

BCD 演算ライブラリ (vlbcd_lib) デモプログラム説明書 (RH850 版)

(R1.30)

商標等の表記について

- ・全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。
- ・パーソナルコンピュータの略号を PC とします。

1. 概要

本プログラムは、RH850-can5 実験キットのテストプログラムに BCD 演算ライブラリを組込んだデモ用プログラムです。

テストプログラムに対して、以下の相違があります。

①1,000 桁関数電卓機能の追加：ホスト PC から以下のコマンドにより計算します。

va 有効桁数	有効桁数(12~1,000)を設定
ve モード	有効桁数のみ表示時は 0, 内部桁数全てを表示時は 1 を設定
vf モード	通常(高速乗除算)時は 0, 省メモリ(低速乗除算)時は 1 を設定
vg モード	三角関数の単位が角度の場合は 0, ラジアンの場合は 1 を設定
vn 数値	n=0~9。BCD 変数 R0~R9 に数値を設定
vd n	n=0~10 時は BCD 変数 R0~R10 の内容を表示 n=11: スクリプト表示、n=12: メッセージ表示 (R1.3 で変更)
vb スクリプト	精度や時間を測定するスクリプトの書込み
vc コマンド番号	コマンド番号に応じた演算を実行

35, 36, 37, 38, 39 : +, -, *, /, %(剰余)

40, 41, 42, 43, 44 : 移動 Rn→Rn+1, 移動 Rn+1→Rn, 逆数, $\sqrt{\quad}$, $\sqrt[3]{\quad}$

45, 46, 47, 48, 49 : 複写 R0→R1, 交換 R0↔R1, x^y , x^2 , x^3

50, 51, 52, 53, 54 : sin, cos, tan, exp, 10^x

55, 56, 57, 58, 59 : asin, acos, atan, loge, log10

60, 61, 62, 63, 64 : sinh, cosh, tanh, π , e(ネイピア数)

65, 66, 67, 68, 69 : asinh, acosh, atanh, ACLR, MCLR

70~74, 75~79 : 移動 R0→R5~R0→R9, 複写 R0←R5~R0←R9

85, 86, 87, 90, 91 : !, +/, |x|, 9 詰め, 丸め

92, 93, 94, 95 : 整数部取得, 小数部取得, 指数部取得, スクリプト実行

②バッファサイズ変更：1,000 桁の数値を送受信するために UART および stdio のバッファサイズを変更しています。コンパイル・オプションにより変更できるように task_UART.h, stdio.h を若干修正しています。

2. フォルダ／プログラム構成

構成は以下の通りです(はフォルダ名)。

<input type="text" value="RH850_vlbc"/>	
└ RH850_vlbc.mot	ビルド済み実行コード
└ RH850_vlbc.mtpj	ビルド用プロジェクト・ファイル
└ iodef.h	IO 定義 (テストプログラムと同じ)
└ <input type="text" value="SRC"/>	ビルド用ソース・コード (機種依存部)
└ main.c	main 関数 (関数電卓機能の呼出しを追加)
└ TSmain.c	関数電卓機能
└ all_define.h	main.c と TSmain.c 間のインタフェース用定義
└ vlbc_api.h	BCD 演算ライブラリを使用するためのヘッダ
└ vlbc_lib.h/c	BCD 演算ライブラリ
└ vlbc_lib_xxx.h	BCD 演算ライブラリに関連した数値テーブル
└ その他	テストプログラムと同じ
<input type="text" value="COMMON"/>	
└ stdio.h	標準入出力関数ヘッダ(バッファ長設定対応)
└ task_uart.h	UART 通信関数ヘッダ(バッファ長設定対応)
└ その他	テストプログラムと同じ

3. テスト方法

3.1. ソフトウェアの準備

(1)汎用通信ソフト

1,000 桁の数値を入出力するため、端末サイズが設定できる通信ソフト (例えば Tera Term 最新版) をインストールします。

- 端末サイズ : 220 x 24~60
- 改行コード : 受信 LF、送信 LF
- ローカルエコー : あり
- 漢字コード : SJIS (シフト JIS)
- スピード : 460800 [bps]
- データ : 8bit
- パリティ : none
- ストップビット : 1bit
- フロー制御 : none

