプリング点数 32」を有効にする例

3. 複数のサンプリング点数を使用する場合は、1～2の手順を繰り返します。

複数のサンプリング定数を有効にする場合、統合開発環境のコンパイルオプションのマクロ定義は、有効にする定義をすべて追加します。複数の API 関数を使用する場合も同様です。

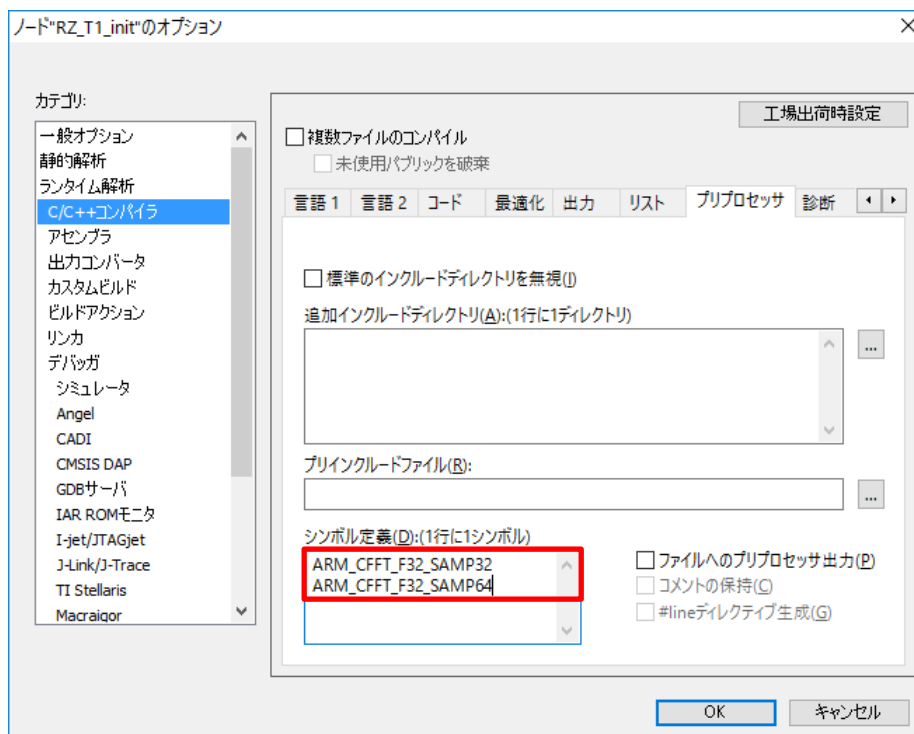


図 5. EWARM にて、API 関数「arm\_cfft\_f32」の「サンプリング点数 32 と 64」を有効にする例



## 5.2 消費メモリサイズ

選択した API 関数のサンプリング点数が消費するメモリサイズは、「表 8. Transform Functions の消費メモリサイズ一覧」にて確認できます。

例、API 関数「arm\_cfft\_f32」の「サンプリング点数 32」を選択する場合、消費するメモリサイズは、図 6（赤枠部分）より、「0.6 KB」となります。

No.	API関数名	サンプリング点数	テーブル(KB)	バッファ(KB)	合計(KB)
2	arm_cfft_f32	32	0.3	0.3	0.6

図 6. API 関数「arm\_cfft\_f32」の「サンプリング点数 32」選択時の消費メモリサイズ (\*2)

複数のサンプリング点数を選択する場合、消費するメモリサイズは、該当のテーブルサイズの合計と最大バッファサイズの合計となります。

例、API 関数「arm\_cfft\_f32」の「サンプリング点数 32 と 64」を選択する場合、消費するメモリサイズは、図 7（赤枠部分）より、以下となります。

テーブルサイズ：0.3 (KB) + 0.6 (KB) = 0.9 (KB)  
 バッファサイズ：最大バッファサイズ 0.5 (KB)  
 合計：1.4 (KB)

