

SH-2, SH-2A

R01AN1355JJ0102

Rev.1.02

固定小数点ライブラリ (Ver. 1.02) 活用ガイド<コンパイラ活用ガイド>

2012.09.11

本ドキュメントでは、固定小数点ライブラリについて説明します。

目次

1. 固定小数点ライブラリ	2
1.1 概要	2
1.2 固定小数点数の形式	2
1.3 提供ライブラリ	3
1.4 使用例	3
1.5 ライブラリ使用時の注意事項	4
2. 固定小数点ライブラリの詳細	5
2.1 "fixmath.h"	5
2.2 各関数の説明	10
3. 性能	19
3.1 測定条件	19
3.2 実行サイクル数	19

1. 固定小数点ライブラリ

1.1 概要

SH-2、SH-2Aでの実数の演算を固定小数点数(*1)のデータとして扱うライブラリを提供します。固定小数点ライブラリは、特に浮動小数点ユニットを持たない CPU において、高速な実数演算をサポートします。

本ライブラリは、小数部が 16、24、29 ビットの固定小数点数の型に対する以下の関数をサポートしています。

- (1) 乗除算
- (2) sin、cos、atan、sqrt の関数
- (3) 浮動小数点数との変換

アプリケーションの要求する精度に合わせて 16 ビットまたは 24 ビットの精度を選択してください。29 ビット精度は、三角関数を使用するときに必要な $-\pi \sim +\pi$ の範囲を表現できる最大の精度を提供しており、精度の高い演算が必要な場合に有用です。

また、固定小数点では、浮動小数点とことなり、表現できる値の範囲が限定されるため、演算の途中で入力値、出力値の範囲を表現できる型を使用する必要があります。このため、本ライブラリでは、乗除算、型変換に対しては小数部が 1 ビットから 31 ビットまでのすべての固定小数点型をサポートしています。

*1：固定小数点数とは、小数点が特定の位置に固定されている数値の表現手法です。

1.2 固定小数点数の形式

本ライブラリでサポートする固定小数点数の基本的な形式は下記の通りです。

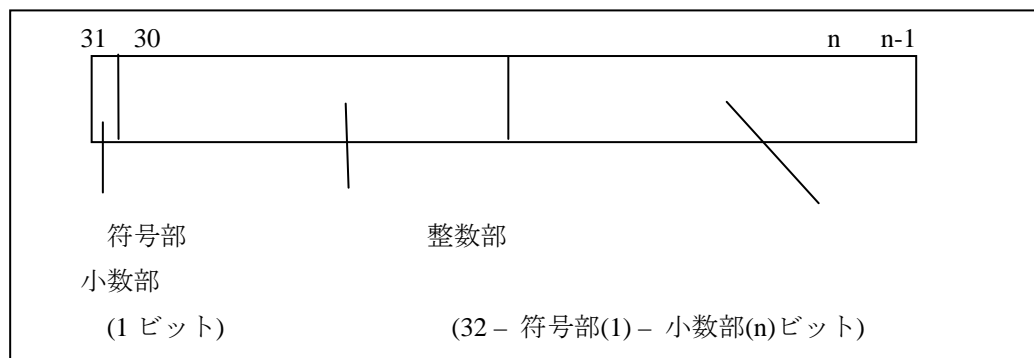


図 1-1 固定小数点数の形式

小数部のビット数に応じて FIX1 から FIX31 までの型をサポートしています。

F I X<n>の n の部分が小数部のビット数を表しています。

また、FIX1 から FIX31 を総称する型として FIX 型をサポートします。この型のデータに対しては、小数部のビット数を指定しない一般的な固定小数点演算をサポートします。

1.3 提供ライブラリ

下表に示すインクルードファイルとライブラリを提供します。

本ライブラリを使用するときは、表 1-1に示すファイルをインクルードしてください。

また、表 1-2に示すように、コンパイラオプションに対応するライブラリをリンクしてください。

表 1-1 固定小数点ライブラリ用のインクルードファイル

ライブラリの種類	内容	インクルードファイル
固定小数点ライブラリ	固定小数点演算を行うライブラリです。	"fixmath.h"

