



BCD ライブラリマニュアル

このマニュアルは、BCD ライブラリの仕様について記載します。

ご注意

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2009 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

目次

1. 概要	1
2. 動作環境	2
3. 構造体一覧	3
3.1 BCD_DATA	3
3.2 bcd_exception	4
4. 関数一覧	5
4.1 bcd_mov	6
4.2 bcd_atobcd	7
4.3 bcd_bcdtoa	8
4.4 bcd_ltobcd	9
4.5 bcd_bcdtol	10
4.6 bcd_ultobcd	11
4.7 bcd_bcdtoul	12
4.8 bcd_abs	13
4.9 bcd_modfint	14
4.10 bcd_modflow	15
4.11 bcd_sign	16
4.12 bcd_round	17
4.13 bcd_add	18
4.14 bcd_add2	19
4.15 bcd_sub	20
4.16 bcd_sub2	21
4.17 bcd_mul	22
4.18 bcd_mul2	23
4.19 bcd_div	24
4.20 bcd_div2	25
4.21 bcd_pow	26
4.22 bcd_log10	27
4.23 bcd_accuracy	28
4.24 bcd_getaccuracy	29
4.25 bcd_matherr	30
4.26 bcd_geterr	31

2. 動作環境

BCD ライブラリの動作環境を以下に示します。

対象機種

- DT-9800
- IT-9000

対象 OS

- Microsoft WindowsCE 5.0
- Microsoft WindowsCE 6.0

開発環境

- Microsoft embedded Visual C++ Version 4.0 + SP4
- Microsoft Visual Studio 2005 + SP1
- Microsoft Visual Studio 2008 + SP1

提供ファイル

- DTBCD.h
- LDTBCD.lib
- LDERR.lib
- LDTBCD.dll

使用方法

- プログラムソース内に **DTBCD.h** をインクルードし、リンカの依存ファイルとして **LDTBCD.lib** (必要に応じて **LDERR.lib**) を指定してください
- **LDTBCD.dll** は本体に内蔵されています。

3. 構造体一覧

3.1 BCD_DATA

本ライブラリを使用して演算を行うための情報を格納します。

```
#define BCD_PLUS    1          // 正の数値
#define BCD_MINUS  -1         // 負の数値

typedef struct{
  char          exp;          // 指数-99~99
  char          sign :4;     // 符号 1: 正 -1: 負
  unsigned char integer :4;  // 仮数の整数部 BCD1桁
  unsigned char kasu[12];    // 仮数の小数部
                                // 倍精度(12バイト) BCD24桁
                                // 単精度(6バイト) BCD12桁
}BCD_DATA;
```

- 単精度場合、仮数の小数部 kasu のうち、6バイトのみ使用します
- 倍精度のデータを単精度で使用する場合、下位6バイト分のデータが無効となります
- 単精度のデータを倍精度で使用する場合、下位6バイト分のデータが有効となります

例

BCD_DATA: 14Byte

-123.456 = -1.23456 × E+2

exp : 指数(8ビット: -99~99)
sign : 符号(4ビット: 1, -1)
integer : 仮数の整数部(4ビット: 1~9)
kasu : 仮数の小数部(12バイト)

	2
	-1
	1
2	3
4	5
6	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0

定義例

```
BCD_DATA = bcd_data =
{ 2, BCD_MINUS, 1, 0x23, 0x45, 0x60 };
```

3.2 bcd_exception

本ライブラリ内で発生したエラー情報を格納します。

```
struct bcd_exception{
    int         type;           // エラーの種類
    char        *name;         // エラー発生関数名
    BCD_DATA    *arg1, *arg2;  // 関数の第1引数, 第2引数
};
```