No.	API関数名	サンプリング点数	テーブル(KB)	バッファ(KB)	合計(KB)
2	arm_cfft_f32	32	0.3	0.3	0.6
3		64	0.6	0.5	1.1

図 7. API 関数「arm\_cfft\_f32」の「サンプリング点数 32 と 64」選択時の消費メモリサイズ (\*2)

\*2 「表 8. Transform Functions の消費メモリサイズ一覧」より抜粋

## 5.3 注意事項

「5. コンフィグレーション」を使用する場合は、「表 5. 初期化関数一覧」を参照し、使用する API 関数実行前に、対応する初期化関数を呼び出してください。

該当する初期化関数がない場合は、不要です。

## 6. Transform Functions 情報

Transform Functions の API 関数を使用するための情報を以下に示します。

### 6.1 初期化関数

Transform Functions の各 API 関数実行前に呼び出す初期化関数を下記表に示します。

- 記号説明  
— : 該当なし

表 5. 初期化関数一覧

No.	API 関数名	初期化関数名
1	arm_cfft_f32	—
2	arm_cfft_q15	—
3	arm_cfft_q31	—
4	arm_cfft_radix2_f32	arm_cfft_radix2_init_f32
5	arm_cfft_radix2_q15	arm_cfft_radix2_init_q15
6	arm_cfft_radix2_q31	arm_cfft_radix2_init_q31
7	arm_cfft_radix4_f32	arm_cfft_radix4_init_f32
8	arm_cfft_radix4_q15	arm_cfft_radix4_init_q15
9	arm_cfft_radix4_q31	arm_cfft_radix4_init_q31
10	arm_cfft_radix8_f32	—
11	arm_dct4_f32	arm_dct4_init_f32
12	arm_dct4_q15	arm_dct4_init_q15
13	arm_dct4_q31	arm_dct4_init_q31
14	arm_rfft_f32	arm_rfft_init_f32
15	arm_rfft_fast_f32	arm_rfft_fast_init_f32
16	arm_rfft_q15	arm_rfft_init_q15
17	arm_rfft_q31	arm_rfft_init_q31

## 6.2 サンプルング点数

Transform Functions の各 API 関数がサポートするサンプルング点数を下記表に示します。

- 記号説明
  - ：対応
  - －：非対応

表 6. サポートするサンプルング点数一覧

No.	API 関数名	サポートするサンプルング点数									
		16	32	64	128	256	512	1024	2048	4096	8192
1	arm_cfft_f32	○	○	○(*1)	○	○	○(*1)	○	○	○(*1)	－
2	arm_cfft_q15	○(*2)	○	○(*2)	○	○(*2)	○	○(*2)	○	○(*2)	－
3	arm_cfft_q31	○(*2)	○	○(*2)	○	○(*2)	○	○(*2)	○	○(*2)	－
4	arm_cfft_radix2_f32	○	○	○	○	○	○	○	○	○	－
5	arm_cfft_radix2_q15	○	○	○	○	○	○	○	○	○	－
6	arm_cfft_radix2_q31	○	○	○	○	○	○	○	○	○	－
7	arm_cfft_radix4_f32	○	－	○	－	○	－	○	－	○	－
8	arm_cfft_radix4_q15	○	－	○	－	○	－	○	－	○	－
9	arm_cfft_radix4_q31	○	－	○	－	○	－	○	－	○	－
10	arm_cfft_radix8_f32	－	－	○	－	－	○	－	－	○	－
11	arm_dct4_f32	－	－	－	○	－	○	－	○	－	○
12	arm_dct4_q15	－	－	－	○	－	○	－	○	－	○
13	arm_dct4_q31	－	－	－	○	－	○	－	○	－	○
14	arm_rfft_f32	－	－	－	○	－	○	－	○	－	○
15	arm_rfft_fast_f32	－	○	○	○	○	○	○	○	○	－
16	arm_rfft_q15	－	○	○	○	○	○	○	○	○	○
17	arm_rfft_q31	－	○	○	○	○	○	○	○	○	○

