
V850E2M シリーズ

R01AN1349JS0100

Rev.1.00

V850E2M 固定小数点ライブラリ (32 レジスタ・モード適用時)

2012.09.26

はじめに

本書は、V850E2M 固定小数点ライブラリの使用方法について説明します。

対象デバイス

V850E2M シリーズ

目次

1. 固定小数点ライブラリ	4
1.1 概要	4
1.2 固定小数点データの書式.....	4
1.3 ライブラリファイル.....	4
1.4 使用例.....	5
1.5 ライブラリ使用に関する注記	7
2. 固定小数点ライブラリの仕様	8
2.1 "fixmath.h"	8
2.2 各関数の説明.....	12
2.2.1 乗算 (マクロ)	12
2.2.2 除算 (マクロ)	12
2.2.3 変換 (マクロ)	13
2.2.4 乗算	15
2.2.5 正弦関数	15
2.2.6 余弦関数	16
2.2.7 逆正接関数.....	16
2.2.8 平方根関数.....	17
2.2.9 乗算 (小数部) (マクロ)	17
2.2.10 乗算 (飽和) (マクロ)	18
2.2.11 乗算 (FIX 型) (マクロ)	18
2.2.12 乗算 (FIX 型)	19
2.2.13 乗算 (小数部、FIX 型) (マクロ)	19
2.2.14 乗算 (小数部、FIX 型) (マクロ)	20
2.2.15 乗算 (飽和、FIX 型) (マクロ)	20
2.2.16 乗算 (飽和、FIX 型)	21
3. 性能と精度	22

3.1 評価条件	22
3.2 実行サイクル数	22
3.3 精度	22

1. 固定小数点ライブラリ

1.1 概要

本ライブラリは、V850E2M シリーズ用の固定小数点形式¹を使用した実数演算を提供するものです。

固定小数点ライブラリにより、特に FPU (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が可能となります。

本ライブラリは、以下に示す、小数部が 16、24、29 ビットの固定小数点型の関数をサポートしています。

1. 乗算と除算
2. 数学関数 (sin、cos、atan、sqrt)
3. 浮動小数点データ間の変換

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 ビットの精度を使用してください。29 ビット精度は、三角関数の入力範囲 (- ~ +) を表現できる最高精度タイプとしてサポートされています。

固定小数点では、浮動小数点に比べて値の範囲が限定されているため、各演算の入力値/出力値に応じて適切な精度を選択する必要があります。このため、本ライブラリでは、固定小数点型のすべての精度 (1 ~ 31) について乗算、除算、変換をサポートしています。

1.2 固定小数点データの書式

本ライブラリがサポートする固定小数点データの書式は、次のとおりです。

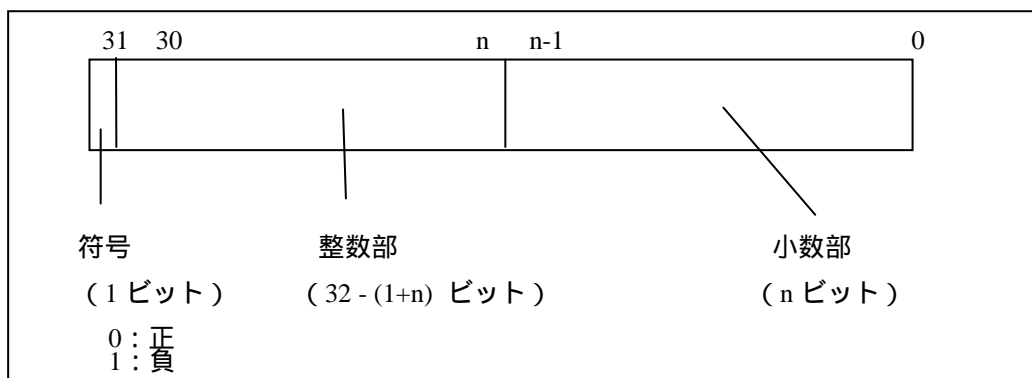


図 1. 固定小数点データの書式

小数部のビット数に応じて FIX1 ~ FIX31 の型がサポートされています。数値は小数部のビット数を表しています。

一般的な固定小数点型 FIX もサポートされています。この型については、一般的な固定小数点演算がサポートされています。

1.3 ライブラリファイル

以下のインクルードファイルとライブラリファイルが提供されています。

本ライブラリを使用するときには、表 1 に示すファイルをインクルードし、表 2 に示す (コンパイラオプションに対応した) ライブラリファイルをリンクしてください。