4. 関数一覧

関数名	機能	DT-9800	IT-9000
bcd_mov	指定した数値を代入します	○	○
bcd_atobcd	指定した文字列を数値に変換します	○	○
bcd_bcdtoa	指定した数値を文字列に変換します	○	○
bcd_ltobcd	符号付 long 型の数値を BCD_DATA 型に変換します	△	△
bcd_bcdtol	BCD_DATA 型の数値を符号付 long 型の数値に変換します	△	△
bcd_ultobcd	符号無 long 型の数値を BCD_DATA 型に変換します	△	△
bcd_bcdtoul	BCD_DATA 型の数値を符号無 long 型の数値に変換します	△	△
bcd_abs	指定した数値の絶対値を求めます	○	○
bcd_modfint	指定した数値の整数部を取得します	○	○
bcd_modflow	指定した数値の小数部を取得します	○	○
bcd_sign	指定した数値の符号を取得します	○	○
bcd_round	数値を指定した桁数で切り上げ/切り下げ/四捨五入します	○	○
bcd_add	指定した数値を加算します	○	○
bcd_add2	指定した数値を加算します	○	○
bcd_sub	指定した数値を減算します	○	○
bcd_sub2	指定した数値を減算します	○	○
bcd_mul	指定した数値を乗算します	○	○
bcd_mul2	指定した数値を乗算します	○	○
bcd_div	指定した数値を除算します	○	○
bcd_div2	指定した数値を除算します	○	○
bcd_pow	指定した数値をべき乗します	○	○
bcd_log10	指定した数値を常用対数演算します	○	○
bcd_accuracy	演算の精度を設定します	○	○
bcd_getaccuracy	演算の精度を取得します	△	△
bcd_matherr	本ライブラリ内で発生したエラー処理を行います	○	○
bcd_geterr	本ライブラリ内で発生したエラー情報を取得します	△	△

○ 関数サポート

△ 開発予定関数(使用できません)

— 関数未サポート= 関数を呼ぶと未サポートエラーが返ります。

※ BCD 演算ライブラリは、本体にインストールすることにより上記全てのモデルにおいて、使用可能ですが、製品仕様として、○以外の機種に関しては機能が存在しないため、使用禁止です。

4.1 bcd_mov

指定した数値を代入します。

```
[C++]
BCD_DATA *bcd_mov (
    BCD_DATA *src,
    BCD_DATA *dst
)
```

説明

本関数は指定した数値を代入します。

パラメータ

src

代入する数値の構造体を指定します。

dst

代入先の構造体を指定します。

戻り値

正常終了時は *dst* を返します。

4.2 bcd_atobcd

指定した文字列を数値に変換します。

```
[C++]
void bcd_atobcd (
    char      *string,
    BCD_DATA *result
)
```

説明

本関数は指定した文字列を数値に変換します。

変換可能な文字列は、10進の数字/符号('+', '-')/先頭のスペース/小数点('.')です。

'¥0'もしくは、変換不可能な文字の直前の文字までを変換します。

パラメータ

string

変換する文字列を指定します。

dst

変換した数値を格納する構造体を指定します。

戻り値

なし

4.3 bcd_bcdtoa

指定した数値を文字列に変換します。

```
[C++]
void bcd_bcdtoa (
    BCD_DATA *value,
    char      *restring,
    int       strsize
)
```

説明

本関数は指定した数値を文字列に変換します。

数値が小数の場合は '.' を、負の場合は先頭に '-' を付加します。

restring に格納する文字列は *strsize*-1 文字までであり、変換された文字列が *strsize* 以上ある場合は、超える部分が整数部ならエラーとし、少数部ならば *strsize*-1 までを格納し正常終了します。

少数部末尾の 0 は文字に変換しません。

パラメータ

value

変換する数値の構造体を指定します。

restring

変換した数値を格納する領域を指定します。

strsize

restring のサイズを指定します。

戻り値

なし

4.4 bcd_ltobcd

符号付 long 型の数値を BCD_DATA 型に変換します。

```
[C++]
void bcd_ltobcd (
    long      ldata,
    BCD_DATA  *result
)
```

説明

本関数は符号付 long 型の数値を BCD_DATA 型に変換します。

パラメータ

ldata

変換する long 型の数値を指定します。

result

変換した数値を格納する構造体のポインタを指定します。

戻り値

なし

4.5 bcd_bcdtol

BCD_DATA 型の数値を符号付 long 型の数値に変換します。

```
[C++]  
long bcd_bcdtol (  
    BCD_DATA *value  
)
```

説明

本関数は BCD_DATA 型の数値を符号付 long 型の数値に変換します。

パラメータ

value

変換する数値の構造体を指定します。

戻り値

正常終了時は符号付 long 型の数値を返します。

4.6 bcd_ultobcd

符号無 long 型の数値を BCD_DATA 型に変換します。

```
[C++]
void bcd_ultobcd (
    unsigned long ldata,
    BCD_DATA      *result
)
```

説明

本関数は符号無 long 型の数値を BCD_DATA 型に変換します。

パラメータ

ldata

変換する unsigned long 型の数値を指定します。

result

変換した数値を格納する構造体を指定します。

戻り値

なし

4.7 bcd_bcdtoul

BCD_DATA 型の数値を符号無 long 型の数値に変換します。

```
[C++]
unsigned long bcd_bcdtoul (
    BCD_DATA *value
)
```