\*1 arm\_cfft\_radix8\_f32 として扱っています。

\*2 arm\_cfft\_radix4\_q15 / arm\_cfft\_radix4\_q31 と同等処理として扱っています。

## 6.3 マクロ定義

使用する Transform Functions の API 関数のサンプルング点数有効にする場合のマクロ名の命名方法を以下に示します。

表 7. マクロ定義

マクロ名	説明
[XXX]_SAMP[YYYY]	有効にする API 関数のサンプルング点数を指定します。 [XXX] = Transform Functions の API 関数名を全て大文字で指定します。 [YYY] = サンプルング点数を指定します。 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192 のいずれかの値です。

API 関数「arm\_cfft\_f32」の「サンプルング点数 16」を指定するマクロ名の例

```
ARM_CFFT_F32_SAMP16
```

## 6.4 消費メモリサイズ

Transform Functions の各 API 関数のサンプリング点数毎のテーブル、バッファサイズを下記表に示します。バッファサイズは、API 関数使用時に、ユーザが入出力バッファ用などで最低限確保する必要があるメモリサイズの参考値です。

表 8. Transform Functions の消費メモリサイズ一覧 (1 / 3)

No.	API関数名	サンプリング点数	テーブル(KB)	バッファ(KB)	合計(KB)
1	am_cfft_f32	16	0.2	0.1	0.3
2		32	0.3	0.3	0.6
3		64	0.6	0.5	1.1
4		128	1.4	1.0	2.4
5		256	2.9	2.0	4.9
6		512	4.9	4.0	8.9
7		1024	11.5	8.0	19.5
8		2048	23.4	16.0	39.4
9		4096	39.9	32.0	71.9
10	am_cfft_q15	16	0.1	0.1	0.1
11		32	0.1	0.1	0.3
12		64	0.3	0.3	0.5
13		128	0.6	0.5	1.1
14		256	1.2	1.0	2.2
15		512	2.4	2.0	4.4
16		1024	4.9	4.0	8.9
17		2048	9.9	8.0	17.9
18		4096	19.9	16.0	35.9
19	am_cfft_q31	16	0.1	0.1	0.2
20		32	0.2	0.3	0.5
21		64	0.5	0.5	1.0
22		128	1.0	1.0	2.0
23		256	2.0	2.0	4.0
24		512	3.9	4.0	7.9
25		1024	7.9	8.0	15.9
26		2048	15.9	16.0	31.9
27		4096	31.9	32.0	63.9
28	am_cfft_radix2_f32	16	0.1	0.1	0.3
29		32	0.3	0.3	0.5
30		64	0.5	0.5	1.0
31		128	1.1	1.0	2.1
32		256	2.1	2.0	4.1
33		512	4.3	4.0	8.3
34		1024	8.5	8.0	16.5
35		2048	17.0	16.0	33.0
36		4096	34.0	32.0	66.0

表 8. Transform Functions の消費メモリサイズ一覧 (2/3)