表 1-2 固定小数点ライブラリー覧

ライブラリ名	コンパイラオプション		
	cpu	pic	endian
sh2fix.lib	sh2	0	big
sh2afix.lib	sh2a	0	big

各々、任意のフォルダ下にコピーしてご使用ください。

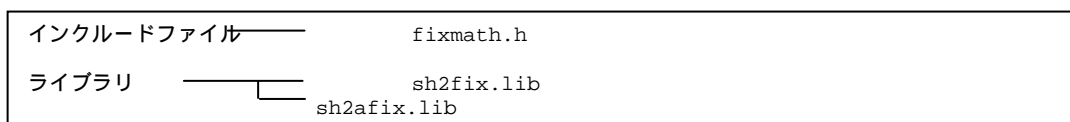


図 1-2 構成例

1.4 使用例

FIX16 型での演算のソース例と High-performance Embedded Workshop でのライブラリ設定方法を示します。

[ソース]

```

#include <stdio.h>
#include "fixmath.h" // 固定小数点ライブラリを使用する場合、インクルード必須

void main()
{
    float r_float;
    FIX16 d_fix16, r_fix16;

    d_fix16 = FIX16_fromfloat(3.14f); // float 型から FIX16 型へ変換
    r_fix16 = FIX16_sin(d_fix16); // 正弦値を求める
    r_float = FIX16_tofloat(r_fix16); // FIX16 型から float 型へ変換
    printf("%f\n", r_float);
}
  
```

[High-performance Embedded Workshop でのライブラリの設定方法]

[ビルド]メニューの[SuperH RISC engine Standard Toolchain]を選択して表示される、[SuperH RISC engine Standard Toolchain]ダイアログボックスにて、[最適化リンカ]タブを選択して、[カテゴリ]に“入力”、[オプション項目]に“ライブラリファイル”を選択し、[追加]ボタンをクリックしてください。表示された[Add library file]ダイアログボックスで、リンクするライブラリを指定してください。