¹ 固定小数点形式とは、小数点を特定のビット位置に固定し実数を表現する形式です。

表 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	機能	
固定小数点ライブラリ	固定小数点演算を行う	"fixmath.h"

表 2. 固定小数点ライブラリ

ライブラリ名	コンパイラオプション
	cpu
V850E2Mfixmath.lib	V850E2M, 32 register mode

使用する前に、ローカルのインクルードディレクトリまたはライブラリディレクトリにこれらのファイルをコピーしてください。

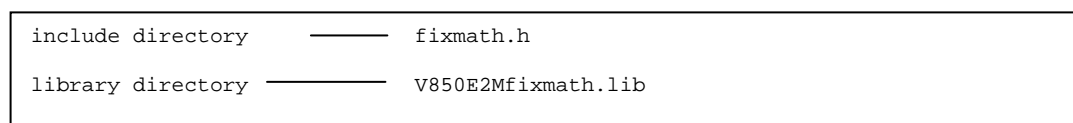


図 2. 構成例

1.4 使用例

以下の例は、FIX16 演算を使用したプログラムと、CubeSuite+でのライブラリの指定方法を示しています。

[ソースプログラム]

```
#include <stdio.h>
#include "fixmath.h"                                     // Necessary when using
                                                         // fixed-point library

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

    d_fix16 = FIX16_fromfloat(3.1415926f/2); // convert float type constant to FIX16
    r_fix16 = FIX16_sin(d_fix16);           // computes sin
    r_flt = FIX16_tofloat(r_fix16);         // Convert back for printing
    printf("%f¥n", r_flt);
}
```

[CubeSuite+でのライブラリの指定方法]

プロジェクトツリーのメニューで[ファイル]を右クリックし、[追加] -> [ファイルを追加]を選択します。[既存のファイルを追加]ダイアログボックスで、ライブラリファイルを選択し、[開く]ボタンを押すと、ライブラリファイルがプロジェクトツリーのメニューに追加されます。