No.	API関数名	サンプリング点数	テーブル(KB)	バッファ(KB)	合計(KB)
37	am_cfft_radix2_q15	16	0.1	0.1	0.1
38		32	0.1	0.1	0.2
39		64	0.2	0.3	0.5
40		128	0.4	0.5	0.9
41		256	0.9	1.0	1.9
42		512	1.8	2.0	3.8
43		1024	3.5	4.0	7.5
44		2048	7.0	8.0	15.0
45		4096	14.0	16.0	30.0
46	am_cfft_radix2_q31	16	0.1	0.1	0.2
47		32	0.2	0.3	0.5
48		64	0.4	0.5	0.9
49		128	0.8	1.0	1.8
50		256	1.6	2.0	3.6
51		512	3.3	4.0	7.3
52		1024	6.5	8.0	14.5
53		2048	13.0	16.0	29.0
54		4096	26.0	32.0	58.0
55	am_cfft_radix4_f32	16	0.1	0.1	0.3
56		64	0.5	0.5	1.0
57		256	2.1	2.0	4.1
58		1024	8.5	8.0	16.5
59		4096	34.0	32.0	66.0
60	am_cfft_radix4_q15	16	0.1	0.1	0.1
61		64	0.2	0.3	0.5
62		256	0.9	1.0	1.9
63		1024	3.5	4.0	7.5
64		4096	14.0	16.0	30.0
65	am_cfft_radix4_q31	16	0.1	0.1	0.2
66		64	0.4	0.5	0.9
67		256	1.6	2.0	3.6
68		1024	6.5	8.0	14.5
69		4096	26.0	32.0	58.0
70	am_cfft_radix8_f32	64	0.6	0.5	1.1
71		512	4.9	4.0	8.9
72		4096	39.9	32.0	71.9

表 8. Transform Functions の消費メモリサイズ一覧 (3/3)

No.	API関数名	サンプリング点数	テーブル(KB)	バッファ(KB)	合計(KB)
73	am_rfft_f32	128	1.5	2.0	3.5
74		512	6.1	8.0	14.1
75		2048	24.5	32.0	56.5
76		8192	98.0	128.0	226.0
77	am_rfft_fast_f32	32	0.3	0.5	0.8
78		64	0.6	1.0	1.6
79		128	1.1	2.0	3.1
80		256	2.4	4.0	6.4
81		512	4.9	8.0	12.9
82		1024	8.9	16.0	24.9
83		2048	19.5	32.0	51.5
84		4096	39.4	64.0	103.4
85	am_rfft_q15	32	0.2	0.3	0.5
86		64	0.4	0.5	0.9
87		128	0.8	1.0	1.8
88		256	1.6	2.0	3.6
89		512	3.2	4.0	7.2
90		1024	6.4	8.0	14.4
91		2048	12.9	16.0	28.9
92		4096	25.9	32.0	57.9
93		8192	51.9	64.0	115.9
94	am_rfft_q31	32	0.4	0.5	0.9
95		64	0.7	1.0	1.7
96		128	1.5	2.0	3.5
97		256	3.0	4.0	7.0
98		512	6.0	8.0	14.0
99		1024	11.9	16.0	27.9
100		2048	23.9	32.0	55.9
101		4096	47.9	64.0	111.9
102		8192	95.9	128.0	223.9
103	am_dct4_f32	128	3.0	2.0	5.0
104		512	12.1	8.0	20.1
105		2048	48.5	32.0	80.5
106		8192	194.0	128.0	322.0
107	am_dct4_q15	128	1.5	1.0	2.5
108		512	6.2	4.0	10.2
109		2048	24.9	16.0	40.9
110		8192	99.9	64.0	163.9
111	am_dct4_q31	128	3.0	2.0	5.0
112		512	12.0	8.0	20.0
113		2048	47.9	32.0	79.9
114		8192	191.9	128.0	319.9

## 6.5 注意事項

API 関数の一部のサンプリング点数は、消費メモリサイズの合計が 32KB 以上となっており、BTCM のメモリサイズ 32KB 以下に調整することはできませんので、使用时はご注意ください。

## 7. CMSIS-DSP プリプロセッサマクロ

本ライブラリで選択している CMSIS-DSP のプリプロセッサマクロは、下記表のとおりです。

プリプロセッサマクロは、ヘッダファイル「arm\_math.h」で定義しています。

変更する場合は、マクロ定義を変更後、ライブラリを再生成してください。

表 9. CMSIS-DSP プリプロセッサマクロ

マクロ名	設定値	説明
ARM_MATH_MATRIX_CHECK	(無)	関数の堅牢性を確保するため、行列の入出力サイズをチェックします。
ARM_MATH_ROUNDING	(無)	演算誤差を考慮して、サポート関数の丸めを行います。
UNALIGNED_SUPPORT_DISABLE	(無)	アライメントフォールトが発生しないように、Unaligned memory access 禁止です。
__FPU_PRESENT	1	ターゲットは、FPU をサポートしているため、FPU サポートでビルドします。
ARM_MATH_CM4	(無)	RZ/T1 用 CMSIS-DSP 対応に最適なソースコードを選択します。