説明

本関数は BCD_DATA 型の数値を符号無 long 型の数値に変換します。

パラメータ

value

変換する数値の構造体を指定します。

戻り値

正常終了時は符号無 long 型の数値を返します。

value に指定した数値が符号無 long のサイズを超えている場合、エラーとなり BCD_VALUERR を返します。

4.8 bcd_abs

指定した数値の絶対値を求めます。

```
[C++]
BCD_DATA *bcd_abs (
    BCD_DATA *value,
    BCD_DATA *absolute
)
```

説明

本関数は指定した数値の絶対値を求めます。

パラメータ

value

絶対値を求める数値の構造体を指定します。

absolute

絶対値を格納する構造体を指定します。

戻り値

正常終了時は *absolute* を返します。

4.9 bcd_modfint

指定した数値の整数部を取得します。

```
[C++]
BCD_DATA *bcd_modfint (
    BCD_DATA *value,
    BCD_DATA *result
)
```

説明

本関数は指定した数値の整数部を取得します。

指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

value

整数部を求める数値の構造体を指定します。

result

整数部を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.10 bcd_modflow

指定した数値の小数部を取得します。

```
[C++]
BCD_DATA *bcd_modflow (
    BCD_DATA *value,
    BCD_DATA *result
)
```

説明

本関数は指定した数値の小数部を取得します。

指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

value

小数部を求める数値の構造体を指定します。

result

小数部を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.11 bcd_sign

指定した数値の符号を取得します。

```
[C++]
int bcd_sign (
    BCD_DATA *value
)
```

説明

本関数は指定した数値の符号を取得します。

パラメータ

value

符号を求める数値の構造体を指定します。

戻り値

正常終了時は指定した数値の符号を返します。

-1	:	*value < 0
0	:	*value = 0
1	:	*value > 0

4.12 bcd_round

数値を指定した桁数で切り上げ/切り下げ/四捨五入します。

```
[C++]
BCD_DATA *bcd_round (
    BCD_DATA *value,
    int      keta,
    int      mode
)
```

説明

本関数は数値を指定した桁数で切り上げ/切り下げ/四捨五入します。

指定した桁が正または 0 の場合は整数部の桁を、負の場合は小数部の桁とみなします。

処理モードが切り上げの場合は、指定した桁より下の値がすべて 0 ならば切り上げを行います。

パラメータ

value

変換する数値の構造体を指定します。

keta

処理を行う桁を指定します。

mode

処理モードを指定します。

R_UP	: 切り上げ
R_DOWN	: 切り下げ
R_4DOWN5UP	: 四捨五入

戻り値

正常終了時は *value* を返します。

4.13 bcd_add

指定した数値を加算します。

```
[C++]
BCD_DATA *bcd_add (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を加算し、結果を丸めます。
演算結果を丸めない場合は、bcd_add2関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

加算する数値の構造体を指定します。

valueY

加算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.14 bcd_add2

指定した数値を加算します。

```
[C++]
BCD_DATA *bcd_add2 (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を加算し、結果を丸めません。

演算結果を丸める場合は、bcd_add関数を使用してください。

指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

加算する数値の構造体を指定します。

valueY

加算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.15 bcd_sub

指定した数値を減算します。

```
[C++]
BCD_DATA *bcd_sub (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を減算し、結果を丸めます(ValueX-ValueY)。
演算結果を丸めない場合は、bcd_sub2関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

減算する数値の構造体を指定します。

valueY

減算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.16 bcd_sub2

指定した数値を減算します。

```
[C++]
BCD_DATA *bcd_sub2 (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を減算し、結果を丸めません (ValueX-ValueY)。
演算結果を丸める場合は、bcd_sub関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

減算する数値の構造体を指定します。

valueY

減算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.17 bcd_mul

指定した数値を乗算します。

```
[C++]
BCD_DATA *bcd_mul (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を乗算し、結果を丸めます。
演算結果を丸めない場合は、bcd_mul2関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

乗算する数値の構造体を指定します。

valueY

乗算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.18 bcd_mul2

指定した数値を乗算します。

```
[C++]
BCD_DATA *bcd_mul2 (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を乗算し、結果を丸めません (ValueX-ValueY)。
演算結果を丸める場合は、bcd_mul関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