- (2)USB ドライバのインストール テストプログラムと同じ
- (3)ビルド環境 テストプログラムと同じ

3.2. ハードウェアの準備

テストプログラムと同じで良いです。

関数電卓機能のみ実行する場合は、LCD 無しでも構いません。

3.3. テスト手順

3.3.1. スイッチ、LED、ブザーのテスト

3.3.2. 発振周波数の確認

3.3.3. 1文字コマンドによるテスト

3.3.4. 行コマンドによるテスト

行コマンドはリターンキー(Enter キー)を押してから有効になります。リターンキーを押す前まではバックスペースキーによる修正も可能です。バックスペースキー以外の編集キーは無効です。

(1) RS-485 通信

(2)LCD 表示

(3)パラメータ読み書き (本版では I2C メモリに保存)

(4)PWM 調光

(5)I2C EEPROM 制御

(6)関数電卓機能

「1.概要」に記載のコマンドを入力します。応答メッセージは次の通りです。

- ①va #signif:有効桁数, internal:内部桁数
- ②ve #bare=設定値
- ③vf #compact=設定値
- ④vg #angle=設定値
- ⑤v0~v9 #ok (注:数値として認識しなくても ok と出ます)
- ⑥vd #Rn:内容
- ⑦vb #ok (注:内容が正しくなくても ok と出ます)
 スクリプトの内容は vlbcd_lib 説明書(「下記を実行」)を参照
- ⑧vc #R0:演算結果
 #time:実行時間(単位は μ s)

4. ビルド方法

4.1. 既存プロジェクトを使用する場合

(1) 「プロジェクトを開く」で RH850_vlbcd.mtpj ファイルを指定してください。

(2) ソースを閲覧／編集する場合は、下記設定を確認してください。

①エディタ設定

「ツール」→「オプション」→の

「全般」→「表示」→「タブの桁数」は 4

「テキストエディタ」→「エンコーディング」のデフォルトはシフト JIS

4.2. 新規にプロジェクトを作成する場合

(1)新規プロジェクト作成画面

マイクロコントローラとして RH850/F1KM の R7F701690(LQFP64)を選択してください。

(2)CC-RH(ビルドツール)のプロパティ設定

①共通オプション

a) コンパイルの追加のインクルードパスとして下記 3 つを追加してください。

SRC

..¥COMMON

.

b) レジスタ・モード→「ep レジスタの扱い」を「プロジェクト全体で固定する」にします。

②コンパイル・オプション

a)出力コード

・データのデフォルト・セクションを ep_disp16 にします。

・定数のデフォルト・セクションを zconst にします。

b)定義マクロ(テストプログラムと相違)

・次の 5 行は必須です。

nBCD_NUM_SIGNIF=1000

BCD_NON_SECURITY

UART_SEND_SIZE=1150

UART_RECV_SIZE=1150

STDIO_BUFFER_MAX=1200

- ・ スクリプトでプリミティブテストを実行する場合は以下を追加します。

USE_PAID_VERSION

ただし無償版コンパイラの場合はメモリ制限エラーになる可能性があります(未確認)。

- ・ 必要に応じて以下を追加します(vlbcd_lib の説明書参照)。

BCD_NUM_FIXED

BCD_NO_EVALUATION

BCD_BASIC_FUNC

BCD_COMPACT_MEM

③リンク・オプション

a)セクション

- ・ セクションの開始アドレスを以下にします。

```
RESET/%ResetVectorPE1%,.zconst,.text,.const,.edata,.data/0000200,.stack.bss,EIINTT  
BL.bss,.ebss,.edata.R,.bss,.data.R/FEDE0000
```

- ・ ROM から RAM へマッピングするセクションを以下にします。

.data=.data.R

.edata=.edata.R

(3)E1(LPD)もしくは E2(LPD)のプロパティ設定

①接続用設定

- ・ メイン・クロック周波数 16.00

②フラッシュ・オプション設定(「デバッグ」→「デバッグ・ツールへ接続」の後)