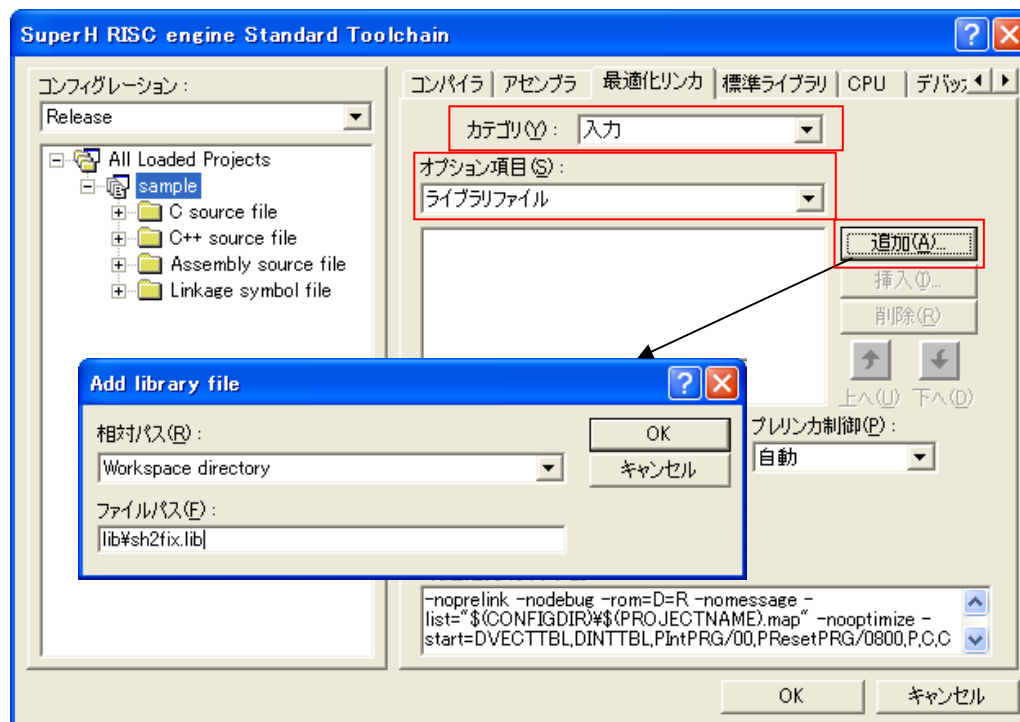


図 1-3 ライブラリの設定方法

1.5 ライブラリ使用時の注意事項

演算結果や変換結果が値の範囲を超えた場合、結果は保証されません。

2. 固定小数点ライブラリの詳細

2.1 "fixmath.h"

固定小数点数値の各種演算を行います。

本ファイルで定義している型およびサポート関数の対応一覧を表 2-1に示します。

記法: 型、関数またはマクロ名の中の <n> は、1 から 31 までの数字を表し、FIX 型的小数点以下のビット数を表します。関数またはマクロ名の中の数字は、型名の中の数字と対応しています。

表 2-1 型およびサポート関数の対応一覧

型	サポート関数及び関数 (マクロ)
FIX1~FIX31 (FIX16, FIX24, FIX29 を除く)	FIX<n>_mul_short, FIX<n>_mul, FIX<n>_div, FIX<n>_tofloat, FIX<n>_fromfloat, FIX<n>_todouble, FIX<n>_fromdouble, FIX<n>_mul_frac, FIX<n>_mul_sat
FIX16, FIX24, FIX29	FIX<n>_mul_short, FIX<n>_mul, FIX<n>_div, FIX<n>_tofloat, FIX<n>_fromfloat, FIX<n>_todouble, FIX<n>_fromdouble, FIX<n>_mul_frac, FIX<n>_mul_sat, FIX<n>_sin, FIX<n>_cos, FIX<n>_atan, FIX<n>_sqrt
FIX	FIX_mul_scale<n>, FIX_mul_frac_scale<n>, FIX_mul_sat_scale<n>, FIX_mul_scale, FIX_mul_frac_scale, FIX_mul_sat_scale

内部的には long 型として定義しています。

固定小数点型 FIX<n> の演算で、オペランドの型と結果の型が一致する場合は、その型に対応する関数を使用します。一方、オペランドの型どうし、あるいはオペランドと結果の型が異なる場合は総称固定小数点型 FIX に対する関数を用いてください。

[固定小数点プログラミングのヒント]

- (1) アプリケーションが必要とする精度に合わせて、FIX16 または FIX24 を選択してください。
- (2) 固定小数点は、浮動小数点に対して表現できる範囲が限定されます。入力値の範囲、演算の途中結果の範囲、必要になる演算の精度に合わせて適切な型を選択する必要がある場合があります。
- (3) 異なる型の固定小数点の間の変換は、C 言語のシフト演算で行ってください。

例: FIX16 から FIX24 への変換.

```
FIX16 x, FIX24 y;
x=y>>8;
```

(4) 同じ型の固定小数点の間の加減算は、C 言語の加減算で行ってください。

例: FIX16 同士の加算

```
FIX16 x, y, z;
```

```
z=x+y;
```

(5) 浮動小数点と固定小数点との間の変換は必要な場合だけ実行してください。固定小数点による高速化の効果が得られません。ただし、プログラム内で固定小数点定数を指定するために変換関数を使用する場合は、マクロにより定数式に変換されるため、オーバーヘッドはありません。

