

# RH850コンパイラ（CC-RH）

## 言語仕様

2016年 11月 2日

Rev.1.01

ソフトウェア事業部 ソフトウェア技術部  
ルネサス システムデザイン株式会社

R20UT3238JJ0101

# 整数型の仕様

型	値の範囲	サイズ
char (signed char型)	-128 ~ 127	1バイト
unsigned char	0 ~ 255	1バイト
short	-32768 ~ 32767	2バイト
unsigned short	0 ~ 65535	2バイト
int	-2147483648 ~ 2147483647	4バイト
unsigned int	0 ~ 4294967295	4バイト
long	-2147483648 ~ 2147483647	4バイト
unsigned long	0 ~ 4294967295	4バイト
long long	-9223372036854775808 ~ 9223372036854775807	8バイト
unsigned long long	0 ~ 18446744073709551615	8バイト

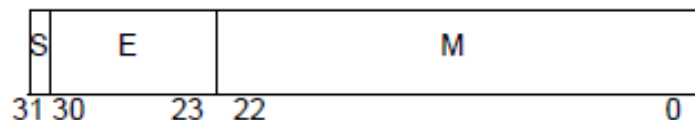
# 浮動小数点型の仕様

浮動小数点型データの内部表現は、IEEE754 に準拠しています。

IEEE754 とは、浮動小数点演算を扱うシステムにおいて、扱うデータ形式や数値範囲などの仕様の統一化を図った標準仕様です。

型	仕様	値の範囲	サイズ
float	単精度形式	1.17549435E-38F ~ 3.40282347E+38F	4バイト
double	倍精度形式	2.2250738585072014E-308 ~ 1.7976931348623158E+308	8バイト
long double	倍精度形式	2.2250738585072014E-308 ~ 1.7976931348623158E+308	8バイト

float

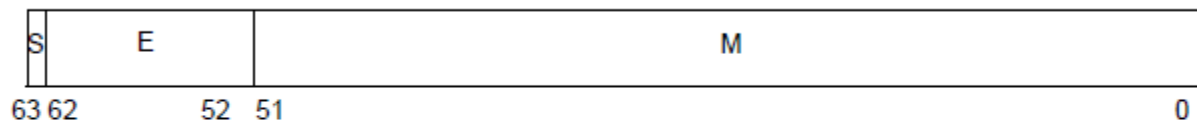


S : 仮数部の符号ビット

E : 指数部 (8 ビット)

M : 仮数部 (23 ビット)

double, long double



S : 仮数部の符号ビット

E : 指数部 (11 ビット)

M : 仮数部 (52 ビット)

# その他データ型の仕様

---

データ	サイズ
ポインタ	4バイト (unsigned intの内部表現と同じ)
列挙型(enum)	4バイト (signed intの内部表現と同じ)
_Bool	1バイト (ゼロビット目のみが意味を持つ)

(※) -Xenum\_type=auto オプション指定時は、各列挙型について、その型のすべての列挙子の値を表現可能な最小の整数型として扱います。

# 整列条件

型	サイズ	整列条件
char	1バイト	1バイト
signed char	1バイト	
unsigned char	1バイト	
short	2バイト	2バイト
unsigned short	2バイト	
int	4バイト	4バイト
unsigned int	4バイト	
long	4バイト	
unsigned long	4バイト	
long long	8バイト	
unsigned long long	8バイト	
enum	4バイト	
float	4バイト	
double	8バイト	
long double	8バイト	
ポインタ	4バイト	

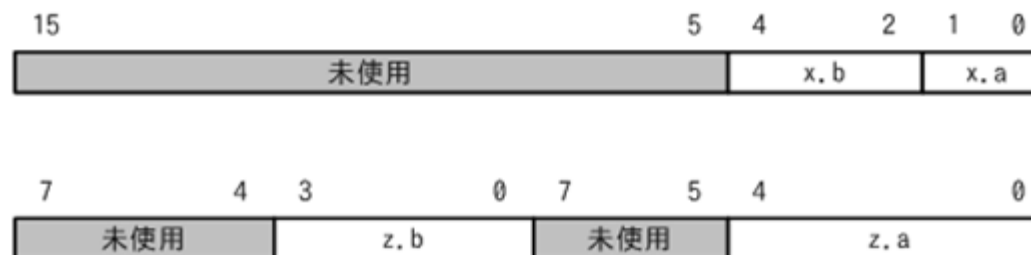
# ビット・フィールドの仕様

- 符号指定のないビットフィールドは符号付き型で扱う
- デフォルトではビットフィールドを下位ビット（右側）から割り付ける
- -Xbit\_order=left オプション、または#pragma bit\_order left を指定することで、ビット・フィールド・メンバを上位ビット側から割り当てることも可能