## 8. ライブラリ使用方法

本ライブラリを使用する手順について説明します。

### ● 手順 (概要)

1. 本ライブラリの API 関数を使用するために、ヘッダファイル「arm\_math.h」をインクルードします。
2. 使用するコンパイラにあわせてライブラリをリンクします。
3. Transform Functions API を使用する場合は、5.1 コンフィグレーション方法」を参照し、使用する API 関数のサンプリング点数のマクロ定義を有効にします。

### ● 手順 (詳細)

1. 本ライブラリの API 関数を使用するために、ヘッダファイル「arm\_math.h」をインクルードします。
2. 使用するコンパイラにあわせてライブラリをリンクします。

表 10. コンパイラ毎に対応するライブラリファイル

No	コンパイラ	ライブラリファイル
1	Renesas GNU コンパイラ	libRZ_T1_CMSIS_DSP_LIB_gcc.a
2	IAR コンパイラ	libRZ_T1_CMSIS_DSP_LIB_iarcc.a

3. Transform Functions API 関数を使用する場合は、「5.1 コンフィグレーション方法」を参照し、使用する API 関数のサンプリング点数のマクロ定義を有効にします。

## 9. ライブラリの最適化の効果

- Transform Functions の API 関数は、オリジナルの CMSIS-DSP ライブラリに比べて、平均して 40%以上(\*1)のパフォーマンス向上が見込めます。
- 上記以外の API 関数についても、オリジナルの CMSIS-DSP ライブラリに比べて、平均して 10%以上(\*1)のパフォーマンス向上が見込めます。

\*1 Renesas GCC コンパイラ で CPU クロック周波数 450MHz 以上の場合です。

## 10. ライセンス

All pre-build libraries contained in the folders "ARM", "GCC" and "IAR" are guided by the following license:

Copyright (C) 2009-2015 ARM Limited.  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of ARM nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 11. 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	2017/06/23	—	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。

- ARM および Cortex は、ARM Limited(またはその子会社)の EU またはその他の国における登録商標です。  
All rights reserved.
- Ethernet およびイーサネットは、富士ゼロックス株式会社の登録商標です。
- IEEE は、the Institute of Electrical and Electronics Engineers, Inc. の登録商標です。
- TRON は” The Real-time Operation system Nucleus” の略称です。
- ITRON は” Industrial TRON” の略称です。
- $\mu$ ITRON は” Micro Industrial TRON” の略称です。
- TRON、ITRON、および $\mu$ ITRON は、特定の商品ないし商品群を指す名称ではありません。
- その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です。

## 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が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、その他の不適切に使用しないでください。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）、交通管制（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、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、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損害が生じて、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。また、当社製品および技術を、(1)核兵器、化学兵器、生物兵器等の大量破壊兵器およびこれらを運搬することができるミサイル（無人航空機を含みます。）の開発、設計、製造、使用もしくは貯蔵等の目的、(2)通常兵器の開発、設計、製造または使用の目的、または(3)その他の国際的な平和および安全の維持の妨げとなる目的で、自ら使用せず、かつ、第三者に使用、販売、譲渡、輸出、賃貸もしくは使用許諾しないでください。
  10. 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。
  11. お客様の転売、貸与等により、本書（本ご注意書きを含みます。）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、当社は一切その責任を負わず、お客様にかかる使用に基づく当社への請求につき当社を免責いただきます。
  12. 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. 本資料に記載された情報または当社製品に関し、ご不明点がある場合には、当社営業にお問い合わせください。  
注2. 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。

(Rev.3.0-1 2016.11)



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/>