例: 固定小数点の定数指定。

```
FIX16 x;
```

```
x=FIX16_fromfloat(3.14f);
```

各々の型の内部表現を表 2-2に示します。

表 2-2 固定小数点数型の内部表現

型	サイズ (byte)	アライメント数 (byte)	符号の有無	値の範囲	
				最小値	最大値
FIX1	4	4	有	$-2^{30}(-1073741824.0)$	$2^{30} \cdot 2^{-1}(1073741823.5)$
FIX2	4	4	有	$-2^{29}(-536870912.0)$	$2^{29} \cdot 2^{-2}(536870911.75)$
FIX3	4	4	有	$-2^{28}(-268435456.0)$	$2^{28} \cdot 2^{-3}(268435455.875)$
FIX4	4	4	有	$-2^{27}(-134217728.0)$	$2^{27} \cdot 2^{-4}(134217727.9375)$
FIX5	4	4	有	$-2^{26}(-67108864.0)$	$2^{26} \cdot 2^{-5}(67108863.96875)$
FIX6	4	4	有	$-2^{25}(-33554432.0)$	$2^{25} \cdot 2^{-6}(33554431.984375)$
FIX7	4	4	有	$-2^{24}(-16777216.0)$	$2^{24} \cdot 2^{-7}(16777215.9921875)$
FIX8	4	4	有	$-2^{23}(-8388608.0)$	$2^{23} \cdot 2^{-8}(8388607.99609375)$
FIX9	4	4	有	$-2^{22}(-4194304.0)$	$2^{22} \cdot 2^{-9}(4194303.998046875)$
FIX10	4	4	有	$-2^{21}(-2097152.0)$	$2^{21} \cdot 2^{-10}(2097151.9990234375)$
FIX11	4	4	有	$-2^{20}(-1048576.0)$	$2^{20} \cdot 2^{-11}(1048575.99951171875)$
FIX12	4	4	有	$-2^{19}(-524288.0)$	$2^{19} \cdot 2^{-12}(524287.999755859375)$
FIX13	4	4	有	$-2^{18}(-262144.0)$	$2^{18} \cdot 2^{-13}(262143.9998779296875)$
FIX14	4	4	有	$-2^{17}(-131072.0)$	$2^{17} \cdot 2^{-14}(131071.99993896484375)$
FIX15	4	4	有	$-2^{16}(-65536.0)$	$2^{16} \cdot 2^{-15}(65535.999969482421875)$
FIX16	4	4	有	$-2^{15}(-32768.0)$	$2^{15} \cdot 2^{-16}(32767.9999847412109375)$
FIX17	4	4	有	$-2^{14}(-16384.0)$	$2^{14} \cdot 2^{-17}(16383.99999237060546875)$
FIX18	4	4	有	$-2^{13}(-8192.0)$	$2^{13} \cdot 2^{-18}(8191.999996185302734375)$
FIX19	4	4	有	$-2^{12}(-4096.0)$	$2^{12} \cdot 2^{-19}(4095.9999980926513671875)$
FIX20	4	4	有	$-2^{11}(-2048.0)$	$2^{11} \cdot 2^{-20}(2047.99999904632568359375)$
FIX21	4	4	有	$-2^{10}(-1024.0)$	$2^{10} \cdot 2^{-21}(1023.999999523162841796875)$
FIX22	4	4	有	$-2^9(-512.0)$	$2^9 \cdot 2^{-22}(511.9999997615814208984375)$
FIX23	4	4	有	$-2^8(-256.0)$	$2^8 \cdot 2^{-23}(255.99999988079071044921875)$
FIX24	4	4	有	$-2^7(-128.0)$	$2^7 \cdot 2^{-24}(127.999999940395355224609375)$
FIX25	4	4	有	$-2^6(-64.0)$	$2^6 \cdot 2^{-25}(63.9999999701976776123046875)$
FIX26	4	4	有	$-2^5(-32.0)$	$2^5 \cdot 2^{-26}(31.99999998509883880615234375)$
FIX27	4	4	有	$-2^4(-16.0)$	$2^4 \cdot 2^{-27}(15.999999992549419403076171875)$
FIX28	4	4	有	$-2^3(-8.0)$	$2^3 \cdot 2^{-28}(7.9999999962747097015380859375)$
FIX29	4	4	有	$-2^2(-4.0)$	$2^2 \cdot 2^{-29}(3.99999999813735485076904296875)$
FIX30	4	4	有	$-2^1(-2.0)$	$2^1 \cdot 2^{-30}(1.999999999068677425384521484375)$
FIX31	4	4	有	$-2^0(-1.0)$	$2^0 \cdot 2^{-31}(0.9999999995343387126922607421875)$
FIX	4	4	有	想定する小数点以下のビット数により上記のどれかの範囲をとります。	