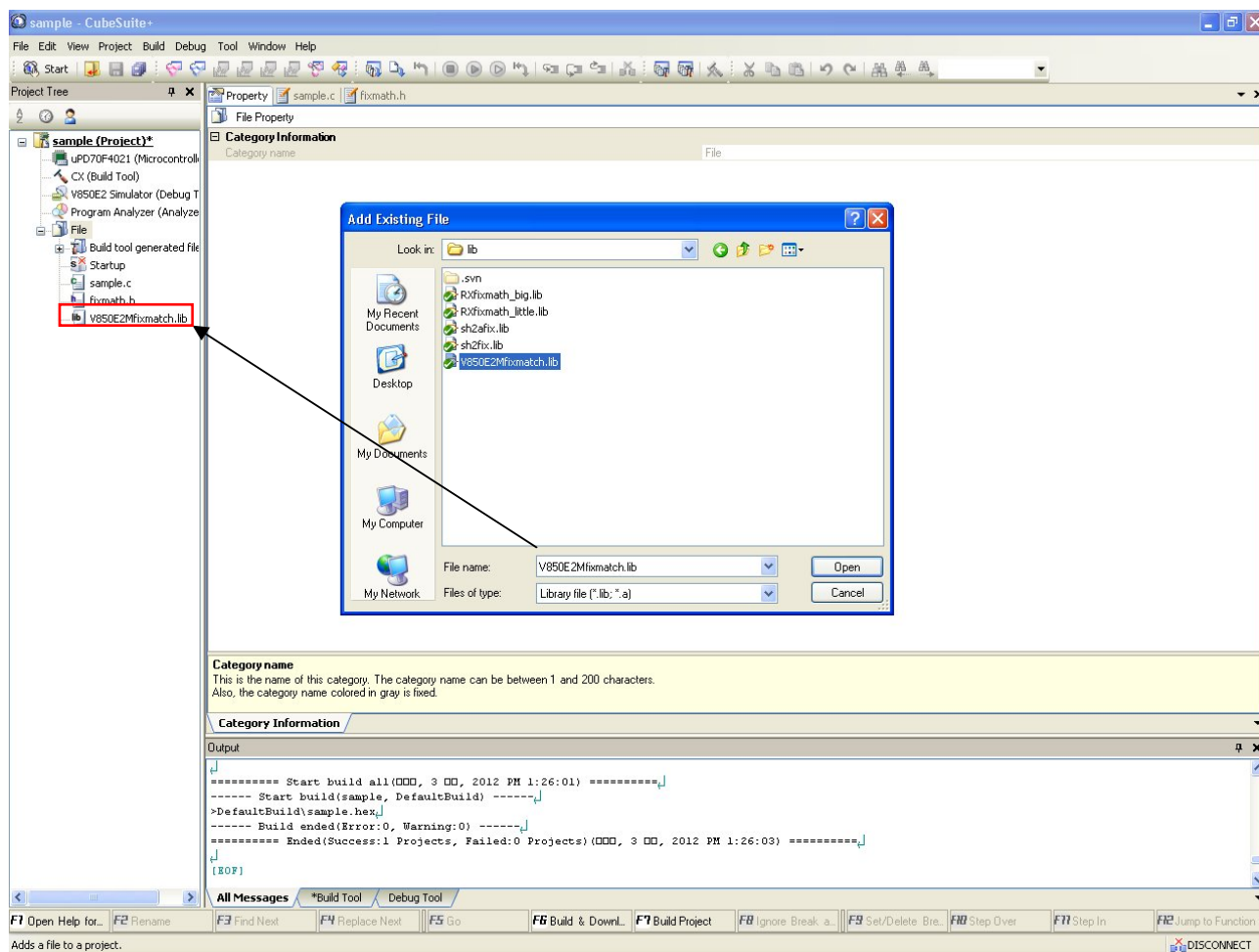


図 3. ライブラリの指定

[CubeSuite+での 32 レジスタ・モード設定方法]

プロジェクトツリーのメニューで[CX (ビルドツール)]をクリックします。

そして、[レジスタ・モード]をクリックし、「32 レジスタ・モード (なし)」が選択されていることを確認します。「32 レジスタ・モード (なし)」が選択されていない場合、「32 レジスタ・モード (なし)」を選択してください。