乗算する数値の構造体を指定します。

valueY

乗算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.19 bcd_div

指定した数値を除算します。

```
[C++]
BCD_DATA *bcd_div (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を除算し、結果を丸めます(ValueX/ValueY)。
演算結果を丸めない場合は、bcd_div2関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

除算する数値の構造体を指定します。

valueY

除算する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.20 bcd_div2

指定した数値を除算します。

```
[C++]
BCD_DATA *bcd_div2 (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を除算し、結果を丸めません (ValueX/ValueY)。
演算結果を丸める場合は、bcd_div関数を使用してください。
指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

除算する数値の構造体を指定します。

valueY

除算する数値の構造体を指定します。

result

除算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.21 bcd_pow

指定した数値をべき乗します。

```
[C++]
BCD_DATA *bcd_pow (
    BCD_DATA *valueX,
    BCD_DATA *valueY,
    BCD_DATA *result
)
```

説明

本関数は指定した数値をべき乗し、結果を丸めます ($(\text{ValueX})^{(\text{ValueY})}$)。

演算結果がアンダーフローした場合、結果は 0 になります。

指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

valueX

べき乗する数値の構造体を指定します。

valueY

べき乗する数値の構造体を指定します。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.22 bcd_log10

指定した数値を常用対数演算します。

```
[C++]
BCD_DATA *bcd_log10 (
    BCD_DATA *value,
    BCD_DATA *result
)
```

説明

本関数は指定した数値を常用対数演算し、結果を丸めます ($\log_{10}(\text{value})$)。指定する数値と演算結果は同じ構造体を指定してもかまいません。

パラメータ

value

常用対数演算する数値の構造体を指定します。
必ず正の値を指定してください。

result

演算結果を格納する構造体を指定します。

戻り値

正常終了時は *result* を返します。

4.23 bcd_accuracy

演算の精度を設定します。

```
[C++]
void bcd_accuracy (
    int mode
)
```

説明

本関数は演算の精度を設定します。

パラメータ

mode

演算の精度を指定します。

BCD_SINGLE : 単精度

BCD_DOUBLE : 倍精度

戻り値

なし

4.24 bcd_getaccuracy

演算の精度を取得します。

```
[C++]  
int bcd_getaccuracy ( )
```

説明

本関数は演算の精度を取得します。

パラメータ

なし

戻り値

正常終了時は演算の精度を返します。

BCD_SINGLE : 単精度

BCD_DOUBLE : 倍精度

4.25 bcd_matherr

本ライブラリ内で発生したエラー処理を行います。

```
[C++]
int bcd_matherr (
    struct bcd_exception *except
)
```

説明

本関数は本ライブラリ内で発生したエラー処理を行います。

本ライブラリの関数は、エラーを検地すると本関数を呼び出します。

独自のエラー処理を行う場合は、本関数を定義して、リンクしてください。

パラメータ

except

数値演算エラー情報構造体を指定します。

戻り値

エラー処理が正常に終了すると、0 以外の値を返します。

本関数でエラーが発生すると、0 を返します。

本関数でエラーが発生した場合でも、エラー情報をセットして終了します。

4.26 bcd_geterr

本ライブラリ内で発生したエラー情報を取得します。

```
[C++]  
int bcd_geterr ( )
```

説明

本関数は本ライブラリ内で発生したエラー情報を取得します。

エラー情報は外部変数 `bcd_errno` に格納されています。

本関数で `bcd_errno` を取得後、`bcd_errno` をクリアします(詳細は補足を参照してください)。

パラメータ

なし

戻り値

エラー情報を返します。

<code>BCD_NOERROR</code>	: 正常終了
<code>BCD_PARAERR</code>	: パラメータエラー
<code>BCD_OVERFLOW</code>	: 演算結果オーバーフロー
<code>BCD_MATHERR</code>	: 数学的エラー
<code>BCD_DOMAIN</code>	: 引数が定義域以外
<code>BCD_STRERR</code>	: 不正な数値文字列
<code>BCD_VALUERR</code>	: 不正な数値データ

補足

各関数実行後は、必ず本関数を実行し、エラー情報を取得してください。

本ライブラリは、`bcd_errno` にエラー番号がセットされていると、処理を行いません。

本関数を実行することにより、`bcd_errno` が 0 (正常状態) になります。

カシオ計算機お問い合わせ窓口

製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<http://casio.jp/support/ht/>

製品の取扱い方法のお問い合わせ

- 情報機器コールセンター



0570-022066

市内通話料金でご利用いただけます。

携帯電話・PHS 等をご利用の場合、**048-233-7241**

カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)