関数（マクロ）の一覧を表 2-3に示します。

表 2-3 関数（マクロ）一覧

項目	関数（マクロ）名	引数の型	リターン値の型	説明
乗算	FIX<n>_mul_short	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点値の乗算を計算します。 (演算の途中結果が 32 ビット以上のものは結果が保証されません)
	FIX<n>_mul_frac	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点値の乗算結果の小数部分 f (0<=f<1.0) を求めます。
	FIX<n>_mul_sat	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点値の乗算を計算し、オーバーフローした場合は最大値または最小値を返します。
除算	FIX<n>_div	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点の除算を計算します。
変換	FIX<n>_tofloat	FIX<n> n=1~31	FIX<n> n=1~31	FIX<n> 型を float 型へ変換します。
	FIX<n>_fromfloat	FIX<n> n=1~31	FIX<n> n=1~31	float 型を FIX<n>型へ変換します。
	FIX<n>_todouble	FIX<n> n=1~31	FIX<n> n=1~31	FIX<n>型を double 型へ変換します。
	FIX<n>_fromdouble	FIX<n> n=1~31	FIX<n> n=1~31	double 型を FIX<n>型へ変換します。
総称固定小数点の乗算	FIX_mul_scale<n>	FIX	FIX	総称固定小数点の乗算を計算します。
	FIX_mul_frac_scale<n>	FIX	FIX	総称固定小数点の乗算結果の小数部分を計算します。
	FIX_mul_sat_scale<n>	FIX	FIX	総称固定小数点の乗算を計算し、オーバーフローの場合は最大値または最小値を返します。
	FIX_mul_frac_scale	FIX	FIX	総称固定小数点の乗算結果の小数部分を計算します。

値の範囲を超える演算結果は不定となります。値の範囲を超えないようご注意ください。

関数の一覧を表 2-4に示します。

表 2-4 関数一覧

項目	関数名	引数の型	リターン値の型	説明
乗算	FIX<n>_mul	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点値の乗算を計算します。
正弦関数	FIX<n>_sin	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点値のラジアン値の正弦を計算します。
余弦関数	FIX<n>_cos	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点値のラジアン値の余弦を計算します。
逆正接関数	FIX<n>_atan	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点値のラジアン値の逆正接を計算します。
平方根関数	FIX<n>_sqrt	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点値の正の平方根を計算します。
総称固定小数点の乗算	FIX_mul_scale	FIX	FIX	総称固定小数点の乗算を計算します。
	FIX_mul_sat_scale	FIX	FIX	総称固定小数点の乗算を計算し、オーバーフローの場合は最大値または最小値を返します。

値の範囲を超える演算結果は不定となります。値の範囲を超えないようご注意ください。

2.2 各関数の説明

2.2.1 乗算関数 (マクロ)

【書式】 `FIX<n> FIX<n>_mul_short(FIX<n> x, FIX<n> y)`
n:1~31

【説明】 2つの32ビットデータの乗算結果を、nビット右シフトをすることで、
`FIX<n>` 型の固定小数点数の乗算をします。

【ヘッダ】 "fixmath.h"

【リターン値】 乗算結果

【引数】 x 固定小数点値

y 固定小数点値

【例】

```
#include "fixmath.h"
FIX16 x, y, ret;
ret = FIX16_mul_short( x, y );
```