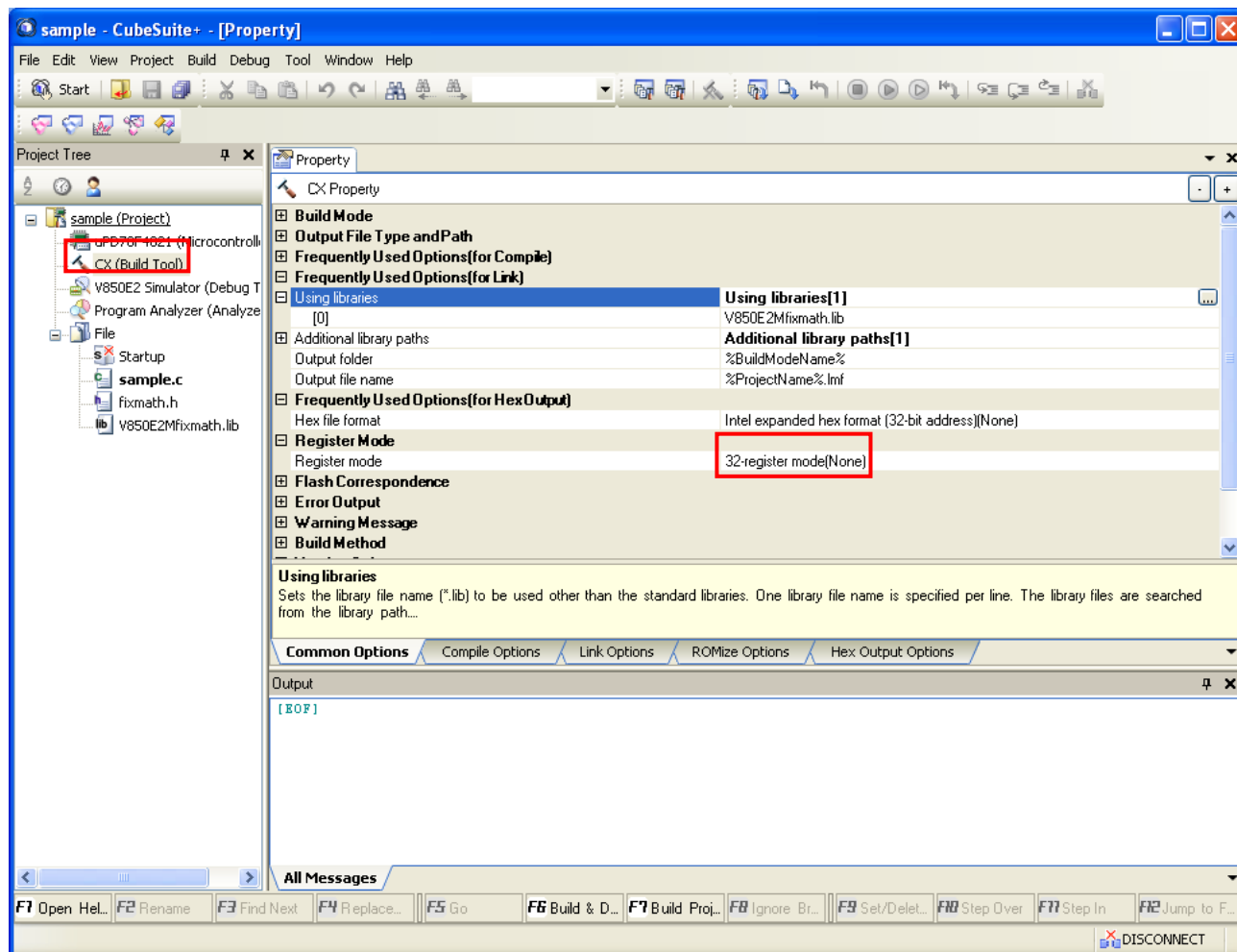


図 4. 32 レジスタ・モードの指定

1.5 ライブラリ使用に関する注記

演算や変換の結果が固定小数点型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仕様

2.1 "fixmath.h"

このヘッダファイルは、固定小数点演算の型と関数を定義しています。

ファイルで定義されている型、およびサポート関数 (マクロ) を表 3 に示します。

表記: 型、関数、またはマクロ名の中の表記<n>は、1~31までの数字を表します。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 3. 型およびサポート関数

型	サポート関数およびマクロ
FIX1-FIX31 (except FIX16, FIX24 or 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
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

これらの型はロング型として定義されています。

オペランドと演算の結果が同じ型 (FIX<n>) であるとき、その型に対応する関数を使用してください。同じ型でなければ、一般固定小数点型 FIX に対応する関数を使用してください。

[固定小数点ライブラリ使用のヒント]

- (1) アプリケーションの要件に従って、標準固定小数点型 (FIX16 または FIX24) を選択してください。
- (2) 固定小数点は、浮動小数点に比べて値の範囲が限定されています。入力の範囲、中間結果、必要な計算精度に合わせて適切な型を選択することをお勧めします。
- (3) 異なる型の固定小数点の間のデータ変換は、C 言語のシフト演算を使用してください。

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データ間の加算または減算は、C 言語の整数加算または整数減算を使用してください。

例: FIX16 の加算

```
FIX16 x, y, z;
z=x+y;
```

- (5) 浮動小数点型と固定小数点型との間の変換は必要となしにのみ実行するようにしてください。不要な変換は効率を低減します。ただし、変換関数を定数に適用すると、マクロの拡張によって定数式が生成されるので、固定小数点の定数は、オーバーヘッドなしで指定することができます。