- ・ OPBT0 BA2AFDCF

(4)ファイル

- ①デフォルトで生成された boot.asm, cstart.asm, main.c を「プロジェクトから外し」ます。iodefine.h はそのまま残します。

- ②ファイル指定

「ファイル」→「追加」→「新しいカテゴリを作成」で、SRC と COMMON を作成します。COMMON および SRC フォルダの下のファイルをすべて登録します。

4.3. ビルド、書き込み、デバッグ

(1)ビルドのみ行う場合

「ビルド」→「ビルド・プロジェクト」を選択します。

(2)ビルドおよび書き込み／実行を行う場合

①E1 または E2 エミュレータを接続していることを確認してください。

②「デバッグ」→「ビルド&デバッグ・ツールヘダウンロード」を選択します。

ダウンロード後 main 関数の最初で止まります。

リセット位置で止めたい場合は、「デバッグ・ツール」のプロパティを開き、

「ダウンロード・ファイル設定」→「デバッグ情報」→「指定シンボル」を

`__reset` に変更します。これにより main 関数前に実行する `init.c` や `device.c` でブレークをかけることができます。

③「デバッグ」→「実行」で動作を開始します。

5. プログラム詳細

5.1. マイコン内のリソース割り当て

5.1.1. クロック周波数

5.1.2. 周辺クロック入力設定

5.1.3. シリアル I/F

5.1.4. タイマ

5.1.5. メモリ消費

(1)FROM 233KB (電卓・BCD 演算ライブラリは 1,000 桁時 31.3KB)

項目	評価機能あり	評価機能なし
桁数動的変更、全機能、	31.3*	28.9*
桁数固定(BCD_NUM_FIXED)、全機能	27.8*	25.4*
桁数固定、四則演算(BCD_BASIC_FUNC)	16.7	14.5
桁数固定、四則演算、低速(BCD_COMPACT_MEM)	(無効)16.7	13.6

評価機能なし：BCD_NO_EVALUATION 指定

単位：KB

*印：vlbcd ライブラリ説明書の値は R1.20 版です。こちらが R1.30 版の値です。

(2)RAM 42KB (内スタック 4KB, ベクターテーブル 2KB)

BCD 演算ライブラリ：10KB (主に数値ワーク 15 本)

TSmain(関数電卓機能)：7KB(主に BCD 変数 10 本)

main.c 内の関数電卓用：14KB(主に文字列パラメータ)

(3)データフラッシュ 本版では不使用

(4) vlbcd_lib 単独の ROM/RAM 消費量

項目	評価機能あり	評価機能なし
桁数動的变化、全機能、	25.4 / 10.0	23.1 / 10.0
桁数固定(BCD_NUM_FIXED)、全機能	22.0 / 10.0	19.6 / 10.0
桁数固定、四則演算(BCD_BASIC_FUNC)	12.4 / 10.0	10.2 / 6.7
桁数固定、四則演算、低速(BCD_COMPACT_MEM)	12.4 / 10.0	9.3 / 2.0

評価機能なし : BCD_NO_EVALUATION 指定

単位 : KB

5.1.6. ポート

5.1.7. 割込み、優先度

5.2. 動作順序

5.3. 主な関数の使用方法

5.3.1. 標準入出力関数 (stdio.h)

5.3.2. アナログ入力取得関数 (task_ad.h)

5.3.3. スイッチ入力関数 (task_sw.h)

5.3.4. ブザー出力関数 (task_beep.h)

5.3.5. クロック出力関数 (task_clock.h)

5.3.6. LED 点滅指示関数 (task_led.h)

5.3.7. CAN 通信関数 (task_can.h)

5.3.8. PWM 出力関数 (task_pwm.h)

5.3.9. I2C EEPROM 制御関数 (task_i2c.h)

5.4. カーネル仕様

改訂履歴

版	日付	記事
1.1	2025/12/17	初回リリース版
1.3	2025/12/23	vlbcd_lib_R130 対応