【備考】 演算の途中結果が32ビット以上のものは結果が保証されません。

2.2.2 除算関数

【書式】 `FIX<n> FIX<n>_div(FIX<n> x, FIX<n> y)`
n:1~31

【説明】 固定小数点値の除算を計算します。

【ヘッダ】 "fixmath.h"

【リターン値】 除数結果

【引数】 x 被除数

y 除数

【例】

```
#include "fixmath.h"
FIX16 x, y, ret;
ret = FIX16_div( x, y );
```

2.2.3 変換関数 (マクロ)

(1)float 型→固定小数点型変換

【書式】 `FIX<n> FIX<n>_fromfloat(float x)`

n:1~31

【説明】 float 型を固定小数点型へ変換します。

【ヘッダ】 "fixmath.h"

【リターン値】 変換結果

【引数】 x 変換元データ

【例】 `#include "fixmath.h"`

```
float x;  
FIX16 ret;  
ret = FIX16_fromfloat( x );
```

(2)double 型→固定小数点型変換

【書式】 `FIX<n> FIX<n>_fromdouble(double x)`

n: 1~31

【説明】 double 型を固定小数点型へ変換します。

【ヘッダ】 "fixmath.h"

【リターン値】 変換結果

【引数】 x 変換元データ

【例】 `#include "fixmath.h"`

```
double x;  
FIX16 ret;  
ret = FIX16_fromdouble( x );
```

(3)固定小数点型→float 型変換

【書式】 `float FIX<n>_tofloat(FIX<n> x)`

n: 1~31

【説明】 固定小数点型を float 型へ変換します。

【ヘッダ】 "fixmath.h"

【リターン値】 変換結果

【引数】 x 変換元データ

【例】 `#include "fixmath.h"`

```
FIX16 x;  
float ret;  
ret = FIX16_tofloat( x );
```

(4)固定小数点型→double 型変換

【書式】 `double FIX<n>_todouble(FIX<n> x)`

n: 1~31

【説明】 固定小数点型を double 型へ変換します。

【ヘッダ】 "fixmath.h"

【リターン値】 変換結果

【引数】 x 変換元データ

【例】 `#include "fixmath.h"`

```
FIX16 x;  
double ret;  
ret = FIX16_todouble( x );
```

2.2.4 乗算関数

【書式】 `FIX<n> FIX<n>_mul(FIX<n> x, FIX<n> y)`

n:1~31

【説明】 固定小数点値の乗算を計算します。

【ヘッダ】 "fixmath.h"

【リターン値】 乗算結果

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX16 x, y, ret;  
ret = FIX16_mul( x , y );
```

2.2.5 正弦関数

【書式】 `FIX<n> FIX<n>_sin(FIX<n> x)`

n:16, 24, 29

【説明】 固定小数点値のラジアン値の正弦を計算します。

【ヘッダ】 "fixmath.h"

【リターン値】 x の正弦値

【引数】 x 正弦を求めるラジアン値

【例】 `#include "fixmath.h"`

```
FIX16 x, ret;  
ret = FIX16_sin( x );
```

2.2.6 余弦関数

【書式】 `FIX<n> FIX<n>_cos(FIX<n> x)`

n:16, 24, 29

【説明】 固定小数点値のラジアン値の余弦を計算します。

【ヘッダ】 "fixmath.h"

【リターン値】 x の余弦値

【引数】 x 余弦を求めるラジアン値

【例】 `#include "fixmath.h"`

```
FIX16 x, ret;
```

```
ret = FIX16_cos( x );
```

2.2.7 逆正接関数

【書式】 `FIX<n> FIX<n>_atan(FIX<n> x)`

n:16, 24, 29

【説明】 固定小数点値のラジアン値の逆正接を計算します。

【ヘッダ】 "fixmath.h"

【リターン値】 x の逆正接値

【引数】 x 逆正接を求めるラジアン値

【例】 `#include "fixmath.h"`