例: 固定小数点の定数

```
FIX16 x;
x=FIX16_fromfloat(3.14f);
```


固定型の表記と範囲を表 4 に示します。

表 4. 固定型の表記と範囲

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

定義されているマクロのリストを表 5 に示します。

表 5. マクロのリスト

分類	名称	引数型	戻り型	説明
乗算	FIX<n>_mul_short	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点データの乗算を計算します (乗算結果が 32 ビットを超えた場合、結果は保証されません)。
	FIX<n>_mul_frac	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点の乗算の小数部分を計算します ($0 \leq f < 1.0$)。
除算	FIX<n>_div	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点データの除算を計算します。
変換	FIX<n>_tfloat	FIX<n> n=1~31	float	FIX<n>型を float 型に変換します。
	FIX<n>_fromfloat	float	FIX<n> n=1~31	float 型を FIX<n>型に変換します。
	FIX<n>_todouble	FIX<n> n=1~31	double	FIX<n>型を double 型に変換します。
	FIX<n>_fromdouble	double	FIX<n> n=1~31	Double 型を FIX<n>型に変換します。
一般固定小数点の乗算	FIX_mul_scale<n>	FIX	FIX	一般固定小数点データの乗算を計算します。
	FIX_mul_frac_scale<n>	FIX	FIX	一般固定小数点の乗算の小数部分を計算します ($0 \leq f < 1.0$)。
	FIX_mul_sat_scale<n>	FIX	FIX	一般固定小数点の乗算を計算します。オーバーフローが発生したとき、結果は範囲の最大値または最小値になります。
	FIX_mul_frac_scale	FIX	FIX	一般固定小数点の乗算の小数部分を計算します ($0 \leq f < 1.0$)。

演算結果がデータ型の範囲から外れた場合、その値は保証されません。

宣言されている関数のリストを表 6 に示します。

表 6. 関数のリスト

分類	名称	引数型	戻り型	説明
乗算	FIX_mul	FIX	FIX	固定小数点データの乗算を計算します。
	FIX<n>_mul_sat	FIX<n> n=1~31	FIX<n> n=1~31	固定小数点データの乗算を計算します。オーバーフローが発生したとき、結果は範囲の最大値または最小値になります。
正弦	FIX<n>_sin	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点データの正弦を計算します (ラジアン)。
余弦	FIX<n>_cos	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点データの余弦を計算します (ラジアン)。
逆正接	FIX<n>_atan	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点データの逆正接のラジアン値を計算します。
平方根	FIX<n>_sqrt	FIX<n> n=16, 24, 29	FIX<n> n=16, 24, 29	固定小数点データの平方根を計算します。
一般固定小数点の乗算	FIX_mul_scale	FIX	FIX	一般固定小数点データの乗算を計算します。
	FIX_mul_sat_scale	FIX	FIX	一般固定小数点データの乗算を計算します。オーバーフローが発生したとき、結果は範囲の最大値または最小値になります。

演算結果がデータ型の範囲から外れた場合、その値は保証されません。

2.2 各関数の説明

2.2.1 乗算 (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ビットデータを乗算してnビットだけ右にシフトします。これは、FIX<n>型の2つの固定小数点データの乗算を計算します。

[ヘッダ] "fixmath.h"

[戻り値] 乗算の結果

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

[使用例]

```
#include "fixmath.h"
fix16 x, y, ret;
```

```
ret=FIX16_mul_short(x, y);
```

[備考] 短い乗算は32ビット整数演算を使用します。中間結果が32ビットを超えた場合、結果は保証されません。

2.2.2 除算 (マクロ)

[書式] `FIX<n> FIX<n>_div(FIX<n> x, FIX<n> y)`

n: 1~31

[説明] 2つの固定小数点データの商を計算します。

[ヘッダ] "fixmath.h"

[戻り値] 除算の結果

[パラメータ] x: 被除数
y: 除数

[使用例]

```
#include "fixmath.h"
fix16 x, y, ret;
```

```
ret=FIX16_div(x, y);
```

2.2.3 変換 (マクロ)

(1) float 型から固定小数点への変換

[書式] `FIX<n> FIX<n>_fromfloat(float x)`

n: 1~31

[説明] float 型を固定小数点型に変換します。

[ヘッダ] "fixmath.h"

[戻り値] 変換の結果

[パラメータ] x: 変換元データ

```
[使用例] #include "fixmath.h"
float x;
FIX16 ret;

ret=FIX16_fromfloat(x);
```

(2) double 型から固定小数点への変換

[書式] `FIX<n> FIX<n>_fromdouble(double x)`

n: 1~31

[説明] double 型データを固定小数点型に変換します。

[ヘッダ] "fixmath.h"

[戻り値] 変換の結果

[パラメータ] x: 変換元データ

```
[使用例] #include "fixmath.h"
double x;
FIX16 ret;

ret=FIX16_fromdouble(x);
```