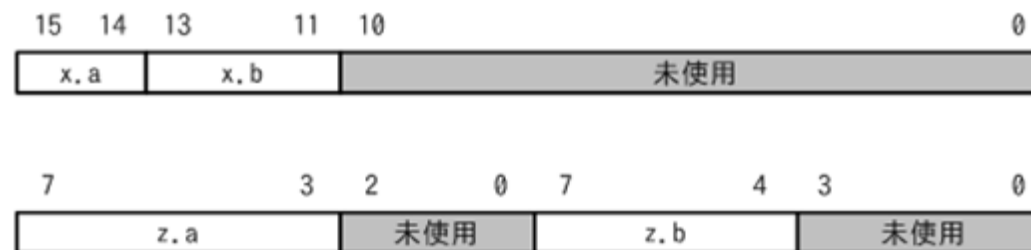
```
struct b1 {  
    short a:2;  
    short b:3;  
} x;
```

```
struct b3 {  
    char a:5;  
    char b:4;  
} z;
```

下位ビット(右側)  
割り付け



上位ビット(左側)  
割り付け



# エンディアン

RH850はリトルエンディアン固定

char a = 0x01;      0 0x01 7      ビッグエンディアン

0 0x01 7      リトルエンディアン

short b = 0x0123;      0 0x01 0x23 15      ビッグエンディアン

0 0x23 0x01 15      リトルエンディアン

int c = 0x01234567;      0 0x01 0x23 0x45 0x67 31      ビッグエンディアン

0 0x67 0x45 0x23 0x01 31      リトルエンディアン

# エンディアンの注意事項 1/2

ビッグエンディアン=>リトルエンディアン変換時のCソースチェック項目を説明します。

## 1. 共用体

配列等を用いた場合には注意が必要です。

```
union {  
    unsigned long l;  
    unsigned short s[2];  
} UN;  
UN.s[0] = 0x1234;  
UN.s[1] = 0x5678;
```

ビッグエンディアン : UN.l = 0x12345678

0	0x12	0x34	0x56	0x78	31
---	------	------	------	------	----

リトルエンディアン : UN.l = 0x56781234

0	0x34	0x12	0x78	0x56	31
---	------	------	------	------	----



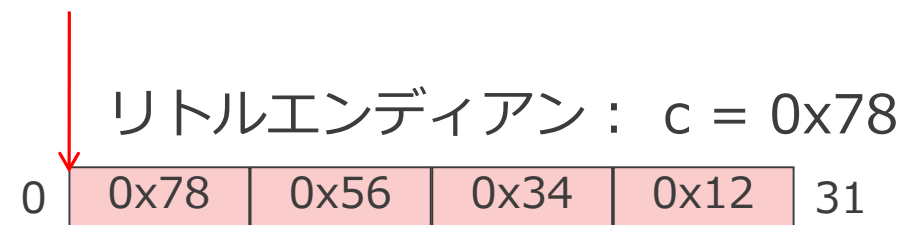
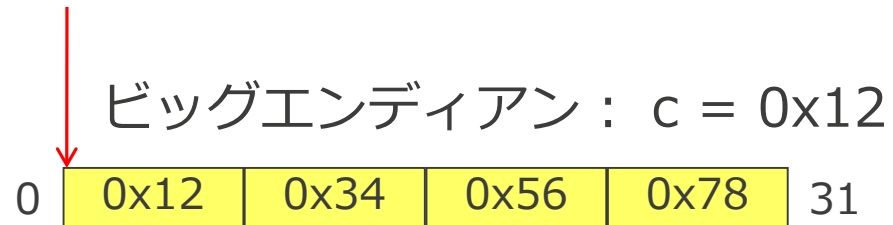
# エンディアンの注意事項 2/2

## 2. ポインタ型変換

ポインタ型を変換してアクセスしている場合には注意が必要です。  
変換後の型に応じてoffset値を調整する必要があります。

```
unsigned long *p;  
(*p) = 0x12345678  
unsigned char c = *((unsigned char *)&p);
```

ポインタ"p"の参照先



# 改訂履歴

---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014/11/11	-	初版
1.01	2016/11/2	P10	Rev.1.00の「補足:エンディアンの注意事項3」を削除

---

ルネサス システムデザイン株式会社