```
FIX16 x, ret;
ret = FIX16_atan( x );
```

2.2.8 平方根関数

【書式】 `FIX<n> FIX<n>_sqrt(FIX<n> x)`

n:16, 24, 29

【説明】 固定小数点値の正の平方根を計算します。

【ヘッダ】 "fixmath.h"

【リターン値】 x の正の平方根の値

【引数】 x 正の平方根を求める固定小数点値

【例】 `#include "fixmath.h"`

```
FIX16 x, ret;
ret = FIX16_sqrt( x );
```

【備考】 x が負の値の場合、結果は保証されません。

2.2.9 乗算 (小数部分) (マクロ)

【書式】 `FIX<n> FIX<n>_mul_frac(FIX<n> x, FIX<n> y)`

n: 1~31

【説明】 固定小数点値の乗算結果の小数部分を計算します。結果の値は常に正 ($0 \leq \text{結果} < 1.0$) になります。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積の小数部分。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX16 x, y, ret;
ret = FIX16_mul_frac( x, y );
```

2.2.10 乗算 (飽和演算) (マクロ)

【書式】 `FIX<n> FIX<n>_mul_sat(FIX<n> x, FIX<n> y)`

n: 1~31

【説明】 固定小数点値の乗算を計算します。結果がオーバーフローした場合、結果の符合にしたがって最大値または最小値を返します。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (飽和演算)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX16 x, y, ret;  
ret = FIX16_mul_sat( x, y );
```

2.2.11 乗算 (FIX 型) (マクロ)

【書式】 `FIX FIX_mul_scale<n>(FIX x, FIX y)`

n: 1~31

【説明】 総称固定小数点値の乗算を計算します。x と y を long 型とみなした場合、x と y の積を右に n ビットシフトした値を求めます。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (FIX 型)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX x, y, ret;  
ret = FIX_mul_scale16( x, y );
```

【備考】 結果の値が 32 ビットで表現できないとき、その値は保証されません。

`FIX<n1>`, `FIX<n2>` 型を乗算して `FIX<n3>` 型を求める場合は、シフトするビット数は `n1+n2-n3` になりますので、n としてこの値 (1~31 の範囲) を指定してください。

2.2.12 乗算関数 (FIX 型)

【書式】 `FIX FIX_mul_scale(FIX x, FIX y, int n)`

n: 1~31

【説明】 総称固定小数点値の乗算を計算します。x と y を long 型とみなした場合、x と y の積を右に n ビットシフトした値を求めます。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (FIX 型)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX x, y, ret;  
int n=16;  
ret = FIX_mul_scale ( x, y, n );
```

【備考】 結果の値が 32 ビットで表現できないとき、その値は保証されません。

`FIX<n1>`, `FIX<n2>` 型を乗算して `FIX<n3>` 型を求める場合は、シフトするビット数は $n_1+n_2-n_3$ になりますので、n としてこの値 (1~31 の範囲) を指定してください。

2.2.13 乗算 (小数部分、FIX 型) (マクロ)

【書式】 `FIX FIX_mul_frac_scale<n>(FIX x, FIX y)`

n: 1~31

【説明】 総称固定小数点値の乗算結果の小数部分を計算します。x と y を long 型とみなした場合、x と y の積を右に n ビットシフトした値の下位 n ビットを求めます。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (FIX 型)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX x, y, ret;  
ret = FIX_mul_frac_scale16( x, y );
```

【備考】 `FIX<n+d>`, `FIX<n-d>` 型を乗算して `FIX<n>` 型を求める場合に使用することができます。

2.2.14 乗算 (小数部分、FIX 型) (マクロ)

【書式】 `FIX FIX_mul_frac_scale(FIX x, FIX y, int n)`

n: 1~31

【説明】 総称固定小数点値の乗算結果の小数部分を計算します。x と y を long 型とみなした場合、x と y の積を右に n ビットシフトした値の下位 n ビットを求めます。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (FIX 型)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX x, y, ret;
```