(3) 固定小数点型から float 型への変換

[書式] float FIX<n>_tofloat(FIX<n> x)
n: 1~31

[説明] 固定小数点データを float 型に変換します。

[ヘッダ] "fixmath.h"

[戻り値] 変換の結果

[パラメータ] x: 変換元データ

```
[使用例] #include "fixmath.h"
          FIX16 x;
          float ret;

          ret=FIX16_tofloat(x);
```

(3) 固定小数点型から double 型への変換

[書式] double FIX<n>_todouble(FIX<n> x)
n: 1~31

[説明] 固定小数点データを double 型に変換します。

[ヘッダ] "fixmath.h"

[戻り値] 変換の結果

[パラメータ] x: 変換元データ

```
[使用例] #include "fixmath.h"
          FIX16 x;
          double ret;

          ret=FIX16_todouble(x);
```

2.2.4 乗算

[書式] `FIX FIX_mul(FIX x, FIX y, int n)`

n: 1~31

[説明] FIX 型の 2 つの固定小数点データの乗算を計算します。64 ビットの間接結果が使用されます。x と y の小数部が n ビットであると想定し、2 つの固定小数点データの積を計算します。x と y の値を long データとして乗算し、右に n ビットだけシフトします。

[ヘッダ] "fixmath.h"

[戻り値] 乗算の結果

[パラメータ] x: 固定小数点データ
y: 固定小数点データ
n: 小数部のビットサイズ

[使用例]

```
#include "fixmath.h"
FIX16 x, y, ret;
```

```
ret=(FIX16)FIX_mul((FIX)x, (FIX)y, 16);
```

2.2.5 正弦関数

[書式] `FIX<n> FIX<n>_sin(FIX<n> x)`

n: 16, 24, 29

[説明] FIX<n>固定小数点データの正弦関数を計算します (ラジアン値)。

[ヘッダ] "fixmath.h"

[戻り値] 正弦関数の結果

[パラメータ] x: 固定小数点データ (ラジアン)

[使用例]

```
#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] `FIX<n>`固定小数点データの余弦関数を計算します (ラジアン値)。

[ヘッダ] `"fixmath.h"`

[戻り値] 余弦関数の結果

[パラメータ] x: 固定小数点データ (ラジアン)

```
[使用例] #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] `FIX<n>`固定小数点データの逆正接関数を計算します。結果はラジアン値です。

[ヘッダ] `"fixmath.h"`

[戻り値] 逆正接の結果 (ラジアン)

[パラメータ] x: 固定小数点データ

```
[使用例] #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] `FIX<n>`固定小数点データの平方根を計算します。

[ヘッダ] "fixmath.h"

[戻り値] 平方根の結果

[パラメータ] x: 固定小数点データ

```
[使用例] #include "fixmath.h"
          FIX16 x, ret;

          ret=FIX16_sqrt(x);
```

2.2.9 乗算 (小数部) (マクロ)

[書式] `FIX<n> FIX<n>_mul_frac(FIX<n> x, FIX<n> y)`

n: 1~31

[説明] 積の小数部を計算します。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"

[戻り値] 積の小数部

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

```
[使用例] #include "fixmath.h"
          FIX16 x, y, ret;

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

2.2.10 乗算 (飽和) (マクロ)

[書式] `FIX<n> FIX<n>_mul_sat(FIX<n> x, FIX<n> y)`
n: 1~31

[説明] 2つの固定小数点データの積を計算します。結果がオーバーフローしたとき、戻り値は、結果の符号に従って最大値または最小値になります。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

[使用例]

```
#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] 2つの一般固定小数点データの積を計算します。x と y の値を long データとして乗算し、右に n ビットだけシフトします。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

[使用例]

```
#include "fixmath.h"
FIX x, y, ret;

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

[備考] 結果を 32 ビットで表せない場合、その値は保証されません。FIX<n₁>と FIX<n₂>を乗算して FIX<n₃>型を得るときには、シフト数は n₁ + n₂ - n₃ となるため、この値 (1 ~ 31) を <n> で指定することができます。

2.2.12 乗算 (FIX 型)