```
int n=16;
```

```
ret = FIX_mul_frac_scale ( x, y, n );
```

【備考】 `FIX<n+d>`, `FIX<n-d>` 型を乗算して `FIX<n>` 型を求める場合に使用することができます。

2.2.15 乗算 (飽和演算、FIX 型) (マクロ)

【書式】 `FIX FIX_mul_sat_scale<n>(FIX x, FIX y)`

n: 1~31

【説明】 総称固定小数点値の乗算を計算します。x と y を long 型とみなした場合、x と y の積を右に n ビットシフトした値を求めます。結果がオーバーフローした場合、結果の符合にしたがって最大値または最小値を返します。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (FIX 型)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX x, y, ret;
```

```
ret = FIX_mul_sat_scale16( x, y );
```

【備考】 結果の値が 32 ビットで表現できないとき、その値は保証されません。

`FIX<n1>`, `FIX<n2>` 型を乗算して `FIX<n3>` 型を求める場合は、シフトするビット数は `n1+n2-n3` になりますので、n としてこの値 (1~31 の範囲) を指定してください。

2.2.16 乗算関数 (FIX 型)

【書式】 `FIX FIX_mul_sat_scale(FIX x, FIX y, int n)`

n: 1~31

【説明】 総称固定小数点値の乗算を計算します。x と y を long 型とみなした場合、x と y の積を右に n ビットシフトした値を求めます。結果がオーバーフローした場合、結果の符合にしたがって最大値または最小値を返します。

【ヘッダ】 "fixmath.h"

【リターン値】 x と y の積 (FIX 型)。

【引数】 x 固定小数点値

y 固定小数点値

【例】 `#include "fixmath.h"`

```
FIX x, y, ret;  
int n=16;  
ret = FIX_mul_sat_scale ( x, y, n );
```

【備考】 結果の値が 32 ビットで表現できないとき、その値は保証されません。

FIX<n₁>, FIX<n₂> 型を乗算して FIX<n₃> 型を求める場合は、シフトするビット数は n₁+n₂-n₃ になりますので、n としてこの値 (1~31 の範囲) を指定してください。

3. 性能

3.1 測定条件

コンパイラ : SuperH RISC engine C/C++ コンパイラ V.9.03.00

ビルド条件 : 表 3-1に示す条件でライブラリを構築

表 3-1 ライブラリ構築時条件

条件	ライブラリ構築時のオプション		
	cpu	pic	endian
1	sh2	0	big
2	sh2a	0	big

3.2 実行サイクル数

固定小数点ライブラリの演算速度を表 3-2に示します。

表 3-2 固定小数点ライブラリの演算速度

CPU		SH-2	SH-2A
ライブラリ構築時条件		1	2
正弦関数	FIX16_sin	105	64
	FIX24_sin	104	64
	FIX29_sin	95	63
余弦関数	FIX16_cos	103	64
	FIX24_cos	105	63
	FIX29_cos	100	69
逆正接関数	FIX16_atan	213	130
	FIX24_atan	214	131
	FIX29_atan	212	133
平方根関数	FIX16_sqrt	179	99
	FIX24_sqrt	179	100
	FIX29_sqrt	177	102

【注】単位は Cycle. 演算速度の測定値には誤差が含まれています。

本ライブラリの演算結果の最大誤差は最下位ビットで±2です。

ホームページとサポート窓口<website and support,ws>

ルネサス エレクトロニクスホームページ

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

お問合せ先

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

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<revision history,rh>

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.9.1	—	初版発行
1.02	2012.9.11	—	Atan 関数の不具合を修正

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

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. リザーブアドレスのアクセス禁止

【注意】リザーブアドレス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、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違くと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。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。

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・電話番号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 エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2 (日本ビル)

(03)5201-5307

■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/>