[書式] `FIX FIX_mul_scale(FIX x, FIX y, int n)`

n: 1~31

[説明] 2つの一般固定小数点データの積を計算します。x と y の値を long データとして乗算し、右に n ビットだけシフトします。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

```
[使用例] #include "fixmath.h"
FIX x, y, ret;
int n=16;

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

[備考] 結果を 32 ビットで表せない場合、その値は保証されません。FIX<n₁>と FIX<n₂>を乗算して FIX<n₃>型を得るときには、シフト数は n₁ + n₂ - n₃ となるため、この値 (1 ~ 31) を <n> で指定することができます。

2.2.13 乗算 (小数部、FIX 型) (マクロ)

[書式] `FIX FIX_mul_frac_scale<n>(FIX x, FIX y)`

n: 1~31

[説明] 2つの一般固定小数点データの積の小数部を計算します。x と y の値を long データとして乗算し、右に n ビットだけシフトして下位の n ビットを返します。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

```
[使用例] #include "fixmath.h"
FIX x, y, ret;

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

[備考] この関数を使用すると、FIX<n+d>型と FIX<n-d>型を乗算することによって FIX<n>型の小数部を計算することができます。

2.2.14 乗算 (小数部、FIX 型) (マクロ)

[書式] `FIX FIX_mul_frac_scale(FIX x, FIX y, int n)`

n: 1~31

[説明] 2つの一般固定小数点データの積の小数部を計算します。xとyの値をlongデータとして乗算し、右にnビットだけシフトして下位のnビットを返します。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

```
[使用例] #include "fixmath.h"
FIX x, y, ret;
int n=16;

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

[備考] この関数を使用すると、FIX<n+d>型とFIX<n-d>型を乗算することによってFIX<n>型の小数部を計算することができます。

2.2.15 乗算 (飽和、FIX 型) (マクロ)

[書式] `FIX FIX_mul_sat_scale<n>(FIX x, FIX y)`

n: 1~31

[説明] 2つの一般固定小数点データの積を計算します。xとyの値をlongデータとして乗算し、右にnビットだけシフトします。結果がオーバフローしたとき、戻り値は、結果の符号に従って最大値または最小値になります。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

```
[使用例] #include "fixmath.h"
FIX x, y, ret;

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

[備考] 結果を32ビットで表せない場合、その値は保証されません。FIX<n₁>とFIX<n₂>を乗算してFIX<n₃>型を得るときには、シフト数はn₁+n₂-n₃となるため、この値(1~31)を<n>で指定することができます。

2.2.16 乗算 (飽和、FIX 型)

[書式] `FIX FIX_mul_sat_scale(FIX x, FIX y, int n)`

n: 1~31

[説明] 2つの一般固定小数点データの積を計算します。x と y の値を long データとして乗算し、右に n ビットだけシフトします。結果がオーバーフローしたとき、戻り値は、結果の符号に従って最大値または最小値になります。

[ヘッダ] "fixmath.h"

[戻り値] 固定小数点データの積

[パラメータ] x: 固定小数点データ
y: 固定小数点データ

[使用例]

```
#include "fixmath.h"
FIX x, y, ret;
int n=16;

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

[備考] 結果を 32 ビットで表せない場合、その値は保証されません。FIX<n₁>と FIX<n₂>を乗算して FIX<n₃>型を得るときには、シフト数は n₁+n₂-n₃ となるため、この値 (1~31) を <n> で指定することができます。

3. 性能と精度

3.1 評価条件

コンパイラ：CX V1.21

3.2 実行サイクル数

固定小数点の数学関数の実行サイクル数を表 7 に示します。

表 7. 固定小数点の数学関数の実行サイクル数

	CPU	V850E2M
正弦	FIX16_sin	67
	FIX24_sin	67
	FIX29_sin	64
余弦	FIX16_cos	61
	FIX24_cos	61
	FIX29_cos	63
逆正接	FIX16_atan	120
	FIX24_atan	120
	FIX29_atan	120
平方根	FIX16_sqrt	75
	FIX24_sqrt	75
	FIX29_sqrt	75

[注記] 数値の単位はサイクル数です。測定値には誤差が含まれています。

3.3 精度

数学関数の最大誤差は、FIX29_sqrt を除き、最下位ビットで ± 2 です。FIX29_sqrt の精度は、最下位ビットで ± 3 です。

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/inquiry>

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	2012.09.26	—	初版発行。32 レジスタ・モードのみ対応。

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に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、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・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/>